



Universidade Federal
do Rio de Janeiro
Escola Politécnica

APLICAÇÃO DE REDES NEURAS ARTIFICIAIS COM TOPOLOGIA DEFINIDA POR
ALGORITMOS GENÉTICOS PARA RESOLUÇÃO DA CINEMÁTICA INVERSA DE
UM MANIPULADOR ROBÓTICO COM 6 GRAUS DE LIBERDADE

Lucas de Oliveira Suplino

Projeto de Graduação apresentado ao Curso de Engenharia Mecânica de Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários a obtenção do título de Engenheiro.

Orientador: Max Suell Dutra

Rio de Janeiro

Julho 2018

APLICAÇÃO DE REDES NEURAIS ARTIFICIAIS COM TOPOLOGIA DEFINIDA POR
ALGORITMOS GENÉTICOS PARA RESOLUÇÃO DA CINEMÁTICA INVERSA DE
UM MANIPULADOR ROBÓTICO COM 6 GRAUS DE LIBERDADE

Lucas de Oliveira Suplino

PROJETO FINAL SUBMETIDO AO CORPO DOCENTE DO DEPARTAMENTO DE
ENGENHARIA MECÂNICA DA ESCOLA POLITÉCNICA DA UNIVERSIDADE
FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS
PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO MECÂNICO.

Aprovado por:

Prof. Max Suell Dutra - Orientador

Prof. Jules Ghislain Slama

Prof. Vitor Ferreira Romano

RIO DE JANEIRO, RJ - BRASIL

JULHO 2018

Suplino, Lucas de Oliveira

Aplicação de redes neurais artificiais com topologia definida por algoritmos genéticos para resolução da cinemática inversa de um manipulador robótico com 6 graus de liberdade / Lucas de Oliveira Suplino – Rio de Janeiro: UFRJ/ Escola Politécnica, 2018.

XIV, 86 p.:il.; 29,7 cm.

Orientador: Max Suell Dutra

Projeto de Graduação – UFRJ/Escola Politécnica/Curso de Engenharia Mecânica, 2018.

Referências Bibliográficas: p. 85 – 86.

1. Manipuladores Robóticos 2. Robótica 3. Redes Neurais Artificiais 4. Algoritmos Genéticos I. Dutra, Max Suell. II. Universidade Federal do Rio de Janeiro, Escola Politécnica, Curso de Engenharia Mecânica. III. Aplicação de redes neurais artificiais com topologia definida por algoritmos genéticos para resolução da cinemática inversa de um manipulador robótico com 6 graus de liberdade.

AGRADECIMENTOS

Agradeço a Deus por firmar meus pés e abrir o caminho. Obrigado pela garantia de que suas promessas sempre se cumprirão, sem Ti, eu nada seria.

Obrigado a minha família, por acreditarem em mim e me apoiarem incondicionalmente. Obrigado pelas orações e pelo encorajamento. Especificamente aos meus pais e minha irmã, obrigado pela paciência.

Aos amigos que fiz durante a jornada, vocês são um presente que a vida me deu. Obrigado pelas xerox, pelas manhãs, tardes e noites de estudo, pessoalmente ou por telefone, pelas conversas sobre a faculdade e pelas conversas sobre a vida. Vocês me ensinaram que é possível sorrir, literalmente, em qualquer situação.

Ao professor Max por manter as portas sempre abertas, e ao professor Luciano pelo apoio. Vocês foram essenciais para a conclusão desse trabalho.

“Se eu vi mais longe, foi por estar sobre o ombro de gigantes”.

*“Pois o Senhor é quem dá sabedoria; de sua boca
procedem o conhecimento e o discernimento.”*

Provérbios 2:6

Resumo do Projeto de Graduação apresentado à Escola Politécnica/UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro Mecânico.

APLICAÇÃO DE REDES NEURAS ARTIFICIAIS COM TOPOLOGIA DEFINIDA
POR ALGORITMOS GENÉTICOS PARA RESOLUÇÃO DA CINEMÁTICA INVERSA
DE UM MANIPULADOR ROBÓTICO COM 6 GRAUS DE LIBERDADE

Lucas de Oliveira Suplino

Julho/2018

Orientador: Max Suell Dutra

Curso: Engenharia Mecânica

No presente trabalho se verifica a possibilidade de utilização de um conjunto de redes neurais para resolução da cinemática inversa de um robô manipulador com 6 graus de liberdade. O robô industrial KR 10 R1420, da empresa Kuka, foi o escolhido para esse estudo por se tratar de um robô amplamente utilizado no mercado, e pela possibilidade de comparação entre os resultados de desempenho conseguidos e aqueles descritos em seu catálogo.

A partir da análise da geometria do manipulador definiram-se os parâmetros de Denavit-Hatenberg para a solução algébrica da cinemática inversa do mesmo, utilizando-se a abordagem proposta por Pieper, como será demonstrado. Para os cálculos das multiplicações matriciais necessárias a solução da cinemática do robô utilizou-se o programa Maple.

Resolvidas as cinemáticas inversa e direta do manipulador, a partir da análise algébrica, dois conjuntos de redes neurais foram treinados. O primeiro utilizando-se a

abordagem convencional de tentativa e erro para definir a topologia das redes neurais de maneira a otimizar o desempenho do conjunto, e o segundo utilizando-se algoritmos genéticos para auxiliar na fase de definição da topologia das redes. O algoritmo genético utilizado foi programado utilizando-se o programa Matlab, e as redes neurais foram treinadas também através do Matlab que dispõe de uma biblioteca com essa finalidade.

Palavras chave: manipuladores robóticos, robótica, redes neurais artificiais, algoritmos genéticos.

Abstract of Undergraduate Project presented to POLI/UFRJ as a partial fulfillment
of the requirements for the degree of Engineer.

APPLICATION OF ARTIFICIAL NEURAL NETWORKS WITH TOPOLOGY
DEFINED BY GENETIC ALGORITHMS TO SOLVE THE INVERSE KINEMATICS OF
A 6 DEGREES OF FREEDOM ROBOTIC MANIPULATOR

Lucas de Oliveira Suplino

July/2018

Advisor: Max Suell Dutra

Course: Mechanical Engineering

It was verified in this work the possibility of using a set of neural networks in order to solve the inverse kinematics of a 6-degree-freedom robotic manipulator. The Kuka industrial robot KR 10 R1420 was chosen during this study since it is largely used in industry and because of the possibility of comparing the results obtained to those described in its catalog.

From the analysis of the manipulator's geometry, the Denavit-Hatenberg parameters were defined and the algebraic solution of the inverse kinematics was found, using the approach proposed by Pieper, as will be demonstrated. To calculate the matrix multiplications necessary to solve the kinematics of the robot, the software Maple was used.

Once solved the manipulator's kinematic, from the algebraic analysis, two sets of neural networks were trained. The first using the conventional trial and error approach

to define the topology of neural networks in order to optimize the performance of the set , and the second using genetic algorithms to aid in the definition step of the network's topology. The genetic algorithm used was programmed using the software Matlab, and the neural networks were also trained through Matlab, which has a library for this purpose.

Keywords: robotic manipulators, robotics, artificial neural networks, genetic algorithms.

SUMÁRIO

| | |
|---|-----|
| LISTA DE FIGURAS..... | xii |
| LISTA DE TABELAS..... | xiv |
| 1. INTRODUÇÃO | 1 |
| 1.1. Motivação..... | 1 |
| 1.2. Objetivos | 3 |
| 1.3. Metodologia | 3 |
| 1.4. Estrutura do trabalho | 4 |
| 2. MODELAGEM CINEMÁTICA DE MANIPULADORES ROBÓTICOS..... | 5 |
| 2.1. Introdução..... | 5 |
| 2.2. Representação Algébrica de Manipuladores no Espaço | 8 |
| 2.3. Cinemática Direta..... | 11 |
| 2.3.1. Método de Denavit-Hatenberg | 11 |
| 2.3.2. Solução da cinemática direta de um manipulador com 3 graus de liberdade..... | 15 |
| 2.4. Cinemática Inversa | 16 |
| 2.4.1. Método Algébrico..... | 17 |
| 2.4.2. Solução de Pieper | 20 |
| 3. REDES NEURAIS ARTIFICIAIS..... | 24 |
| 3.1. Introdução | 24 |
| 3.2. Neurônio Biológico..... | 25 |
| 3.3. Neurônio Artificial: Modelo MCP | 27 |
| 3.4. Redes Perceptron | 28 |
| 3.5. Redes MLP | 31 |
| 3.6. Algoritmo de <i>Back-propagation</i> | 32 |
| 3.6.1. <i>Back-propagation</i> com <i>momentum</i> | 35 |
| 3.7. Funções de Propagação | 36 |
| 4. ALGORITMOS GENÉTICOS | 38 |
| 4.1. Introdução | 38 |
| 4.2. Conceitos Biológicos..... | 40 |
| 4.3. Componentes | 41 |
| 4.3.1. Representação..... | 42 |
| 4.4. Operadores Genéticos..... | 42 |

| | | |
|--------|---|----|
| 4.4.1. | Inicialização | 43 |
| 4.4.2. | Avaliação | 43 |
| 4.4.3. | Seleção | 44 |
| 4.4.4. | Reprodução | 45 |
| 4.4.5. | Mutação | 46 |
| 4.4.6. | Atualização e Elitismo | 47 |
| 4.4.7. | Finalização | 47 |
| 5. | APLICAÇÕES PRÁTICAS: KR 10 R1420..... | 48 |
| 5.1. | Introdução..... | 48 |
| 5.2. | Cinemática direta do manipulador | 52 |
| 5.3. | Cinemática inversa do manipulador | 54 |
| 5.4. | Utilização de redes neurais na solução da cinemática inversa do manipulador | 63 |
| 5.5. | Definição da topologia de uma rede neural através de algoritmos genéticos | 71 |
| 5.5.1. | Treinamento da nova rede neural definida | 76 |
| 6. | CONCLUSÕES..... | 82 |
| | BIBLIOGRAFIA..... | 84 |

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 - Manipulador paralelo modelo ABB IRB 340, retirado de [13]. | 5 |
| Figura 2 - Manipulador serial modelo ABB IRB6600, retirado de [4]. | 5 |
| Figura 3 - Volume de trabalho: Manipulador cartesiano (a); Manipulador esférico (b) e Manipulador SCARA (c), adaptado de [13]. | 6 |
| Figura 4 - Relação entre o problema cinemático direto e inverso. | 7 |
| Figura 5 - Posição do ponto P em relação ao referencial {A}, retirado de [16]. | 8 |
| Figura 6 - Rotação e translação do referencial {B} em relação ao referencial {A}, retirado de [16]. | 9 |
| Figura 7 - Representação algébrica de um manipulador serial composto por n elos [16]. | 10 |
| Figura 8 - Descrição de elos adjacentes como parte de uma cadeia cinemática, adaptado de [13]. | 12 |
| Figura 9 - Manipulador serial planar com 3 graus de liberdade, adaptado de [13]. | 15 |
| Figura 10 - Pulso esférico, retirado de [13]. | 21 |
| Figura 11 – Exemplo de arquitetura de uma RNA, retirado de [11]. | 24 |
| Figura 12 – Neurônio biológico, retirado de [11]. | 25 |
| Figura 13 - Potencial de ação de um neurônio, retirado de [1]. | 26 |
| Figura 14 - Neurônio de McCulloch e Pitts, adaptado de [17]. | 28 |
| Figura 15 - Rede Perceptron | 29 |
| Figura 16 - Rede MLP. | 31 |
| Figura 17 - Gráficos das funções de propagação. | 38 |
| Figura 18 - Estrutura de funcionamento de um AG, retirado de [7]. | 48 |
| Figura 19 - Manipulador KR 10 R1420 com 6 graus de liberdade, retirado de [18]. | 49 |
| Figura 20 – Representação simplificada do rôbo manipulador KR 10 R1420. | 49 |
| Figura 21 – Volume de trabalho do manipulador: vista frontal (sup.) e vista superior (inf.), retirado de [18]. | 51 |
| Figura 22 – Disposição dos referenciais locais e global no manipulador segundo o método de Denavit-Hartenberg. | 51 |
| Figura 23 –Projeção do vetor posição pQ do pulso do efetuador no plano x_0y_0 . | 56 |
| Figura 24 –Plano vertical formado pelos elos 2 e 3 do manipulador. | 57 |
| Figura 25 – As duas soluções possíveis para cinemática inversa. | 59 |
| Figura 26 - Entradas e saídas das redes neurais na solução do problema de posição inverso. | 64 |
| Figura 27 - Entradas e saídas das redes neurais na solução do problema de orientação inverso. | 65 |
| Figura 28 - Trajetória para teste das redes neurais. | 66 |
| Figura 29 - Erro nas coordenadas previstas para o pulso esférico. | 67 |
| Figura 30 - Erro no valor calculado para as juntas ao longo da trajetória. | 67 |
| Figura 31 - Erro na posição do pulso nas coordenadas x, y e z. | 68 |
| Figura 32 - Erro total de posição do pulso. | 68 |
| Figura 33 - Erro na posição do efetuador nas coordenadas x, y e z. | 69 |
| Figura 34 - Erro total de posição do efetuador. | 70 |
| Figura 35 - Erro na orientação do efetuador. | 70 |
| Figura 36 - Rastreamento da trajetória obtido. | 71 |

| | |
|---|----|
| Figura 37 – Diversidade de população após 50 gerações. | 74 |
| Figura 38 - Grau de adaptabilidade média, do indivíduo melhor adaptado e do indivíduo pior adaptado pelas gerações. | 75 |
| Figura 39 - Erro quadrático médio do habitante melhor adaptado. | 75 |
| Figura 40 – Avanço do erro quadrático médio durante o treinamento da rede neural original. | 76 |
| Figura 41 – Avanço do erro quadrático médio durante o treinamento da rede neural com a topologia definida pelo algoritmo genético. | 77 |
| Figura 42 - Erro no valor calculado para as juntas ao longo da trajetória. | 78 |
| Figura 43 - Erro na posição do pulso nas coordenadas x, y e z. | 79 |
| Figura 44 - Erro total de posição do pulso. | 79 |
| Figura 45 - Erro na posição do efetuador nas coordenadas x, y e z. | 80 |
| Figura 46 - Erro total de posição do efetuador. | 80 |
| Figura 47 - Erro na orientação do efetuador. | 81 |

LISTA DE TABELAS

| | |
|---|----|
| Tabela 1 - Parâmetros de DH para o manipulador planar de 3 juntas | 15 |
| Tabela 2 – Correspondência: Biologia x Algoritmos Genéticos | 41 |
| Tabela 3 – Exemplos de representações e seus correspondentes fenótipos para alguns problemas | 42 |
| Tabela 4 – Intervalo de operação das juntas | 50 |
| Tabela 5 – Parâmetros de Denavit-Hartenberg para o manipulador estudado..... | 52 |
| Tabela 6 – Parâmetros de treinamento das redes..... | 66 |

1. INTRODUÇÃO

1.1. Motivação

A ideia da automatização de sistemas e da utilização de robôs para realização de tarefas, com diferentes graus de complexidade, auxiliando ou substituindo completamente o elemento humano não é recente. Ainda em 1495 Leonardo da Vinci já havia esboçado planos para criação de um robô humanoide [3].

Em 1913 Henry Ford instalou a primeira linha de montagem do mundo baseada em esteiras na sua fábrica de automóveis. Em 1961 o Unimate, primeiro robô programável e propriedade da Unimation Company, foi instalado em uma linha de montagem da General Motors em Nova Jersey. Em 1976 braços robóticos foram acoplados nas sondas espaciais Viking I e II enviadas a Marte. E em 1996, a Honda revelou seu projeto iniciado 10 anos antes de robôs humanóides, conhecidos como ASIMO [4].

Se tratando da ambiente doméstico, em 1999 a Sony lançou a primeira versão do seu cão robótico, o Aibo, e em 2002 o robô aspirador de pó Roomba, lançado pela iRobot, ganhou o mercado, tendo vendido 2,5 milhões de unidades até 2008 [3].

De uma maneira geral, a robótica pode ser pensada para atender a ambientes com diferentes necessidades e graus de especificidade. Em 2012 a Universidade de Pittsburgh desenvolveu um braço mecânico que pode ser controlado pelo pensamento. Atualmente, diversas empresas têm estendido suas pesquisas para a criação de exoesqueletos.

Uma classe de robôs já utilizada em grande escala são os manipuladores robóticos, definidos pela norma ISO 10218 como máquinas manipuladoras, com vários graus de liberdade, controladas automaticamente, reprogramáveis, multifuncionais e que podem ter base fixa ou móvel, sendo utilizados em aplicações de automação industrial. Eles fazem parte das linhas de produção na indústria automobilística, auxiliam na exploração em ambientes inóspitos, como o espaço e os oceanos, e

fornece suporte para realização de cirurgias. Eles também estão presentes na indústria farmacêutica, química, cinematográfica, entre outros.

Os manipuladores robóticos em cadeia cinemática aberta são constituídos por uma cadeia de elos, rígidos ou flexíveis, conectados por juntas que conferem n graus de liberdade aos mesmos e um órgão terminal referente a tarefa para a qual foram projetados.

No contexto de se controlar a posição e orientação dos manipuladores no espaço 3D para execução de uma tarefa específica se torna crucial o entendimento dos aspectos geométricos do seu movimento. A cinemática é o ramo da mecânica que estuda a descrição do movimento, ignorando as forças e momentos que causam o movimento, campo estudado pela cinética. Em relação a robôs manipuladores, a cinemática analisa a relação entre as posições das juntas e o posicionamento do órgão terminal [13] [2].

A formulação da cinemática de um robô é dividida em dois problemas: a solução da cinemática direta do mesmo, que consiste em, dados os deslocamentos das juntas que compõe o manipulador, determinar a posição e orientação do efetuador no espaço; e a solução da cinemática inversa, que se trata da determinação do deslocamento das juntas do robô para que o órgão terminal alcance a coordenada desejada no espaço, com a orientação especificada [2].

A cinemática direta de robôs manipuladores, com elos rígidos, pode ser determinada pelo método de Denavit-Hatenberg [15]. Entretanto, a solução da cinemática inversa dos mesmos torna-se muito mais complexa dependendo das geometrias adotadas, e ainda é foco de diversos estudos, o que se justifica pela relevância dos manipuladores no mundo atual e pelas possibilidades de utilização ainda a serem exploradas. Assim, o presente trabalho foca na solução da cinemática inversa do manipulador KR 10 R1420 com elos rígidos e 6 graus de liberdade através do uso de redes neurais em conjunto com algoritmos genéticos.

1.2. Objetivos

Esse trabalho tem como objetivo analisar a possibilidade de utilização de um conjunto de redes neurais na solução da cinemática inversa do manipulador robótico KR 10 R1420, da empresa Kuka, com 6 graus de liberdade.

Além disso, o processo para determinação da arquitetura de uma rede neural dedicada para solução de um problema específico é altamente empírico, uma vez que não existe um método sistêmico capaz de determinar previamente a topologia ótima na solução do problema proposto. Assim, estuda-se nesse trabalho a possibilidade de utilização de algoritmos genéticos no processo de otimização da topologia da rede neural utilizada para o controle do manipulador robótico.

1.3. Metodologia

Inicialmente a convenção de Denavit-Hartenberg foi utilizada para solução da cinemática direta do manipulador. Obtida a matriz de transformação referente ao efetuador do manipulador estudado, o problema da cinemática inversa, foco do presente estudo, foi desacoplado em dois subproblemas: o problema de posição inverso do manipulador e o problema de orientação inverso, como proposto por Pieper [14].

O problema de posição inverso foi resolvido através de uma abordagem geométrica, enquanto o problema de orientação inverso foi resolvido de maneira algébrica, a partir das matrizes de orientação obtidas previamente, durante a solução da cinemática direta do manipulador.

Conhecida a solução matemática para o problema de posicionamento inverso do manipulador iniciou-se o estudo proposto, a saber, resolver o problema em questão através do uso de redes neurais. Para isso dois conjuntos de redes neurais foram treinados. No primeiro conjunto todas as redes neurais tiveram sua topologia definida pelo processo empírico de tentativa e erro, como é feito usualmente, já no segundo

conjunto foi utilizado um algoritmo genético que deveria auxiliar na etapa de definição da topologia das redes através de seu processo metódico de busca. Todas as redes neurais foram treinadas considerando-se todo o volume de trabalho do robô.

Para o treinamento das redes foi utilizada uma biblioteca específica fornecida pelo programa MatLab®. As simulações, algoritmos e gráficos também foram feitos utilizando-se esse programa. Os cálculos das matrizes para solução da cinemática direta do manipulador foram feitos utilizando-se o programa Maple®.

1.4. Estrutura do trabalho

Esse trabalho é composto por 6 capítulos, conforme segue:

- O capítulo 2 explica como é feita a análise cinemática de um manipulador. Nele destaca-se a diferença entre o problema da cinemática direta e o problema da cinemática inversa, e se exemplifica como ambos os problemas podem ser resolvidos para um manipulador com 3 graus de liberdade.
- Os capítulos 3 e 4 abordam o conceito de computação biológica falando, respectivamente, sobre redes neurais e algoritmos genéticos, as ferramentas utilizadas na solução do problema proposto.
- No capítulo 5 o manipulador KR 10 R1420, foco desse trabalho, é apresentado. A cinemática direta do mesmo é resolvida através do método de Denavit-Hartenberg, e a cinemática inversa é resolvida através da solução de Pieper. Finalmente, também afim de solucionar a cinemática inversa do manipulador, redes neurais são aplicadas em conjunto com algoritmos genéticos. Os resultados obtidos, em ambas as abordagens, são apresentados e comparados no final nesse capítulo.
- As conclusões advindas do trabalho, suas contribuições para a pesquisa da robótica e possibilidades de trabalhos futuros são discutidas no capítulo 6.

2. MODELAGEM CINEMÁTICA DE MANIPULADORES ROBÓTICOS

2.1. Introdução

Os manipuladores mecânicos são compostos por elos conectados por juntas rotacionais ou prismáticas constituindo as cadeias cinemáticas. Manipuladores de cadeias cinemáticas fechadas são chamados de manipuladores paralelos, onde um modelo desse tipo de manipulador é apresentado na Figura 1 enquanto manipuladores de cadeia aberta são chamados de manipuladores sereais, Figura 2.



Figura 1 - Manipulador paralelo modelo ABB IRB 340, retirado de [13].



Figura 2 - Manipulador serial modelo ABB IRB6600, retirado de [4].

Uma das extremidades desses manipuladores costuma ser fixada a uma base, enquanto a outra extremidade possui o órgão terminal, que é a ferramenta a ser utilizada na execução de uma determinada tarefa, e é livre para se movimentar, dentro do seu volume de trabalho, segundo a geometria e os graus de liberdade do manipulador [13]. Se assume que cada junta confere ao manipulador um grau de liberdade adicional, de maneira que um manipulador com n pares juntas-elos possuirá n graus de liberdade [4].

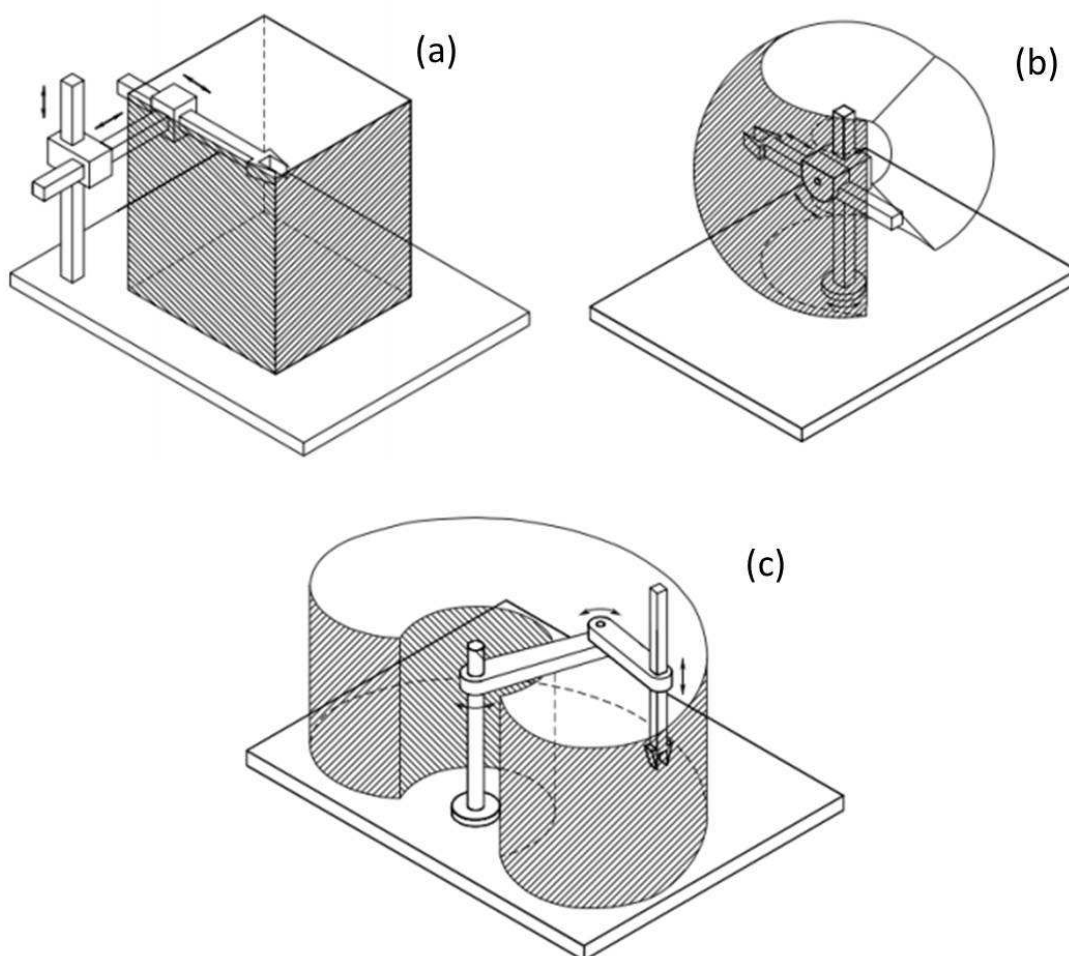


Figura 3 - Volume de trabalho: Manipulador cartesiano (a); Manipulador esférico (b) e Manipulador SCARA (c), adaptado de [13].

O volume de trabalho representa a porção no espaço 3D que pode ser alcançada pelo órgão terminal do manipulador, seu formato está diretamente relacionado com a

estrutura do manipulador e com os limites mecânicos das juntas do mesmo [13]. Na Figura 3 estão representados os volumes de trabalho de alguns manipuladores típicos:

A cinemática estuda a dependência entre a postura do manipulador robótico e o posicionamento das juntas no espaço de juntas em relação a um sistema de coordenadas fixo, sem considerar as forças e torques causadores do movimento. O estudo cinemático pode ser dividido em dois problemas fundamentais: o problema da cinemática direta, que consiste na determinação da posição e orientação do efetuador no espaço, conhecido os valores de suas variáveis de juntas; e o problema da cinemática inversa, onde se busca determinar o valor das variáveis de juntas correspondentes a uma postura desejada para o órgão terminal no espaço [2]. A relação entre o problema cinemático direto e inverso é resumida na Figura 4.

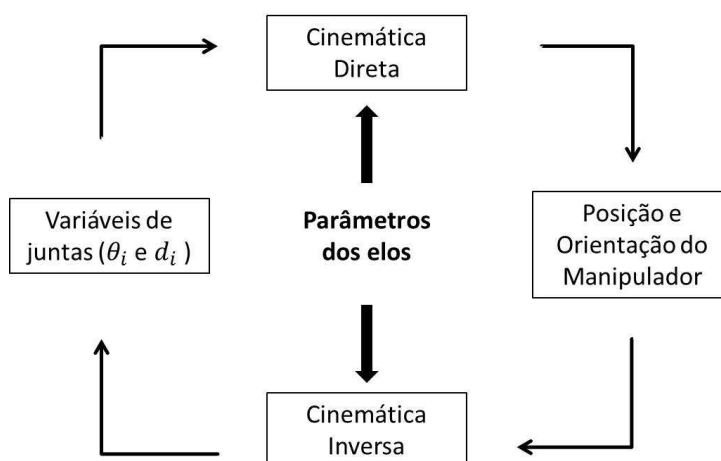


Figura 4 - Relação entre o problema cinemático direto e inverso.

A seguir se explica como é feita a descrição espacial de manipuladores em relação a um referencial e as transformações necessárias para a transição entre sistemas coordenados com diferentes orientações e posições no espaço, uma vez que essas ferramentas serão necessárias para compreensão da solução do problema cinemático enunciado.

2.2. Representação Algébrica de Manipuladores no Espaço

Para que a postura de um objeto seja completamente representada no espaço tridimensional faz-se necessário que tanto a posição quanto a orientação do mesmo sejam expressas em relação a um sistema de coordenadas. Nos manipuladores seriais, aqueles formados por cadeias cinemáticas abertas, a completa descrição de sua postura requer o estabelecimento de sistemas de coordenadas locais, fixados em cada um dos corpos componentes do manipulador, além do sistema de coordenadas inercial, fixo na base. Assim, um manipulador com n elos possuirá n sistemas de coordenadas locais e um sistema de coordenadas inercial [16].

Estabelecido um sistema de coordenadas $\{A\}$ onde, a_1, a_2, e, a_3 são vetores unitários mutuamente perpendiculares fixos em $\{A\}$, um ponto P pode ser localizado no espaço tridimensional a partir de um vetor 3×1 , cujos valores relacionam-se com os vetores a_1, a_2, e, a_3 como a seguir demonstrado na equação (2.1) e exemplificado na Figura 5:

$${}^A P = {}^A P \cdot a_1 a_1 + {}^A P \cdot a_2 a_2 + {}^A P \cdot a_3 a_3 \quad (2.1)$$

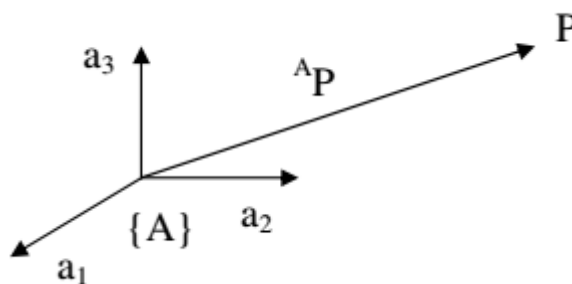


Figura 5 - Posição do ponto P em relação ao referencial $\{A\}$, retirado de [16].

A operação de rotação de um sistema de coordenadas $\{B\}$ em relação a um sistema de coordenadas $\{A\}$ pode ser descrita através de uma matriz de rotação 3×3

obtida a partir do cálculo do produto escalar, dois a dois, dos vetores de base dos sistemas, como demonstrado nas equações (2.2, 2.3 e 2.4):

$$\begin{bmatrix} {}^A P_1 \\ {}^A P_2 \\ {}^A P_3 \end{bmatrix} = \begin{bmatrix} a_1 \cdot b_1 & a_1 \cdot b_2 & a_1 \cdot b_3 \\ a_2 \cdot b_1 & a_2 \cdot b_2 & a_2 \cdot b_3 \\ a_3 \cdot b_1 & a_3 \cdot b_2 & a_3 \cdot b_3 \end{bmatrix} \begin{bmatrix} {}^B P_1 \\ {}^B P_2 \\ {}^B P_3 \end{bmatrix} \quad (2.2)$$

Ou,

$${}^A P = {}^A R_B {}^B P \quad (2.3)$$

Onde:

$${}^A R_B = \begin{bmatrix} {}^A X_B & {}^A Y_B & {}^A Z_B \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2.4)$$

Dados dois referenciais {A} e {B} inicialmente coincidentes e um ponto P fixo no referencial {B}, uma rotação do referencial {B} seguida de uma translação, ambas em relação ao referencial {A} gera uma mudança de orientação do vetor ${}^A P$ ao mesmo tempo em que vetor ${}^B P$ permanece invariável em {B}, conforme Figura 6:

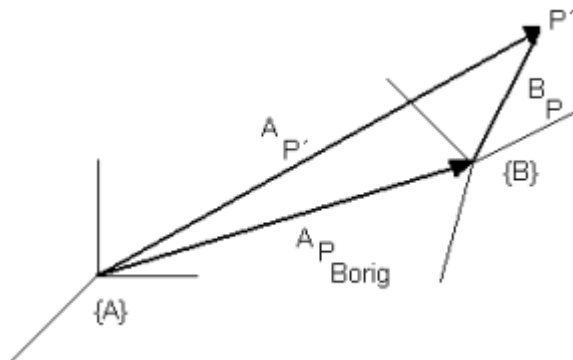


Figura 6 - Rotação e translação do referencial {B} em relação ao referencial {A}, retirado de [16].

Segundo ROMANO [16], as operações de translação e rotação podem ser representadas de maneira unívoca através da equação (2.5).

$${}^A P = {}^A P_{Borig} + {}^A R_B {}^B P \quad (2.5)$$

A equação (2.5) pode ser reescrita na forma matricial, conforme equação (2.6) utilizando-se a matriz de transformação homogênea ${}^A T_B$, uma matriz 4x4, estruturada conforme equação (2.7).

$${}^A P = {}^A T_B {}^B P \quad (2.6)$$

$${}^A T_B = \begin{bmatrix} {}^A R_B & {}^A P_{Borig} \\ 0 & 1 \end{bmatrix} \quad (2.7)$$

A equação (2.6) representa a movimentação, seja em termos de posição, seja em termos de orientação, entre quaisquer dois sistemas coordenados genéricos, locais ou inerciais [16].

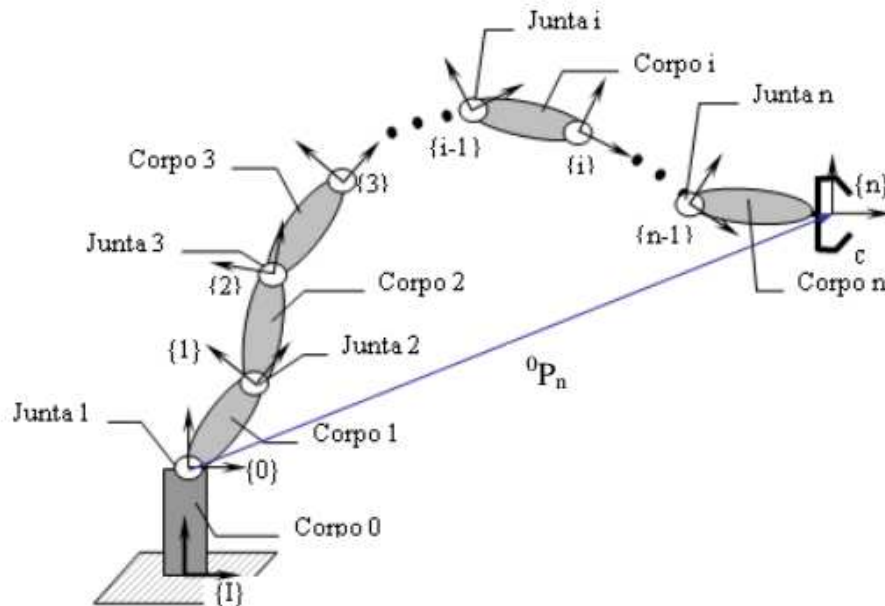


Figura 7 - Representação algébrica de um manipulador serial composto por n elos [16].

Para uma cadeia cinemática com n corpos, como um manipulador serial com n elos, a matriz de transformação homogênea do órgão terminal pode ser obtida como demonstrado na equação (2.8).

$${}^0T_n = {}^0T_1 {}^1T_2 \dots {}^{n-1}T_n \quad (2.8)$$

No próximo tópico será demonstrado o procedimento a ser seguido para obtenção das matrizes de transformação de um manipulador mecânico.

2.3. Cinemática Direta

2.3.1. Método de Denavit-Hatenberg

Em 1955 Jacques Denavit e Richard Hartenberg sistematizaram um método capaz de relacionar as translações e rotações entre elos adjacentes através de uma matriz de transformação homogênea 4x4, associando cada elo a um sistema de coordenadas local fixo no elo em questão e tornando possível a transformação do elo i desejado em relação ao referencial inercial de base através de transformações sequenciais entre os elos consecutivos.

As juntas do manipulador conectam dois elos adjacentes, os elos $i - 1$ e i , de maneira que quando a junta i é atuada o elo i se move enquanto o elo $i - 1$ permanece parado. Um manipulador com n juntas, numeradas de 1 a n , possuirá $n + 1$ elos, numerados de 0 a n a partir de base. Cada uma das juntas possui apenas um grau de liberdade, podendo ser de revolução ou prismática, de maneira que seus deslocamentos são descritos, respectivamente, por um ângulo de revolução ou por um deslocamento [6].

Dado um elo rígido i , haverão duas dimensões que caracterizam uma relação fixa entre o elo e as juntas adjacentes a ele, que são o comprimento a_i do elo e seu ângulo de twist α_i . Em relação a cadeia cinemática haverão outras duas dimensões que

relacionam o elo i com o elo $i - 1$ da cadeia, a distância d_i entre os elos e o ângulo θ_i entre eles.

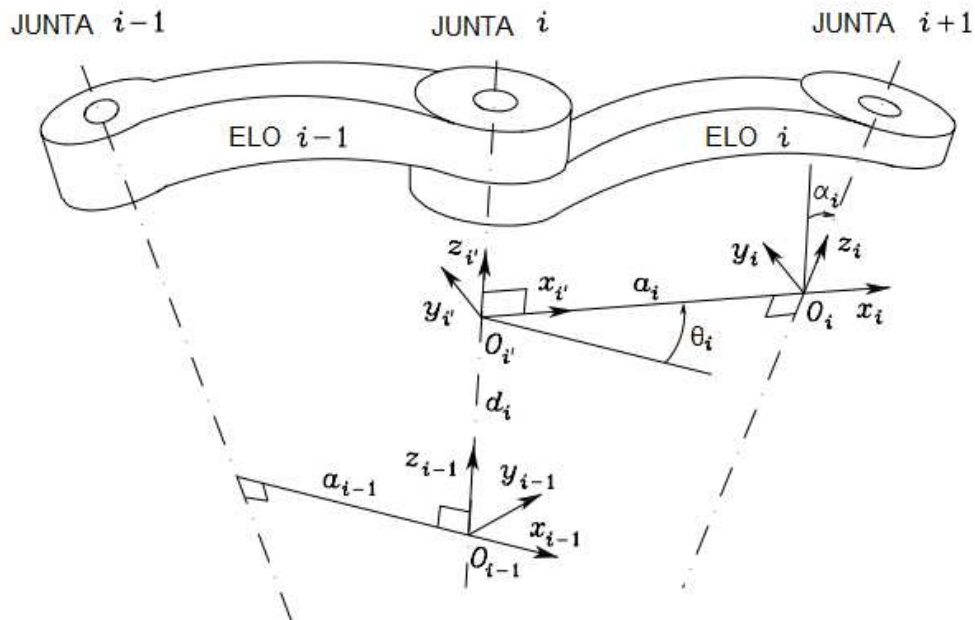


Figura 8 - Descrição de elos adjacentes como parte de uma cadeia cinemática, adaptado de [13].

A matriz de transformação homogênea ${}^{i-1}A_i$ é função das dimensões a_i , α_i , d_i , e θ_i , e representa a posição e orientação do sistema coordenado $o_i x_i y_i z_i$ em relação ao sistema $o_{i-1} x_{i-1} y_{i-1} z_{i-1}$. Se tratando das juntas de revolução o parâmetro θ_i será variável, e no caso das juntas prismáticas o parâmetro d_i que será variável. Assim, a matriz ${}^{i-1}A_i$ não será constante, sendo função de uma variável de junta (representada por q_i) que pode ser uma distância d_i ou um ângulo θ_i , dado que cada junta possui apenas um grau de liberdade.

A matriz de transformação ${}^i T_n$ que determina a postura do sistema de coordenadas $o_n x_n y_n z_n$ em relação ao sistema $o_i x_i y_i z_i$ é calculada através da equação (2.9).

$${}^i T_n = {}^i A_{i+1} {}^{i+1} A_{i+2} \dots {}^{n-2} A_{n-1} {}^{n-1} A_n \quad (2.9)$$

Assim, a posição e orientação de um efetuador de um manipulador robótico com n graus de liberdade será determinada pela matriz 0T_n , tal que:

$${}^0T_n = \begin{bmatrix} & {}^0R_n & & {}^0P_{norig} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

A matriz de transformação homogênea ${}^{i-1}A_i$ pode ser obtida através do produto de 4 transformações básicas que são função de um dos 4 parâmetros propostos por Denavit-Hartenberg:

$$\begin{aligned} {}^{i-1}A_i &= Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} = \\ &= \begin{bmatrix} C_{\theta_i} & -S_{\theta_i} & 0 & 0 \\ S_{\theta_i} & C_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_{\alpha_i} & -S_{\alpha_i} & 0 \\ 0 & S_{\alpha_i} & C_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.11) \\ &= \begin{bmatrix} C_{\theta_i} & -S_{\theta_i}C_{\alpha_i} & S_{\theta_i}S_{\alpha_i} & a_iC_{\theta_i} \\ S_{\theta_i} & C_{\theta_i}C_{\alpha_i} & -C_{\theta_i}S_{\alpha_i} & a_iS_{\theta_i} \\ 0 & S_{\alpha_i} & C_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

Onde,

$$\begin{aligned} C_{\theta_i} &= \cos(\theta_i) \\ S_{\theta_i} &= \sin(\theta_i) \\ C_{\alpha_i} &= \cos(\alpha_i) \\ S_{\alpha_i} &= \sin(\alpha_i) \end{aligned} \quad (2.12)$$

É possível determinar a cinemática direta de qualquer manipulador robótico utilizando-se o método proposto por Denavit-Hartenberg. Assim, os passos a baixo, resumidos por ORIOLO *et al.* [13], servem como um guia na determinação dos sistemas de coordenadas locais e global de um manipulador genérico e na definição dos parâmetros a_i, α_i, d_i , e θ_i relacionados ao elo e a junta i .

1. Defina as direções dos eixos z_0 até z_{n-1} . A direção do eixo z_i deve coincidir com a direção do eixo de atuação da junta $i + 1$.
2. Determine a origem o_0 do referêcia de base deve ao longo do eixo z_0 e os eixos x_0 e y_0 são determinados segundo a regra da mão direita.

3. Identifique a interseção entre o eixo z_i e a normal comum aos eixos z_{i-1} e z_i e localize a origem o_i nesse ponto. Se os eixos z_{i-1} e z_i forem paralelos e a junta i for de revolução localizar o_i de maneira que d_i seja igual a 0. Todavia, se a junta for prismática, localize o_i em uma posição de referência para o intervalo de operação da junta, por exemplo, no seu limite mecânico.
4. Defina o eixo x_i ao longo da normal comum aos eixos z_{i-1} e z_i apontando da junta i para a junta $i + 1$.
5. Obtenha a direção e orientação do eixo y_i segundo a regra da mão direita.
6. Estabeleça a origem $o_n x_n y_n z_n$ alinhando o eixo z_n com o eixo z_{n-1} se a junta n for de revolução ou escolhendo z_n de maneira arbitrária caso contrário. Siga os passos 4 e 5 para determinação dos eixos x_n e y_n , respectivamente.
7. Monte uma tabela com os parâmetros a_i , α_i , d_i , e θ_i dos elos, para $i = 1, \dots, n$, onde:
 - **a_i (comprimento do elo i)** - é a distância entre os eixos z_{i-1} e z_i medida ao longo do eixo x_i .
 - **α_i (ângulo de torção do elo i)** - é o ângulo entre os eixos z_{i-1} e z_i medido ao longo do eixo x_i .
 - **d_i (distância entre elos adjacentes)** - é a distância entre os eixos x_{i-1} e x_i medida ao longo do eixo z_{i-1} .
 - **θ_i (ângulo entre elos adjacentes)** - é o ângulo entre os eixos x_{i-1} e x_i medida ao longo do eixo z_{i-1} .
8. A partir da equação (2.11) calcule as matrizes de transformação ${}^{i-1}A_i$, para $i = 1, \dots, n$.
9. Calcule a matriz de transformação homogênea que determina a posição e orientação do órgão terminal do manipulador mecânico no sistema de coordenadas de base utilizando a equação (2.9).

A seguir será demonstrada a aplicação do método de Denavit-Hartenberg para o caso prático de um manipulador com 3 graus de liberdade.

2.3.2. Solução da cinemática direta de um manipulador com 3 graus de liberdade

Abaixo se demonstra a solução da cinemática direta do braço planar composto por 3 juntas rotativas (3 graus de liberdade) e representado na Figura 9, a partir do método proposto por Denavit-Hartenberg.

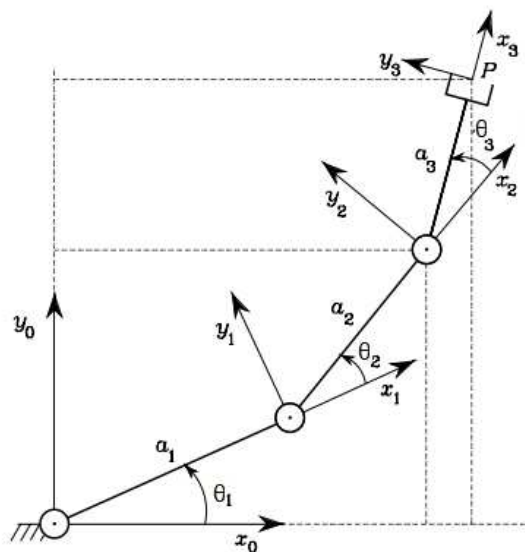


Figura 9 - Manipulador serial planar com 3 graus de liberdade, adaptado de [13].

Estabelecidos os eixos coordenados locais e globais, Figura 9, definem-se os parâmetros de Denavit-Hartenberg, demonstrados na Tabela 1:

Tabela 1 - Parâmetros de DH para o manipulador planar de 3 juntas

| ELO | a_i | α_i | d_i | θ_i |
|-----|-------|------------|-------|------------|
| 1 | a_1 | 0 | 0 | θ_1 |
| 2 | a_2 | 0 | 0 | θ_2 |
| 3 | a_3 | 0 | 0 | θ_3 |

A partir da equação (2.11) calcula-se as matrizes de transformação ${}^{i-1}A_i$:

$${}^0A_1 = \begin{bmatrix} c_1 & -s_1 & 0 & a_1c_1 \\ s_1 & c_1 & 0 & a_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.13)$$

$${}^1A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2c_2 \\ s_2 & c_2 & 0 & a_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.14)$$

$${}^2A_3 = \begin{bmatrix} c_3 & -s_3 & 0 & a_3c_3 \\ s_3 & c_3 & 0 & a_3s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.15)$$

E utilizando-se a equação (2.9) calcula-se a matriz de transformação homogênea que computa a cinemática direta do braço planar exemplificado:

$${}^0T_3 = {}^0A_1 {}^1A_2 {}^2A_3 = \begin{bmatrix} c_{123} & -s_{123} & 0 & a_1c_1 + a_2c_{12} + a_3c_{123} \\ s_{123} & c_{123} & 0 & a_1s_1 + a_2s_{12} + a_3s_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.16)$$

Onde:

$$\begin{aligned} c_i &= \cos(\theta_i) \\ s_i &= \text{sen}(\theta_i) \\ c_{ij} &= \cos(\theta_i + \theta_j) \\ s_{ij} &= \text{sen}(\theta_i + \theta_j) \\ c_{ijk} &= \cos(\theta_i + \theta_j + \theta_k) \\ s_{ijk} &= \text{sen}(\theta_i + \theta_j + \theta_k) \end{aligned} \quad (2.17)$$

2.4. Cinemática Inversa

Como discutido, o problema cinemático inverso constitui-se em se determinar as variáveis de junta do manipulador conhecidas a posição e orientação desejadas para o órgão terminal.

Neste t3pico ser3 apresentada a solu33o alg3brica da cinem3tica inversa e as condi33es nas quais ela 3 aplic3vel. Para fins de exemplifica33o ser3 usado o manipulador planar, Figura 9, cuja cinem3tica direta foi resolvida no t3pico anterior.

Dada a matriz ${}^0T_n^d$ que representa a postura desejada para o efetuador do manipulador e a matriz 0T_n obtida a partir do modelo cinem3tico direto, a solu33o da cinem3tica inversa do manipulador inicia-se igualando-se as duas matrizes, o que resulta na obten33o de 16 equa33es relacionadas as vari3veis de junta. A por33o da matriz de transforma33o correspondente a transla33o fornece 3 equa33es linearmente independentes, E, a partir da por33o da matriz de transforma33o correspondente a rota33o, obt3m-se 9 equa33es acopladas. Al3m disso, as 4 equa33es correspondentes a 3ltima linha da matriz de transforma33o s3o triviais.

2.4.1. M3todo Alg3brico

Para o manipulador serial planar discutido, sejam:

$${}^0T_3 = \begin{bmatrix} c_{123} & -s_{123} & 0 & a_1c_1 + a_2c_{12} + a_3c_{123} \\ s_{123} & c_{123} & 0 & a_1s_1 + a_2s_{12} + a_3s_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.16)$$

$${}^0T_3^d = \begin{bmatrix} c_\emptyset & -s_\emptyset & 0 & x \\ s_\emptyset & c_\emptyset & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.18)$$

Igualando-se as duas matrizes se obt3m o sistema de equa33es:

$$c_\emptyset = c_{123} \quad (2.19)$$

$$s_\emptyset = s_{123} \quad (2.20)$$

$$x = a_1c_1 + a_2c_{12} + a_3c_{123} \quad (2.21)$$

$$y = a_1s_1 + a_2s_{12} + a_3s_{123} \quad (2.22)$$

Substituindo-se as expressões (2.19) e (2.20) nas equações (2.21) e (2.22) e rearranjando-se as equações resultantes para que as incógnitas fiquem do lado esquerdo das equações temos:

$$x - a_3 c_\emptyset = a_1 c_1 + a_2 c_{12} \quad (2.23)$$

$$y - a_3 s_\emptyset = a_1 s_1 + a_2 s_{12} \quad (2.24)$$

Então, elevando-se as equações (2.23) e (2.24) ao quadrado e somando-as obtém-se:

$$(x - a_3 c_\emptyset)^2 + (y - a_3 s_\emptyset)^2 = a_1^2 + a_2^2 + 2a_1 a_2 c_2 \quad (2.25)$$

E assim:

$$c_2 = \frac{(x - a_3 c_\emptyset)^2 + (y - a_3 s_\emptyset)^2 - a_1^2 - a_2^2}{2a_1 a_2} \quad (2.26)$$

$$s_2 = \pm \sqrt{1 - c_2^2} \quad (2.27)$$

$$\theta_2 = \text{atan2}(s_2, c_2) \quad (2.28)$$

Uma vez que θ_2 é conhecido, torna-se vantajoso reescrever-se as equações (2.23) e (2.24) conforme abaixo:

$$x - a_3 c_\emptyset = a_1 c_1 + a_2 c_1 c_2 - a_2 s_1 s_2 \quad (2.29)$$

$$y - a_3 s_\emptyset = a_1 s_1 + a_2 s_1 c_2 + a_2 c_1 s_2 \quad (2.30)$$

Fazendo,

$$\begin{aligned} K_1 &= a_1 + a_2 c_2 \\ K_2 &= a_2 s_2 \end{aligned} \quad (2.31)$$

As equações (2.29) e (2.30) podem ser reescritas na forma:

$$x - a_3 C_\emptyset = K_1 c_1 + K_2 s_2 \quad (2.32)$$

$$y - a_3 S_\emptyset = K_1 s_1 + K_2 c_1 \quad (2.33)$$

Através das mudanças de variáveis a seguir,

$$\begin{aligned} r &= \sqrt{K_1^2 + K_2^2} \\ \gamma &= \text{atan2}(K_2, K_1) \end{aligned} \quad (2.34)$$

Tem-se

$$K_1 = r \cos(\gamma) \quad (2.35)$$

$$K_2 = r \text{sen}(\gamma) \quad (2.36)$$

Substituindo-se as transformações (2.35) e (2.36) nas equações (2.32 e (2.33), obtêm-se

$$x - a_3 c_\emptyset = r c_\gamma c_1 - r s_\gamma s_1 \quad (2.37)$$

$$y - a_3 s_\emptyset = r c_\gamma s_1 - r s_\gamma c_1 \quad (2.38)$$

As equações (2.37) e (2.38) podem ser reescritas como:

$$\frac{x - a_3 c_\emptyset}{r} = \cos(\gamma + \theta_1) \quad (2.39)$$

$$\frac{y - a_3 s_\emptyset}{r} = \text{sen}(\gamma + \theta_1) \quad (2.40)$$

Das equações (2.39) e (2.40) obtém-se o valor de θ_1 :

$$\theta_1 + \gamma = \text{atan2}\left(\frac{y - a_3 s_\emptyset}{r}, \frac{x - a_3 c_\emptyset}{r}\right) = \text{atan2}(y - a_3 s_\emptyset, x - a_3 c_\emptyset) \quad (2.41)$$

Substituindo γ , e isolando θ_1 obtém-se:

$$\theta_1 = \text{atan2}(y - a_3 s_\phi, x - a_3 c_\phi) - \text{atan2}(a_2 s_2, a_1 + a_2 c_2) \quad (2.42)$$

Das equações 2.19 e 2.20 sabe-se que:

$$\text{atan2}(s_\phi, c_\phi) = \text{atan2}(s_{123}, c_{123}) = \theta_1 + \theta_2 + \theta_3 \quad (2.43)$$

Então

$$\theta_3 = \text{atan2}(s_\phi, c_\phi) - \theta_1 - \theta_2 \quad (2.44)$$

Como demonstrado no exemplo acima, muitas vezes a solução algébrica da cinemática inversa de um robô manipulador demanda o conhecimento de diversas ferramentas matemáticas, sem as quais a solução do sistema de equações não seria possível. A complexidade da solução dessas equações se dá pelo fato de tratarem-se de equações não lineares.

O problema cinemático pode ser resolvido através de outros métodos além do algébrico, como por exemplo o método geométrico, mas a solução continua não sendo trivial, mesmo para um manipulador com apenas 3 graus de liberdade. Além disso, para se determinar a solução geral para manipuladores com 4, 5 ou 6 graus de liberdade é necessário que os manipuladores estudados apresentem um configuração geométrica específica, como demonstrado por Pieper [14]. Essa configuração e a solução proposta serão discutidas no próximo tópico.

2.4.2. Solução de Pieper

Foi demonstrado que a solução da cinemática direta de robôs manipuladores utilizando-se o método de Denavit-Hartenberg, garante a existência de 12 equações, 3

independentes e 9 acopladas, a serem utilizadas na solução da cinemática inversa do mesmo.

Na prática, como demonstrado por Pieper [14] [2], a álgebra é ferramenta suficiente para solução da cinemática inversa de um robô manipulador que possua entre 4 e 6 graus de liberdade se apenas 3 das suas juntas estão dispostas de forma independente, e as outras 1, 2 ou 3 juntas (para robôs com 4, 5 ou 6 graus de liberdade, respectivamente) obedecem uma configuração no espaço onde seus eixos de atuação se interceptam em um mesmo ponto.

Na indústria é comum que até 3 juntas sejam utilizadas para determinação da localização do efetuador no espaço, enquanto as últimas 3 juntas, obedecendo o arranjo espacial citado anteriormente, formando uma estrutura chamada de punho esférico, Figura 10, responsável pelo controle da orientação do efetuador. Exemplos dessa configuração são os manipuladores Puma, Kuka e Stanford.

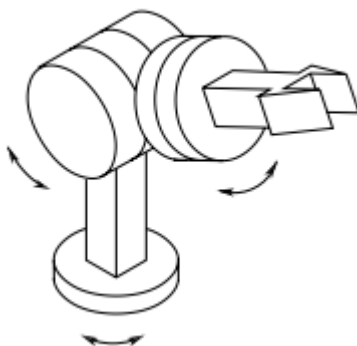


Figura 10 - Pulso esférico, retirado de [13].

A solução apresentada por Pieper para manipuladores que obedecem a configuração em questão consiste no desacoplamento do problema cinemático inverso em dois problemas, o problema de posição inverso e o problema de orientação inverso.

Assim, como feito anteriormente, iguala-se a matriz de transformação homogênea desejada e a matriz correspondente a cinemática direta do manipulador.

$${}^0T_n^d = {}^0T_n \quad (2.45)$$

A expressão (2.45) é então desmembrada em duas equações, a primeira correspondente as especificações de posição, equação (2.46) e a segunda as especificações de orientação, equação (2.47):

$${}^0P_n^d = {}^0P_n \quad (2.46)$$

$${}^0R_n^d = {}^0R_n \quad (2.47)$$

Se os eixos das juntas $n - 2$, $n - 1$ e n se interceptam no ponto Q e os sistemas de coordenadas foram atribuídos segundo o convencionalizado por Denavit-Hartenberg, então as origens $\{n - 1\}$ e $\{n - 2\}$ estarão nesse ponto, o que significa que o movimento das juntas $n - 2$, $n - 1$ e n não modificará a posição do ponto Q no espaço. E, uma vez que a origem do sistema $\{n\}$ é apenas uma translação ao longo de \hat{Z}_{n-1} , de valor d_n a partir de Q e \hat{Z}_{n-1} e \hat{Z}_n estão alinhados, tem-se:

$${}^0P_{norg} = {}^0Q + {}^0R_{n-1}d_n {}^{n-1}\hat{Z}_{n-1} = {}^0Q + {}^0R_n d_n {}^n\hat{Z}_n \quad (2.48)$$

Para se posicionar o efetuador no manipulador no ponto ${}^0P_n^d$ desejado, faz-se:

$${}^0Q = {}^0P_n^d - {}^0R_n^d d_n {}^n\hat{Z}_n \quad (2.49)$$

Além disso,

$${}^0Q = {}^0P_{(n-2)org} = {}^0T_{n-3} {}^{n-3}P_{(n-2)org} \quad (2.50)$$

Assim, pode-se determinar $q_1 \dots q_{n-3}$ igualando-se das expressões (2.49) e (2.50) que são função apenas de $q_1 \dots q_{n-3}$.

$${}^0P_n^d - {}^0R_n^d d_n {}^n\hat{Z}_n = {}^0T_{n-3} {}^{n-3}P_{(n-2)org} \quad (2.51)$$

Ao determinarem-se os valores $q_1 \dots q_{n-3}$, a matriz de rotação ${}^0R_{n-3}$ está determinada, e como:

$${}^0R_n^d = {}^0R_{n-3} {}^{n-3}R_n \quad (2.52)$$

Então a matriz ${}^{n-3}R_n$ pode ser calculada conforme a seguir:

$${}^{n-3}R_n = {}^0R_{n-3}^T {}^0R_n^d \quad (2.53)$$

Quando os métodos algébricos demonstrados não são suficientes para a solução da cinemática inversa de um manipulador, no caso de manipuladores cuja configuração não atenda aos requisitos discutidos, por exemplo, possuindo mais de 6 variáveis de juntas a serem determinadas, existem métodos numéricos alternativos, cuja viabilidade de uso pode ser estudada para solução do problema em questão. As desvantagens desses métodos são que eles podem demandar muito tempo na busca de uma solução satisfatória, além de nem sempre haver uma garantia de convergência dos mesmos. Ainda assim, robôs redundantes são de interesse no que diz respeito a maior versatilidade, por exemplo, desviando de obstáculos ou evitando configurações singularidades (nas quais a perda de mobilidade), desde que possam ser controlados de forma satisfatória.

3. REDES NEURAIS ARTIFICIAIS

3.1. Introdução

Redes neurais artificiais (RNAs) são estruturas que tentam reproduzir as capacidades de aprendizado e generalização características do cérebro humano [10], e podem ser utilizadas na solução de problemas de classificação e categorização, processamento de sinais, previsões e controle, entre outros.

Elas são compostas por unidades de processamento simples, chamadas nodos, nós ou neurônios, baseadas nos neurônios biológicos. Essas unidades estão distribuídas em uma ou mais camadas intermediárias, além das camadas de entrada e saída de dados, e são interligadas por conexões representadas pela sua força de ligação, ou pesos sinápticos [6] [1].

As redes neurais artificiais são capazes de mapear funções aprendendo através da generalização dos exemplos recebidos e então fornecendo respostas coerentes para dados não conhecidos. Na Figura 11 está representado o exemplo de uma arquitetura genérica de uma RNA.

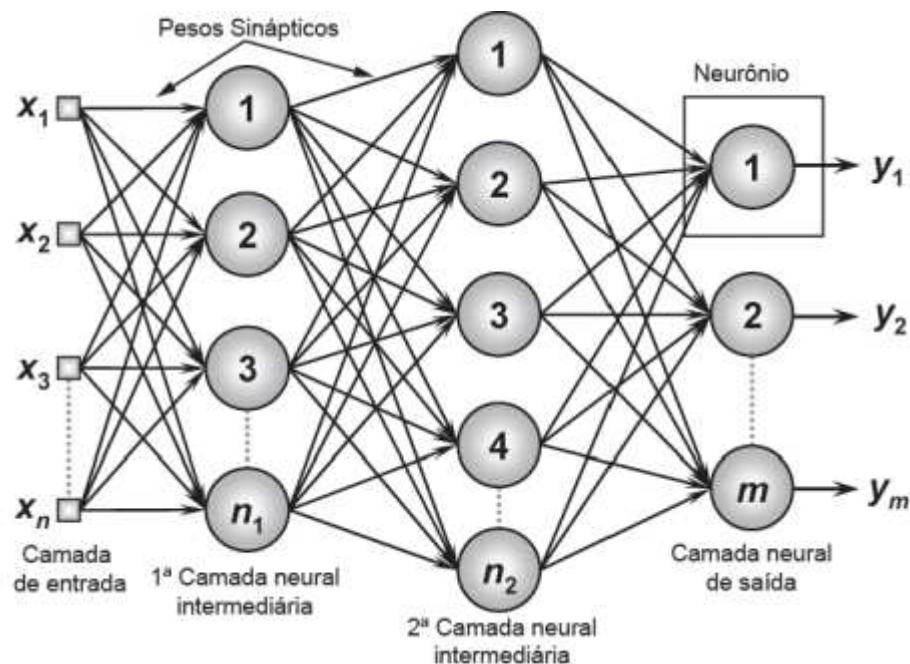


Figura 11 – Exemplo de arquitetura de uma RNA, retirado de [11].

3.2. Neurônio Biológico

O cérebro humano possui em torno de 10^{11} neurônios, sua célula fundamental. Os neurônios são divididos em 3 seções fundamentais: os dendritos, o corpo celular e o axônio, Figura 12.

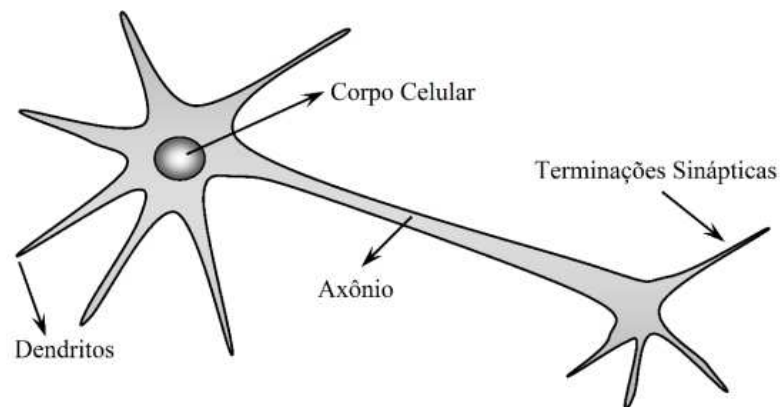


Figura 12 – Neurônio biológico, retirado de [11].

Os neurônios unem-se através das sinapses, que são o ponto de contato entre o axônio de um neurônio e o dendrito de outro, formando as redes neurais biológicas.

Os dendritos recebem as informações fornecidas por neurônios pré-sinápticos na forma de sinais químicos e as conduzem até o corpo celular onde essa informação é processada, gerando impulsos elétricos que são transmitidos através do axônio do neurônio até os dendritos dos próximos neurônios da rede neural biológica.

As sinapses são capazes de controlar a transmissão dos impulsos entre os neurônios da rede neural e é dessa habilidade que surge a capacidade de adaptação dos neurônios e da rede como um todo. Os sinais recebidos por um neurônio através de suas sinapses com outros neurônios da rede neural são combinados e comparados com um valor de limiar de excitação do neurônio. Se em um determinado intervalo de tempo o valor for suficientemente alto o neurônio “dispara”, transmitindo um impulso para os neurônios seguintes. A capacidade do cérebro de realizar operações

complexas decorre da operação em paralelo dos 10^{11} neurônios constituintes da rede neural biológica.

A membrana dos neurônios pode produzir e transmitir sinais químicos, transmitidos através das sinapses, e sinais elétricos, gerados dentro dos neurônios. Há um potencial elétrico de -70 mV (potencial de repouso) dentro das células nervosas em relação ao exterior graças a uma concentração de íons de potássio no seu interior. Para que a célula dispare produzindo um potencial de ação (impulso nervoso), é preciso que os impulsos da sinapses reduzam esse nível para cerca de -50 mV. O impulso nervoso disparado pelo neurônio se propaga pelo axônio e libera moléculas neurotransmissoras que determinarão a polarização ou despolarização do corpo do neurônio pós-sináptico dependendo do tipo de sinapse, inibitória ou excitatória. Assim, o disparo de um neurônio será determinado pelo acúmulo de entradas inibitórias e excitatórias medidas pelo corpo da célula em um curto intervalo de tempo.

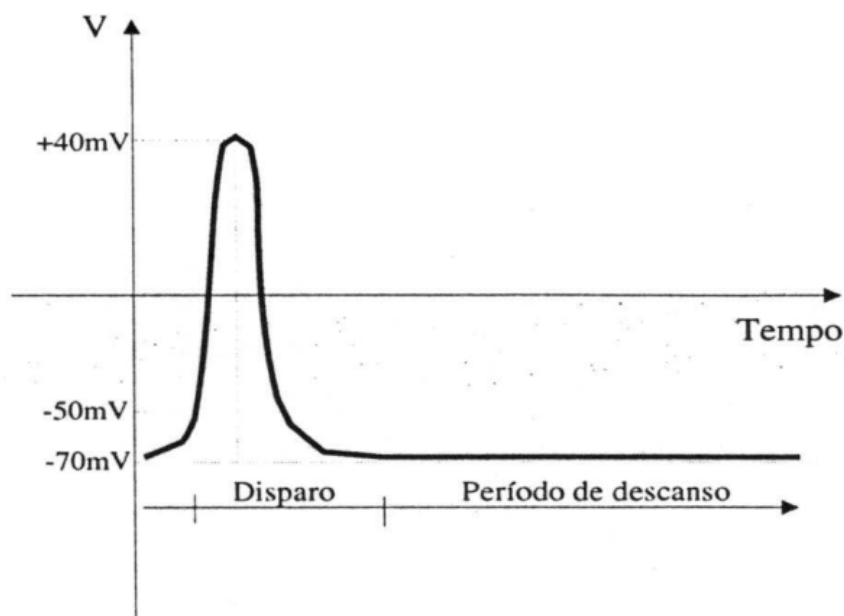


Figura 13 - Potencial de ação de um neurônio, retirado de [1].

Após gerar um impulso o neurônio entra em um período de refração, onde não pode ser estimulado novamente, enquanto retorna para seu potencial de repouso e se prepara para geração de um novo impulso, Figura 13 [1].

3.3. Neurônio Artificial: Modelo MCP

Em 1943 foi criado o primeiro modelo artificial para os neurônios biológicos, conhecido como modelo MCP, em homenagem a seus criadores, Warren McCulloch e Walter Pitts. O modelo MCP é uma simplificação matemática do que se sabia sobre neurônios biológicos na época.

Nele os dendritos são representados por n terminais de entrada x_1, x_2, \dots, x_n encarregados de receber as entradas, simbolizando os impulsos enviados por outros neurônios, e um terminal de saída y representa o axônio. Os terminais de entrada então associados a pesos w_1, w_2, \dots, w_n , com valores positivos ou negativos, que emulam as sinápticas nos neurônios biológicos, sinalizando o quanto o neurônio deve considerar sinais de disparo que ocorram naquela ligação.

O efeito de uma sinapse i sobre o neurônio pós-sináptico k é definido como $w_i x_i$, de maneira que o neurônio terá sua saída ativada apenas quando o somatório das entradas ponderadas por seus respectivos pesos for maior ou igual ao limiar θ (chamado de *threshold*), como demonstrado na equação (3.1).

$$\sum_{i=1}^n w_i x_i \geq \theta \quad (3.1)$$

Sejam as entradas de um neurônio representadas por um vetor $\mathbf{x} = (x_1, \dots, x_n)^T$, seu respectivos pesos representados por $\mathbf{w} = (w_1, \dots, w_n)^T$ e o escalar b , chamado de bias, definido por $b = -\theta$. A saída y do neurônio MCP em questão será o resultado da aplicação de uma função de propagação f sobre o resultado de uma função *net*, chamada de função de ativação, tal que:

$$y = f(net) \quad (3.2)$$

Onde,

$$net = b + \mathbf{w}^T \mathbf{x} \quad (3.3)$$

E,

$$f(z) = \begin{cases} 1 & \text{se } z \geq 0 \\ 0 & \text{se } z < 0 \end{cases} \quad (3.4)$$

Se observa que o modelo de neurônio proposto por McCulloch e Pitts não possuía pesos ajustáveis, não apresentando, portanto, propriedades adaptativas. Na Figura 14 encontra-se uma representação do neurônio MCP.

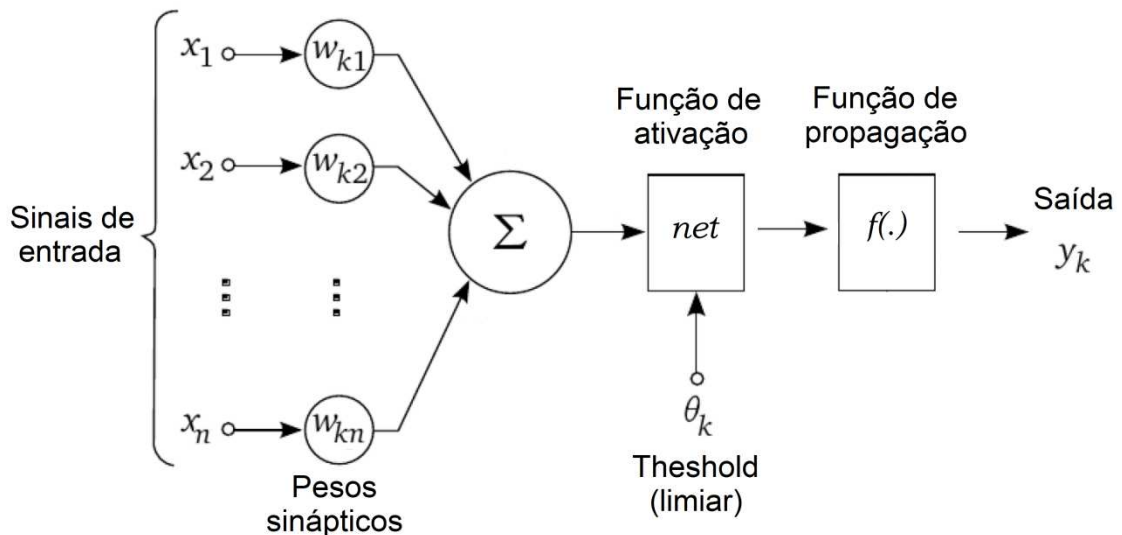


Figura 14 - Neurônio de McCulloch e Pitts, adaptado de [17].

3.4. Redes Perceptron

Frank Rosenblatt propôs, em 1958, a criação de uma rede neural artificial composta por vários neurônios MCP. O modelo ficou conhecido como Perceptron e, diferentemente do proposto por McCulloch e Pitts, a rede fazia uso de uma regra de aprendizado para que seus pesos fossem ajustado aumentando o desempenho da mesma [11] [10].

A rede proposta por Rosenblatt possuía 3 níveis: o primeiro nível era responsável apenas por obter as entradas a serem processadas pela rede de forma que não

possuía pesos atrelados a ele; o segundo nível era composto por neurônios MCP sem pesos ajustáveis, que funcionavam como unidades de associação; e o terceiro nível era composto por neurônios MCP com pesos ajustáveis, o que conferia ao terceiro nível propriedades adaptativas, sendo esses neurônios eram as unidades de resposta da rede. A topologia do modelo Perceptron original está representada na Figura 15. Se considera que esse modelo possuía apenas uma camada, uma vez que apenas os neurônios da última camada possuíam pesos ajustáveis.

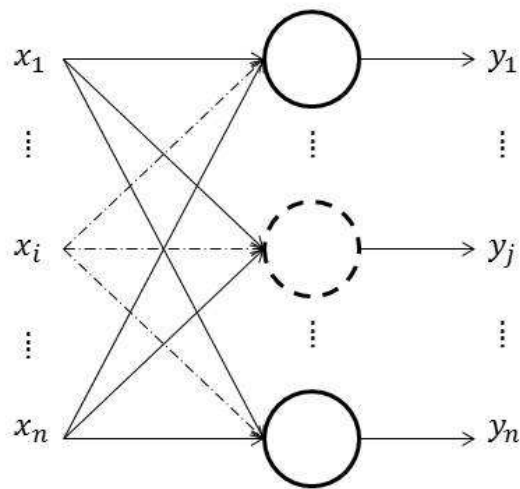


Figura 15 - Rede Perceptron

Sejam x e w^k , respectivamente, o vetor de entradas e o vetor de pesos na k -ésima iteração do treinamento (com $k = 0, 1, \dots$) de um neurônio com n terminais de entrada e $b^{(k)}$ o bias desse neurônio. E sendo $w^{(0)}$ e $b^{(0)}$ definidos de forma aleatória, o treinamento da rede neural consiste em se calcular o incremento no vetor peso (Δw^k) e o incremento no bias (Δb^k) visando o aprendizagem da entrada x , ou seja:

$$w^{k+1} = w^k + \Delta w^k \quad (3.5)$$

$$b^{k+1} = b^k + \Delta b^k \quad (3.6)$$

O erro e^k do neurônio na k -ésima iteração do treinamento para o vetor de entradas x é dado pela expressão (3.7) abaixo:

$$e^k = d - y^k \quad (3.7)$$

Onde a saída y^k gerada pelo neurônio, função de x , w^k , e b^k , é calculada através da equação (3.2), e d é a saída esperada para a entrada. Então, demonstra-se em BRAGA *et al.* [1] que:

$$\Delta w^k = \eta e^k x \quad (3.8)$$

$$\Delta b^k = \eta e^k \quad (3.9)$$

Onde η é a chamada taxa de aprendizado da rede.

O método utilizado acima pode ser generalizado para uma rede neural com n terminais de entrada, m terminais de saída e um número p de exemplos de maneira que: $X_{n \times p}$ será a matriz de entrada dos exemplos, $W_{m \times n}^k$ a matriz dos pesos na k -ésima iteração do treinamento, $b_{m \times 1}^k$ os bias, $Y_{m \times p}^k$ as saídas geradas pela rede para as entradas fornecidas, $D_{m \times p}$ as saídas esperadas, $E_{m \times p}^k$ a matriz de erros para cada uma das saídas, $\Delta W_{m \times n}^k$ e $\Delta b_{m \times 1}^k$ os incrementos dos pesos e dos bias respectivamente, e $\mathbf{1}_{1 \times p}$ um vetor com todos os componentes iguais a 1. Então [10]:

$$\mathbf{net}^k = b^k \mathbf{1} + W^k X \quad (3.10)$$

$$Y^k = F(\mathbf{net}^k) \quad (3.11)$$

$$E^k = D - Y^k \quad (3.12)$$

$$\Delta W^k = \eta E^k X^T \quad (3.13)$$

$$\Delta b^k = \eta E^k \mathbf{1}^T \quad (3.14)$$

3.5. Redes MLP

Demonstra-se que existe a garantia de convergência para rede neurais com apenas uma camada, como a Perceptron, porém elas só são capazes de resolver problemas linearmente separáveis, o que restringe significativamente as possibilidades de utilização. Cybenko afirma que uma rede com uma camada intermediária pode implementar qualquer função contínua enquanto uma rede com duas camadas intermediárias permite a aproximação de qualquer função [1].

As redes Perceptron Multicamadas, ou MLP (de *Multi-Layer Perceptron*) apresentam-se como uma solução às limitações apresentadas pelas redes com apenas uma camada. Elas são compostas por mais de uma camada intermediária (ou camada escondida) com pesos ajustáveis, e pelo menos uma dessas camadas tem uma função de propagação não linear, Figura 16.

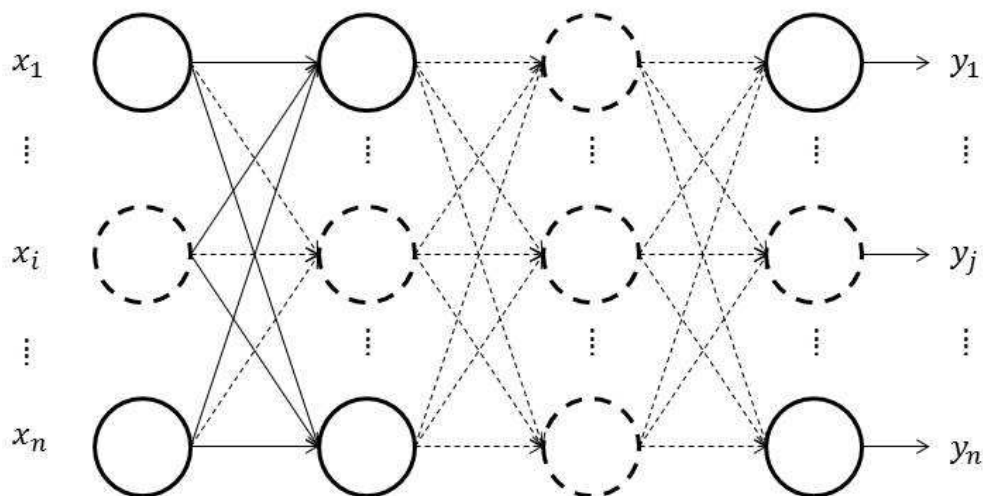


Figura 16 - Rede MLP.

Uma vez que o algoritmo de treinamento para redes Perceptron utiliza o erro em relação à saída esperada como parâmetro para atualização dos pesos, torna-se necessário calcular o erro das camadas intermediárias para redes com mais de uma camada escondida, e então o valor de atualização dos seus pesos. Trata-se, no

entanto, de uma tarefa difícil, já que não se conhecem as saídas esperadas para essas camadas.

Em 1986, Rumelhart propôs um algoritmo para o treinamento de redes MLP chamado *Back-Propagation*, dando um novo fôlego as pesquisas na área de RNAs, abandonadas por mais de 10 anos devido às limitações das redes Perceptron e a dificuldade de treinamento das redes MLP.

3.6. Algoritmo de *Back-propagation*

O algoritmo de *back-propagation* é composto de duas fases por ciclo: a fase *forward* e a fase *backward*. Na fase de *forward* a rede neural é percorrida no caminho convencional, da entrada para a saída, onde as saídas geradas pelos neurônios de cada uma das camadas são calculadas e utilizadas para alimentar a camada seguinte. Na fase de *backward* os pesos dos neurônios das camadas da RNA são atualizados. Essa fase recebe esse nome porque a atualização dos pesos é feita da camada de saída da rede para a camada de entrada. Na camada de saída utiliza-se o erro em relação às saídas desejada, e nas camadas intermediárias faz-se uma estimativa desse erro, como função do erro da camada seguinte.

Sendo p o número de exemplos disponíveis para o treinamento, n_L o número de neurônios da camada de saída da rede, \mathbf{T} uma matriz $n_L \times p$ com as saídas esperadas e \mathbf{Y}_L^k uma matriz $n_L \times p$ com as saídas calculadas pela rede na k -ésima iteração, então o erro médio quadrático, na k -ésima iteração de treinamento, é definido como o somatório dos quadrados das diferenças entre as saídas esperadas e as saídas geradas pela rede.

$$e^k = \sum_{i=1}^p \sum_{j=1}^{n_L} \left(T_{ji} - (Y_L^k)_{ji} \right)^2 \quad (3.15)$$

Uma superfície de erro é gerada segundo a equação (3.16).

$$S^k = \frac{(e^k)^2}{2} \quad (3.16)$$

O algoritmo de *back-propagation* percorre essa superfície buscando o ponto de mínimo através da técnica de gradiente descendente, que consiste em caminhar pela superfície utilizando a direção oposta à indicada pelo gradiente, assim o ajuste dos pesos deve ser proporcional a

$$\Delta \mathbf{w}^k \propto -\nabla S^k \quad (3.17)$$

Em BRAGA et al. [1] foi demonstrado que

$$\nabla S^k = -e^k \mathbf{x} \quad (3.18)$$

Então, retorna-se a expressão (3.8)

$$\Delta \mathbf{w}^k = \eta e^k \mathbf{x} \quad (3.8)$$

Assim, os cálculos realizados na etapa de *backward* do *back-propagation* são definidos em [10] como a seguir.

Sejam:

- $(L + 1)$ o número de camadas da rede (a camada 0 é a camada de entrada e a camada L é a camada de saída);
- n_l o número de neurônios na camada l ($0 \leq l \leq L$);
- p o número de exemplos;
- \mathbf{P} uma matriz $n_0 \times p$ com as entradas dos exemplos;

- \mathbf{T} uma matriz $n_L \times p$ com as saídas esperadas para as entradas \mathbf{P} ;
- \mathbf{X}_l^k uma matriz $n_{l-1} \times p$ com as entradas para os neurônios da camada l ($0 \leq l \leq L$) na k -ésima iteração do treinamento;
- \mathbf{Y}_l^k uma matriz $n_l \times p$ com as saídas dos neurônios da camada l ($0 \leq l \leq L$) na k -ésima iteração do treinamento;
- \mathbf{W}_l^k uma matriz $n_l \times n_{l-1}$ com os pesos dos neurônios da camada l ($0 \leq l \leq L$), onde $(\mathbf{W}_l^k)_{ij}$ é o peso relativo à j -ésima entrada do i -ésimo neurônio;
- b_l^k um vetor $n_l \times 1$ com o *bias* dos neurônios da camada l ($0 \leq l \leq L$) na k -ésima iteração do treinamento;
- $f_l: \mathbb{R} \rightarrow \mathbb{R}$ a função de propagação da camada l ($0 \leq l \leq L$);
- \mathbf{E}_l^k uma matriz $n_l \times p$ com os erros estimados para os neurônios da camada l ($0 \leq l \leq L$) na k -ésima iteração do treinamento;
- $\Delta \mathbf{W}_l^k$ uma matriz $n_l \times n_{l-1}$ com os incrementos dos pesos dos neurônios da camada l ($0 \leq l \leq L$),
- Δb_l^k um vetor $n_l \times 1$ com os incrementos dos *bias* dos neurônios da camada l ($0 \leq l \leq L$) na k -ésima iteração do treinamento;
- $\mathbf{1}$ um vetor $1 \times p$ onde todos os componentes têm valor 1.
-

Então:

$$\mathbf{X}_l^k = \mathbf{Y}_l^k, \quad 0 < l \leq L \quad (3.19)$$

$$\mathbf{net}_l^k = b_l^k \mathbf{1} + \mathbf{W}_l^k \mathbf{X}_l^k, \quad 0 < l \leq L \quad (3.20)$$

$$\mathbf{Y}_l^k = \begin{cases} \mathbf{P} & \text{se } l = 0 \\ \mathbf{F}_l(\mathbf{net}_l^k) & \text{se } 0 < l \leq L \end{cases} \quad (3.21)$$

$$\mathbf{E}_l^k = \begin{cases} \mathbf{F}'_l(\mathbf{net}_l^k) \bullet ((\mathbf{W}_{l+1}^k)^T \mathbf{E}_{l+1}^k) & \text{se } l = 0 \\ \mathbf{F}'_l(\mathbf{net}_l^k) \bullet (\mathbf{T} - \mathbf{Y}_L^k) & \text{se } 0 < l \leq L \end{cases} \quad (3.22)$$

$$\Delta \mathbf{W}_l^k = \eta \mathbf{E}_l^k (\mathbf{X}_l^k)^T, \quad 0 < l \leq L \quad (3.23)$$

$$\Delta b_l^k = \eta \mathbf{E}_l^k \mathbf{1}^T, \quad 0 < l \leq L \quad (3.24)$$

Onde:

$$(\mathbf{A} \cdot \mathbf{B})_{ij} = A_{ij} \times B_{ij} \quad (3.25)$$

$$(\mathbf{F}_l(\mathbf{Z}))_{ij} = f_l(\mathbf{Z}_{ij}), \quad 0 < l \leq L \quad (3.26)$$

$$(\mathbf{F}'_l(\mathbf{Z}))_{ij} = f'_l(\mathbf{Z}_{ij}), \quad 0 < l \leq L \quad (3.27)$$

3.6.1. *Back-propagation com momentum*

Existem várias propostas de alteração no algoritmo de *back-propagation* com o intuito de acelerar sua velocidade de convergência, considerada lenta, e que piora de maneira sensível conforme aumenta o grau de complexidade do problema a ser resolvido [1].

Dentre essas alterações citaremos o algoritmo de *back-propagation* com *momentum*, que será utilizado no treinamento das redes nesse trabalho. Essa proposta constitui-se numa alteração na fórmula de atualização dos pesos adicionando um termo de *momentum* nas mesmas afim de considerar no cálculo dos novos incrementos a serem utilizados os incrementos utilizados anteriormente:

$$\Delta \mathbf{W}_l^k = \alpha \Delta \mathbf{W}_l^{k-1} + (1 - \alpha)(\eta \mathbf{E}_l^k (\mathbf{X}_l^k)^T) \quad (3.28)$$

Onde, $k > 1$ e α ($0 < \alpha < 1$) é a chamada constante de *momentum*.

Essa técnica prevê ainda a utilização de uma taxa de aprendizado adaptativa, que significa o aumento da taxa de aprendizado para sucessivas diminuições do erro e diminuição da taxa de aprendizado caso contrário, mas que não será utilizada nesse trabalho.

Alguns dos critérios de parada utilizados durante o treinamento de uma rede neural são: um valor máximo de iterações, ou ciclos de treinamento, a serem executados; o valor do erro médio quadrático (parando o treinamento da rede quando seu erro médio quadrático é menor ou igual a um valor pré-estabelecido); ou ainda a

validação, onde um conjunto auxiliar de dados é utilizado para validar o processo de aprendizado da rede evitando o *overfitting*, problema no qual a rede passa a “decorar” as respostas para os exemplos de entrada, perdendo sua capacidade de generalização. Esse fenômeno pode ser percebido quando, no decorrer das iterações, o valor do erro quadrático médio da rede continua diminuindo para o conjunto de dados utilizados no treinamento, mas começa a aumentar para dados auxiliares, usados apenas para validação da rede.

3.7. Funções de Propagação

Abaixo são listadas algumas das funções de propagação mais comumente usadas no treinamento de redes neurais:

- Função Degrau

$$f(x) = \begin{cases} 1 & \text{se } x \geq 0; \\ 0 & \text{se } x < 0; \end{cases} \quad (3.29)$$

- Tangente hiperbólica logística

$$f(x) = \frac{2}{1 + e^{-2x}} - 1; \quad (3.30)$$

- Linear logística

$$f(x) = \begin{cases} -\log(1-x) & \text{se } x \leq 0 \\ x & \text{se } 0 < x \leq 1; \\ 1 + \log x & \text{se } x > 1 \end{cases} \quad (3.31)$$

- Sigmoides logística

$$f(x) = \frac{1}{1 + e^{-x}}; \quad (3.32)$$

- Linear pura

$$f(x) = x; \quad (3.33)$$

- Base radial

$$f(x) = e^{-x^2} \quad (3.34)$$

- Linear saturada

$$f(x) = \begin{cases} 0 & \text{se } x \leq 0 \\ x & \text{se } 0 < x \leq 1; \\ 1 & \text{se } x > 1 \end{cases} \quad (3.35)$$

- Base triangular

$$f(x) = \begin{cases} 1 - |x| & \text{se } -1 \leq x \leq 1 \\ 0 & \text{caso contrário} \end{cases} \quad (3.36)$$

Na Figura 17 encontram-se os gráficos das funções de propagação citadas com o valor de x compreendido no intervalo $[-2, 2]$.

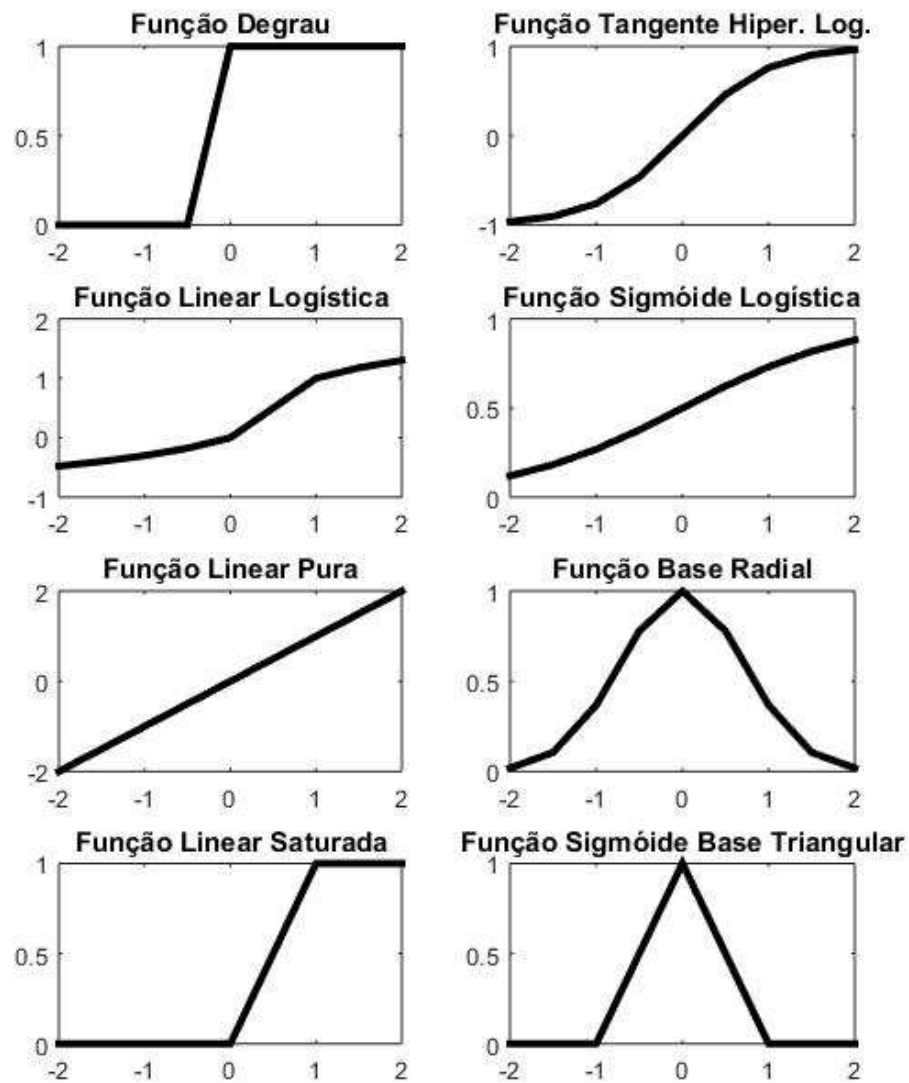


Figura 17 - Gráficos das funções de propagação.

4. ALGORITMOS GENÉTICOS

4.1. Introdução

O método de algoritmos genéticos é inspirado na teoria da evolução das espécies proposta por Darwin em 1858, que estabeleceu alguns princípios básicos para o processo de evolução das espécies [7]:

1. Indivíduos da mesma espécie ou de espécies diferentes disputam continuamente pelos recursos disponíveis no meio ambiente uma vez que esses recursos são limitados.
2. São classificados como os indivíduos mais adaptados ao ambiente aqueles que possuem maior probabilidade de sobrevivência devido a características próprias específicas.
3. Os indivíduos melhor adaptados ao ambiente possuem maior chance de reprodução já que possuem uma maior probabilidade de sobrevivência.
4. No processo de reprodução um grande número de características dos pais são repassadas aos filhos, assim, as características dos indivíduos mais adaptados, que se reproduzem mais, tendem a serem propagadas nas gerações seguintes da população.
5. Durante o processo de evolução as características mais favoráveis à sobrevivência da espécie tendem a ser propagadas aumentando o grau de adaptação das mesmas.
6. Durante a reprodução pode acontecer o fenômeno de mutação onde ocorrem falhas durante a replicação e transmissão das características dos pais para os filhos. Normalmente a mutação é um processo prejudicial ao indivíduo, mas algumas vezes pode incluir nele uma característica desejável não pertencente aos pais. A mutação permite que um número maior de combinações e possibilidades sejam exploradas.

Assim, a evolução das espécies pode ser considerada um processo de otimização ao longo do tempo, onde os indivíduos tornam-se cada vez mais adaptados ao ambiente onde habitam. Daí surgiu a ideia de mimetizar esse processo digitalmente para a otimização de problemas em geral.

Os primeiros trabalhos relacionados a algoritmos genéticos surgiram por volta de 1950. Contudo, foi em 1970, com uma série de trabalhos publicados por

pesquisadores liderados pelo professor John Holland na Universidade de Michigan, que seu desenvolvimento progrediu significativamente [8].

4.2. Conceitos Biológicos

As células dos seres vivos possuem cromossomos, que são estruturas compostas por cadeias de DNA que contém os genes, definidos na genética clássica como a unidade fundamental de hereditariedade. Os genes armazenam as características específicas de cada indivíduo, como a cor da pele e dos olhos, entre outras. O conjunto de genes de um indivíduo é chamado de genoma. No caso dos seres humanos, estima-se que o genoma seja composto por cerca de 25000 genes [12].

Os alelos são os diferentes valores que um gene pode assumir. A combinação de alelos formando os genes de um indivíduo dá-se o nome de genótipo, e a manifestação das características dos seres vivos no meio ambiente dá-se o nome de fenótipo, que é a combinação entre o genótipo do indivíduo e a influência do ambiente no mesmo.

Os seres que se reproduzem sexuadamente possuem células diploides, onde os cromossomos se organizam em pares de cromossomos homólogos de maneira que cada característica é definida por um número par de genes. Esse seres possuem células reprodutoras especializadas, os gametas, que possuem somente um cromossomo de cada tipo, sendo chamadas haploides [5].

Quando ocorre a fecundação acontece a fusão dos núcleos dos dois gametas dos pais do indivíduo, originando um filho com células diploides que herdam características de ambos os pais.

Durante o processo de produção dos gametas podem ocorrer ainda mutações, onde nesse caso o novo indivíduo gerado poderá apresentar características ausentes nos pais. As mutações contribuem para a diversidade dos seres aumentando a variabilidade genética entre eles.

A reprodução sexuada contribui para a adaptação das espécies ao ambiente já que os indivíduos melhor adaptados têm maiores chances de gerar filhos, pois tem maior chance de sobrevivência, e esses filhos carregam as características dos seus pais tendo uma alta probabilidade de serem tão bem adaptados quanto estes [10].

A seleção natural resulta no processo natural de permanência dos indivíduos melhor adaptados a determinado ambiente e eliminação dos indivíduos que apresentem características desvantajosas à sobrevivência. Junto com a reprodução sexuada, que garante a diversidade genética dos seres vivos, é responsável pela evolução das espécies.

4.3. Componentes

O método de algoritmos genéticos baseia-se nos processos de reprodução sexuada e seleção natural encontrados na biologia aproveitando-se dos conceitos inerentes dos mesmos, eles são robustos e altamente adaptáveis.

Os indivíduos são a unidade fundamental de um algoritmo genético, eles codificam possíveis soluções para o problema analisado, e a resposta ótima para o problema em questão é encontrada através da manipulação (utilizando-se operadores genéticos) desses indivíduos.

A população é o conjunto de indivíduos representantes de possíveis soluções do problema. Cada indivíduo é avaliado de maneira a ser determinada quão boa é a solução que ele representa em relação ao problema proposto. O resultado dessa avaliação determina o grau de adaptação do indivíduo ao ambiente. A correspondência feita entre os termos utilizados na biologia e seus significados em relação aos algoritmos genéticos está resumida na Tabela 2.

Tabela 2 – Correspondência: Biologia x Algoritmos Genéticos

| Biologia | Algoritmos Genéticos |
|-----------------|-----------------------------|
| Genótipo | Estrutura |

| | |
|-----------------------|--|
| Genes | Componentes da estrutura |
| Alelos | Valores que podem ser assumidos pelos componentes da estrutura |
| Fenótipo/indivíduo | Solução |
| Adaptação ao ambiente | Avaliação da solução |
| População | Conjunto de soluções |

4.3.1. Representação

A etapa de escolha de representação para os indivíduos é a etapa mais importante de um algoritmo genético. Nela acontece a tradução do problema a ser resolvido para uma linguagem que possa ser manipulada pelo mesmo, sendo que o desempenho do algoritmo está intrinsecamente ligado a utilização da representação adequada ao problema.

Dentre as diversas representações utilizadas nos cromossomos, as mais comuns são: representação binária e representação por inteiros [8]. Na Tabela 3 estão exemplificadas possíveis representações para alguns dos problemas que podem ser tratados através de algoritmos genéticos.

Tabela 3 – Exemplos de representações e seus correspondentes fenótipos para alguns problemas

| Problema | Alelo | Genótipo | Fenótipo |
|------------------------------------|------------------------------|----------|--|
| Otimização numérica | 0; 1 | 101000 | 10 |
| Problema do caixeiro viajante | A; B; C; D; E | CDAEB | Comece na cidade C, depois pelas cidades D, A e E e termine em B |
| Regras de aprendizado para agentes | C1; C2; C3; R1; R2; R3 | C1R3C3R2 | Se condição 1 execute regra 3, se condição 3 execute regra 2 |

4.4. Operadores Genéticos

Os operadores genéticos são regras baseadas nos conceitos biológicos apresentados na seção 4.2 que buscam renovar a população de maneira eficaz, ou

seja, melhorando a qualidade genética dos indivíduos presentes, e aproximando-se cada vez mais da solução ótima para o problema proposto [8].

Os operadores genéticos comumente utilizados são: inicialização, avaliação, seleção, cruzamento, mutação, atualização e finalização. Nessa seção o funcionamento de cada um deles será explicado.

4.4.1. Inicialização

A inicialização de um algoritmo genético consiste na criação de indivíduos (que representam possíveis soluções para o problema) para compor a população inicial. Dentre os métodos de inicialização utilizados durante a criação da população inicial encontram-se [7]:

- **Inicialização randômica uniforme:** onde os valores dos genes de cada indivíduo são definidos de forma aleatória e uniforme dentro do conjunto de alelos daquele gene.
- **Inicialização randômica não uniforme:** onde alguns dos possíveis valores do conjunto de alelos do gene têm maior probabilidade de serem escolhidos que outros.
- **Inicialização randômica com “dope”:** onde indivíduos otimizados são misturados a população gerada aleatoriamente.

4.4.2. Avaliação

Juntamente com a escolha da representação dos indivíduos, a avaliação é a etapa do algoritmo genético mais dependente do problema analisado. Nela deve estar armazenado todo o conhecimento prévio que existe sobre o problema, sejam suas restrições, sejam seus objetivos de qualidade [6][7].

A função de avaliação também pode ser chamada de função objetivo, para problemas de otimização de funções matemáticas o próprio valor de retorno delas

costumam ser escolhidos para avaliação do indivíduo [7], por exemplo, busca-se minimizar a função

$$f(x) = x^2 - 5x \quad (4.1)$$

O valor de retorno da função de avaliação $f_o(G_d)$ é:

$$f_o(G_d) = f_o(d(G_I)) \quad (4.2)$$

Onde $f_o(G_d)$ é a função de avaliação aplicada sobre o genoma decodificado do indivíduo avaliado (G_d), e $d(G_I)$ é a função que decodifica o genoma (G_I) do indivíduo, por exemplo:

$$G_I = [110]; \text{ (base binária)} \quad (4.3)$$

$$G_d = d(G_I) = 6 \quad (4.4)$$

$$f_o(G_d) = 6^2 - 5 \times 6 = 6 \quad (4.5)$$

4.4.3. Seleção

Na etapa de seleção são escolhidos os indivíduos da população que serão utilizados pelo operador de reprodução. A seleção busca preservar as boas características da espécie, simulando o mecanismo de seleção natural ao qual os organismos biológicos são expostos, baseando-se na avaliação de cada indivíduo da população de maneira que indivíduos melhor avaliados possuem maior probabilidade de serem selecionados [6][10][7].

Assim como na etapa de inicialização, existem diversas formas de se conduzir o processo de seleção, dentre as quais as mais comuns são:

Seleção por giro de roleta: Esse método simula o sorteio feito utilizando uma roleta na qual cada trecho é atribuído a um indivíduo da população, proporcionalmente a avaliação do mesmo. Assim, a probabilidade (p_i) de um indivíduo ser selecionado é

$$p_i = \frac{f_i}{\sum_{j=1}^n f_j} \quad [4.6]$$

Onde: n é o número de indivíduos da população e f_i é a avaliação do indivíduo i . Além disso, para que o método funcione corretamente é necessário que f_j seja positiva para os n indivíduos presentes [10].

Seleção por ranking: Nesse método os indivíduos são rankeados segundo o valor de sua adaptação e uma probabilidade de escolha é atribuída a cada indivíduo de acordo com a posição que ocupam [7].

Seleção por torneio: Nessa seleção os indivíduos da população são sucessivamente agrupados de maneira aleatória em subgrupos, e o indivíduo mais adaptado dentro desses subgrupos de soluções é selecionado para reprodução.

4.4.4. Reprodução

Na reprodução novos indivíduos, chamados de filhos, são gerados a partir da combinação do genótipo de dois ou mais indivíduos sorteados na etapa de seleção, os pais. Assim, espera-se criar indivíduos melhor adaptados, ao se juntarem informações relevantes à solução do problema [10].

Uma vez que não se sabe previamente quais características dos pais são responsáveis pelo seu alto grau de adaptação, a recombinação dessas características para a geração dos filhos é feita de maneira aleatória [6].

Para representações na forma de vetor são encolhidos um ou mais pontos de corte a partir dos quais as informações dos pais são trocadas. Tomando como exemplo a representação binária, e tomando como ponto de corte o 3 bit.

$$\begin{aligned} \text{Pai}_1 &= 001011 \rightarrow 001|011; \\ \text{Pai}_2 &= 101001 \rightarrow 101|001; \end{aligned}$$

Recombinando

$$\begin{aligned} \text{Filho}_1 &= 001|001; \\ \text{Filho}_2 &= 101|011; \end{aligned}$$

A reprodução deve sempre ser pensada de maneira a respeitar a representação dos indivíduos da população e as restrições do problema, para que não sejam gerados filhos que representem soluções inválidas.

4.4.5. Mutação

O operador de mutação é utilizado para auxiliar na variabilidade genética da população, aumentando o nível de abrangência da busca pelo solução ótima do problema. A mutação é aplicada sobre os filhos resultantes da etapa de reprodução, segundo uma probabilidade de mutação (baixa) pré-estabelecida. Dentre os operadores de mutação normalmente utilizados encontram-se [7]:

- **Mutação flip:** A probabilidade de mutação é aplicada sobre cada gene do indivíduo, caso a mutação ocorra o gene tem seu valor alterado para um dos valores possíveis do seu conjunto de alelos.
- **Mutação por troca:** Nela são sorteados n pares de genes, segundo a probabilidade de mutação, que trocam de valores entre si.
- **Mutação creep:** Um valor aleatório é somado ou subtraído do valor do gene que deve sofrer mutação.

4.4.6. Atualização e Elitismo

A atualização é o último operador genético imposto à população e simula a disputa de todos os indivíduos, pais e filhos, por um lugar no ambiente. É comum nos algoritmos genéticos que a população mantenha um tamanho simples, onde nesse caso a atualização da população pode se dar substituindo todos os indivíduos da geração i pelos indivíduos concebidos durante as etapas de reprodução e mutação (geração $i + 1$). A quantidade de filhos gerados deve ser igual ao tamanho da população. Outros critérios de inserção podem ser considerados: os filhos podem substituir seus pais (nesse caso cada indivíduo da população pode gerar apenas um filho) ou podem ser mantidos na população os indivíduos melhor adaptados dentre as gerações i e $i + 1$ [7].

O elitismo consiste na preservação dos n melhores indivíduos da população atual ao se criar a nova população afim de se garantir que a melhor resposta encontrada para o problema não se perca durante as iterações do algoritmo.

4.4.7. Finalização

A partir da inicialização da população o processo de evolução coordenado dentro do algoritmo genético se repete dentro de um loop finito de iterações realizando as operações de avaliação, seleção, reprodução, mutação e atualização da população, e então verificando se algum dos critérios de parada estabelecidos foi atingido. Os critérios de parada mais comumente utilizados são: número de gerações criadas e o grau de convergência da população atual, que é a verificação da proximidade entre os valores de avaliação dos indivíduos da população e valor da avaliação do indivíduo melhor adaptado. Na Figura 18 um fluxograma é apresentado para representar a estrutura de funcionamento de um algoritmo genético:

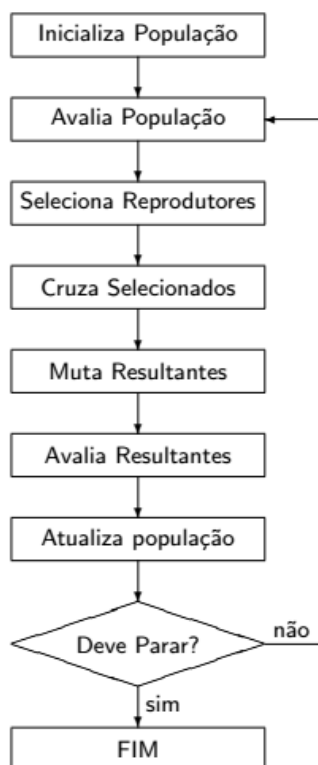


Figura 18 - Estrutura de funcionamento de um AG, retirado de [7].

A partir da maneira como os algoritmos genéticos operam, observa-se que os únicos operadores que exigem conhecimento específico do domínio do problema abordado são a representação e a avaliação. Assim, a variação entre algoritmos genéticos para solução de diferentes tipos de problema encontra-se basicamente nessas duas operações, o que atesta o grau de generalidade dessa ferramenta. Portanto, “a preocupação de um programador de algoritmos genéticos não é *de que forma* chegar a uma solução, mas sim *com o que ela deveria se parecer*” [7].

5. APLICAÇÕES PRÁTICAS: KR 10 R1420

5.1. Introdução

Abaixo, na Figura 19, encontra-se uma imagem do robô KR 10 R1420, que foi o manipulador escolhido para a realização dos teste referentes a solução da cinemática inversa propostos nesse trabalho. Na Figura 20 é possível ver os detalhes construtivos

em relação à disposição das juntas do manipulador em uma representação simplificada do mesmo.



Figura 19 - Manipulador KR 10 R1420 com 6 graus de liberdade, retirado de [18].

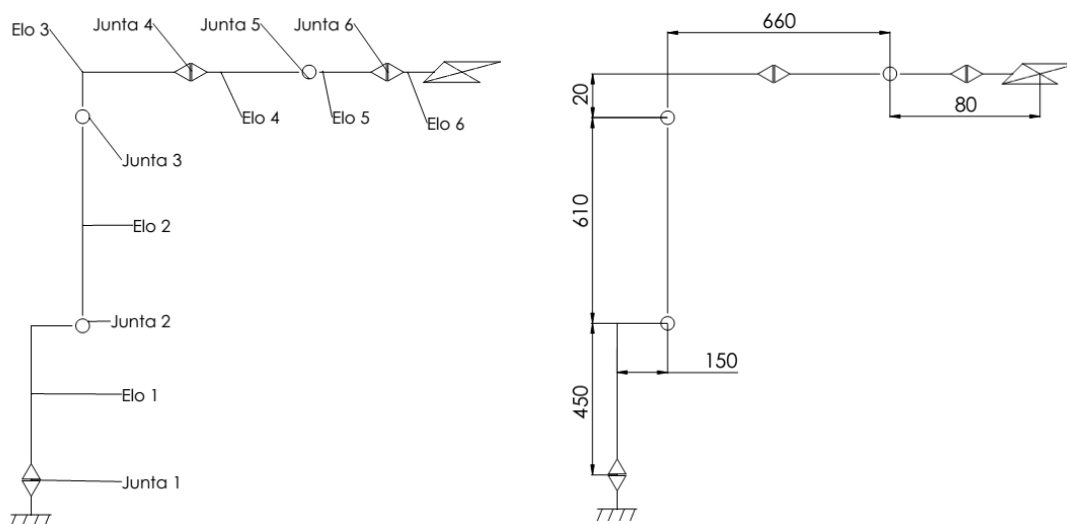


Figura 20 – Representação simplificada do rôbo manipulador KR 10 R1420.

As 3 últimas juntas do manipulador compõem seu pulso esférico, a estrutura exemplificada na Figura 10, sendo responsáveis pelo controle da orientação do

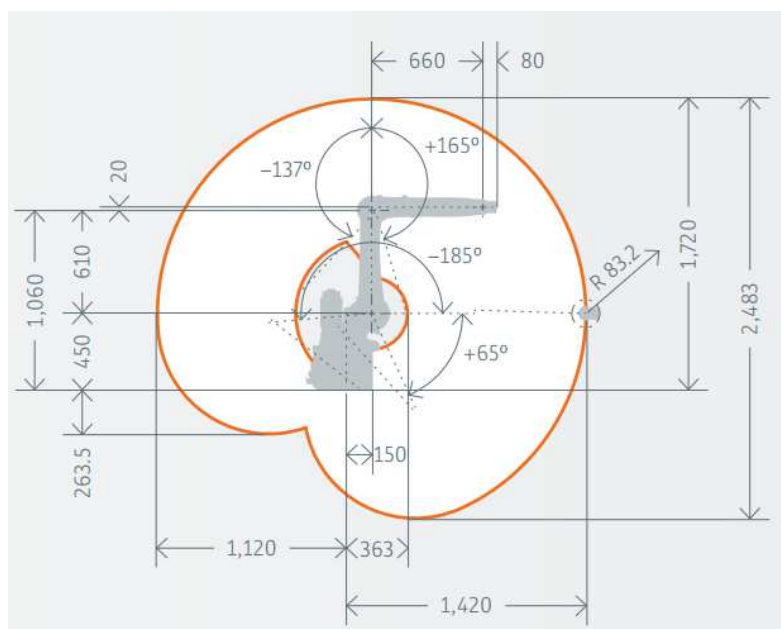
efetuador do manipulador, enquanto as 3 primeiras juntas são responsáveis pelo controle da posição do efetuador nas coordenadas desejadas no espaço.

Se observa, portanto, que a solução da cinemática inversa desse robô pode ser obtida conforme proposto por Pieper, a partir do desacoplamento do problema cinemático inverso num problema de posição inverso e num problema de orientação inverso.

Na Tabela 4 os intervalos de operação das juntas do manipulador podem ser consultados, enquanto na Figura 21 representou-se seu volume de trabalho.

Tabela 4 – Intervalo de operação das juntas

| Elo | Intervalo de operação (°) |
|------------|---------------------------|
| θ_1 | +/-170° |
| θ_2 | +65°/-185° |
| θ_3 | +165°/-137° |
| θ_4 | +/-185° |
| θ_5 | +/-120° |
| θ_6 | +/-350° |



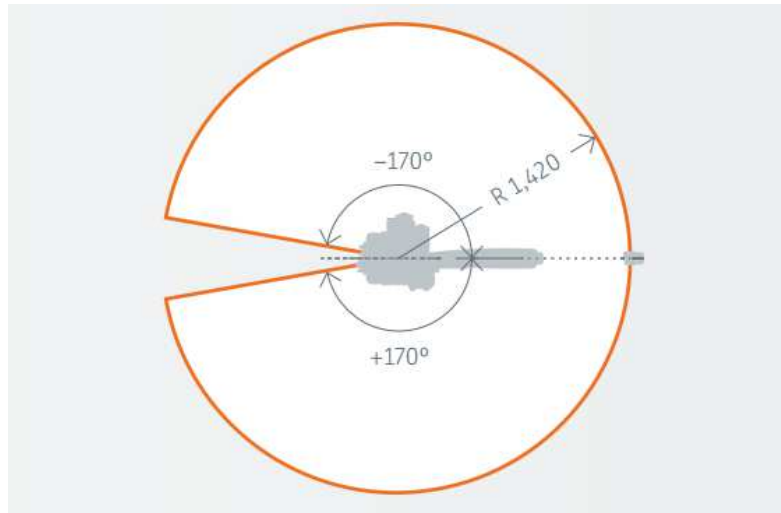


Figura 21 – Volume de trabalho do manipulador: vista frontal (sup.) e vista superior (inf.), retirado de [18].

Na Figura 22 é definida a disposição dos referenciais locais e global no manipulador, obedecendo as condições impostas pelo método de Denavit-Hartenberg e discutidas na seção 2.3.1.

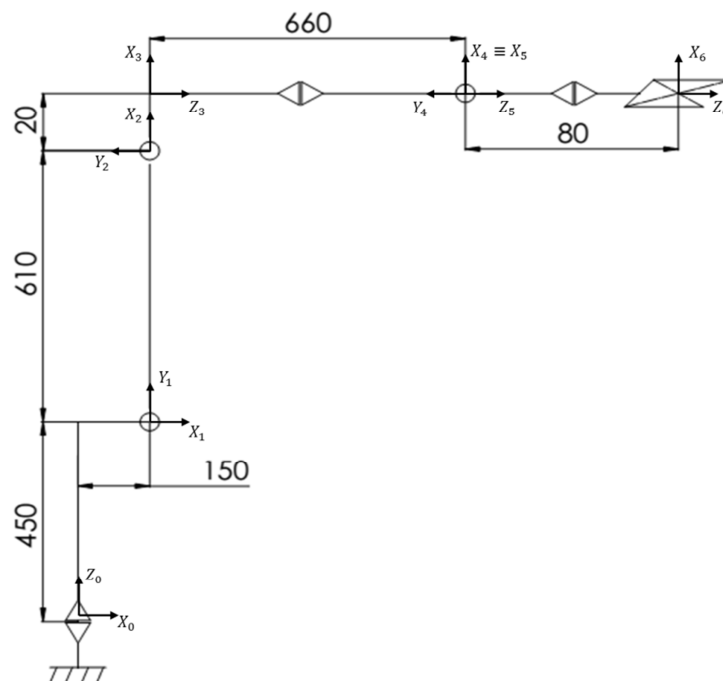


Figura 22 – Disposição dos referenciais locais e global no manipulador segundo o método de Denavit-Hartenberg.

Determinadas as posições dos referenciais global e locais, correspondentes aos elos do manipulador, e conhecidas as suas dimensões, tem-se determinadas todas as informações necessárias para solução da cinemática direta do mesmo, como será demonstrado a seguir.

5.2. Cinemática direta do manipulador

A solução da cinemática direta do manipulador inicia-se com a definição dos parâmetros de Denavit-Hartenberg, conforme detalhado na seção 2.3.1, assim, esses parâmetros foram determinados para o robô KR 10 R1420 e estão explicitados na Tabela 5:

Tabela 5 – Parâmetros de Denavit-Hartenberg para o manipulador estudado

| Elo | a_i | α_i | d_i | θ_i |
|-----|-------|------------------|-------|----------------------------|
| 1 | 150 | $\frac{\pi}{2}$ | 450 | θ_1 |
| 2 | 610 | 0 | 0 | $\theta_2 + \frac{\pi}{2}$ |
| 3 | 20 | $\frac{\pi}{2}$ | 0 | θ_3 |
| 4 | 0 | $-\frac{\pi}{2}$ | 660 | θ_4 |
| 5 | 0 | $\frac{\pi}{2}$ | 0 | θ_5 |
| 6 | 0 | 0 | 80 | θ_6 |

Utilizando-se a equação (2.11) pode-se calcular as matrizes de transformação homogênea ${}^{i-1}A_i$, onde i pretencerá ao intervalo [1,6]:

$${}^0A_1 = \begin{bmatrix} c_1 & 0 & s_1 & 150c_1 \\ s_1 & 0 & -c_1 & 150s_1 \\ 0 & 1 & 0 & 450 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

$${}^1A_2 = \begin{bmatrix} -s_2 & -c_2 & 0 & -610s_2 \\ c_2 & -s_2 & 0 & 610c_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.2)$$

$${}^2A_3 = \begin{bmatrix} c_3 & 0 & s_3 & 20c_3 \\ s_3 & 0 & -c_3 & 20s_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.3)$$

$${}^3A_4 = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & -c_4 & 0 \\ 0 & -1 & 0 & 660 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.4)$$

$${}^4A_5 = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.5)$$

$${}^5A_6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & 80 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.6)$$

Para determinação da matriz de transformação 0T_6 utiliza-se a equação (2.9):

$${}^0T_6 = A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot A_5 \cdot A_6 \quad (5.8)$$

$${}^0T_6 = \begin{bmatrix} n_x & s_x & a_x & x \\ n_y & s_y & a_y & y \\ n_z & s_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.9)$$

Onde:

$$n_x = ((-c_1s_{23}c_4 + s_1s_4)c_5 - c_1c_{23}s_5)c_6 + (c_1s_{23}s_4 + s_1c_4)s_6$$

$$\begin{aligned}
n_y &= ((-s_1 s_{23} c_4 - c_1 s_4) c_5 - s_1 c_{23} s_5) c_6 + (s_1 s_{23} s_4 - c_1 c_4) s_6 \\
n_z &= (c_{23} c_4 c_5 - s_{23} s_5) c_6 - c_{23} s_4 s_6 \\
s_x &= -((-c_1 s_{23} c_4 + s_1 s_4) c_5 - c_1 c_{23} s_5) s_6 + (c_1 s_{23} s_4 + s_1 c_4) c_6 \\
s_y &= -((-s_1 s_{23} c_4 - c_1 s_4) c_5 - s_1 c_{23} s_5) s_6 + (s_1 s_{23} s_4 - c_1 c_4) c_6 \\
s_z &= -(c_{23} c_4 c_5 - s_{23} s_5) s_6 - c_{23} s_4 c_6 \\
a_x &= (-c_1 s_{23} c_4 + s_1 s_4) s_5 + c_1 c_{23} c_5 \\
a_y &= (-s_1 s_{23} c_4 - c_1 s_4) s_5 + s_1 c_{23} c_5 \\
a_z &= c_{23} c_4 s_5 + s_{23} c_5 \\
x &= (80(-c_1 s_{23} c_4 + s_1 s_4)) s_5 + 80 c_1 c_{23} c_5 + 660 c_1 c_{23} + c_1 (150 - 610 s_2 - 20 s_{23}) \\
y &= (80(-s_1 s_{23} c_4 - c_1 s_4)) s_5 + 80 s_1 c_{23} c_5 + 660 s_1 c_{23} + s_1 (150 - 610 s_2 - 20 s_{23}) \\
z &= 450 + 80(c_{23} c_4 s_5 + s_{23} c_5) + 660 s_{23} + 610 c_2 + 20 c_{23}
\end{aligned}$$

Assim a solução da cinemática direta do manipulador foi calculada, tornando-se possível a determinação da posição e orientação do efetuador no referencial global a partir do conhecimento dos valores das variáveis de juntas.

5.3. Cinemática inversa do manipulador

Para a solução da cinemática inversa do manipulador assume-se a matriz de transformação homogênea com a postura desejada para o manipulador igual a

$${}^0T_6^d = \begin{bmatrix} n_x^d & s_x^d & a_x^d & x^d \\ n_y^d & s_y^d & a_y^d & y^d \\ n_z^d & s_z^d & z_z^d & z^d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.10)$$

Que pode ser desacoplada em uma matriz de orientação desejada, ${}^0R_6^d$, e uma matriz de posição desejada, ${}^0P_6^d$, como a seguir:

$${}^0R_6^d = \begin{bmatrix} n_x^d & s_x^d & a_x^d \\ n_y^d & s_y^d & a_y^d \\ n_z^d & s_z^d & z_z^d \end{bmatrix} \quad (5.11)$$

$${}^0P_6^d = \begin{bmatrix} x^d \\ y^d \\ z^d \end{bmatrix} \quad (5.12)$$

Como discutido na seção 2.4.2 chamaremos de ponto Q o ponto de interseção das juntas 4, 5 e 6, que compõe o pulso esférico do rôbo KR 10 R1420. Observa-se, na Figura 22, que esse ponto corresponderá as origens dos sistemas de referência locais $o_4x_4y_4z_4$ e $o_5x_5y_5z_5$. Também é correto afirmar que movimentos das juntas 4, 5 e 6 não alterarão a posição do ponto Q no espaço.

Assim, utilizando-se a equação (2.49), o vetor posição do ponto Q, P^Q , será

$$P^Q = \begin{bmatrix} x^Q \\ y^Q \\ z^Q \end{bmatrix} = \begin{bmatrix} x^d \\ y^d \\ z^d \end{bmatrix} - \begin{bmatrix} n_x^d & s_x^d & a_x^d \\ n_y^d & s_y^d & a_y^d \\ n_z^d & s_z^d & z_z^d \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ d_6 \end{bmatrix} \quad (5.13)$$

Através da expressão 2.51 é possível montar um sistema com 3 equações e 3 variáveis, conforme (5.14).

$${}^0P_6^d - {}^0R_6^d d_6 \hat{Z}_6 = {}^0T_3 {}^3P_{4org} \quad (5.14)$$

A solução do sistema acima corresponde a resolução do problema de posição inverso do manipulador, determinando-se os valores das váriáveis de junta θ_1 , θ_2 e θ_3 para a postura desejada do efetuador do rôbo. A equação (5.14) corresponde a abordagem algébrica do problema. A seguir o problema de posição inverso do manipulador será resolvido através da abordagem geométrica.

Na Figura 23 observa-se o vetor posição pQ do pulso do efetuador em relação ao referencial inercial $O_0x_0y_0z_0$, e a sua projeção pQ_{xy} no plano x_0y_0 .

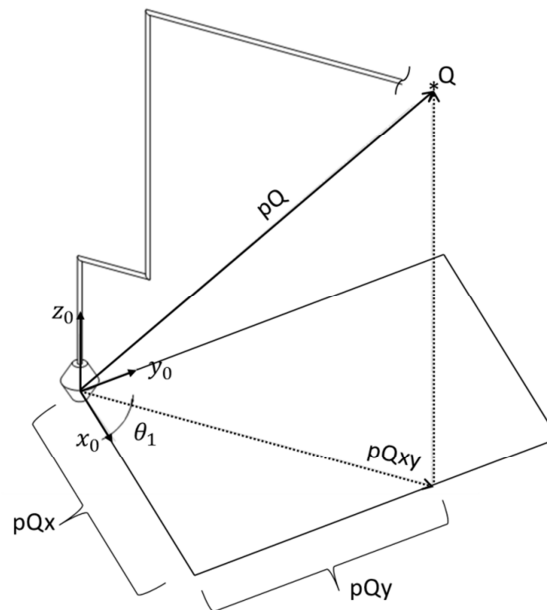


Figura 23 –Projeção do vetor posição pQ do pulso do efetuador no plano x_0y_0 .

A partir da imagem se observa que uma possível solução para o ângulo θ_1 é

$$\theta_1 = \arctg\left(\frac{pQ_y}{pQ_x}\right) = \text{atan2}\left(\frac{y^Q}{x^Q}\right) \quad (5.15)$$

Há ainda uma segunda solução para θ_1 , aqui representada por θ_1'

$$\theta_1' = \text{atan2}\left(\frac{y^Q}{x^Q}\right) + \pi \quad (5.16)$$

Nessa solução a junta 2 do efetuador fica posicionada no quadrante oposto ao do pulso, e ela será ignorada.

É importante lembrar que x^Q e y^Q são determinados a partir da postura desejada para o efetuador segundo a equação (5.13).

Observa-se que as equações (5.15) e (5.16) não são válidas quando x^Q for igual a zero; nesse caso o manipulador se encontra em uma posição singular, onde o centro do punho encontra-se sobre o eixo z_0 , o que significa que qualquer valor de θ_1 satisfará a configuração.

Para se determinar geometricamente os valores de θ_2 e θ_3 considera-se o plano vertical formado pelos elos 2 e 3 do manipulador:

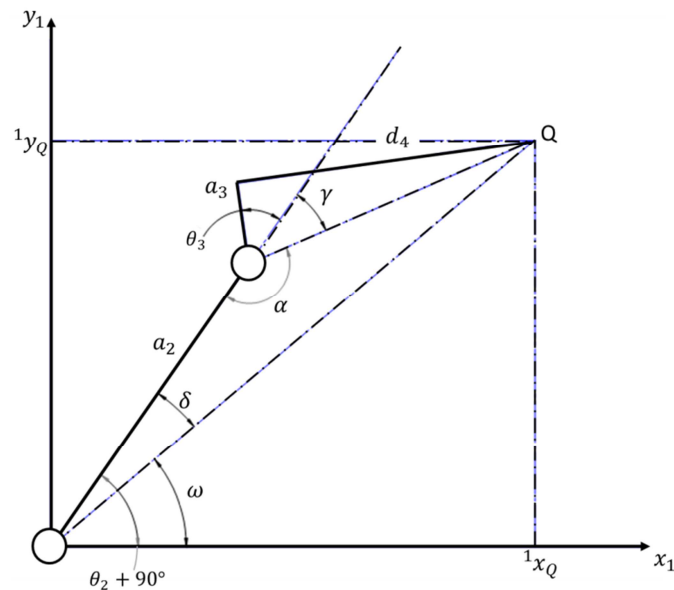


Figura 24 –Plano vertical formado pelos elos 2 e 3 do manipulador.

Para utilização do diagrama acima as coordenadas do ponto Q definidas no referencial inercial $O_0x_0y_0z_0$ através da equação (5.13) são redefinidas no referencial local $O_1x_1y_1z_1$ como demonstrado na equação (5.17) abaixo.

$$\begin{bmatrix} {}^1x_Q \\ {}^1y_Q \\ {}^1z_Q \end{bmatrix} = {}^0T_1^{-1} \begin{bmatrix} x_Q \\ y_Q \\ z_Q \end{bmatrix} \quad (5.17)$$

Assim, usando-se a lei dos cossenos se obtém a equação (5.18).

$${}^1x_Q^2 + {}^1y_Q^2 = d_4^2 + a_3^2 + a_2^2 - 2 \times a_2 \times \sqrt{d_4^2 + a_3^2} \times \cos(\alpha) \quad (5.18)$$

Que pode ser simplificada para determinação de $\cos(\gamma)$.

$${}^1x_Q^2 + {}^1y_Q^2 = d_4^2 + a_3^2 + a_2^2 - 2 \times a_2 \times \sqrt{d_4^2 + a_3^2} \times \cos(180 - \gamma) \quad (5.19)$$

$$\cos(\gamma) = \frac{{}^1x_Q^2 + {}^1y_Q^2 - d_4^2 - a_3^2 - a_2^2}{2 \times a_2 \times \sqrt{d_4^2 + a_3^2}} \quad (5.20)$$

E o ângulo γ é determinado em (5.22).

$$\sin(\gamma) = \pm \sqrt{1 - \cos(\gamma)^2} \quad (5.21)$$

$$\gamma = \arctg\left(\frac{\sin(\gamma)}{\cos(\gamma)}\right) \quad (5.22)$$

Da equação (5.21) verifica-se que haverão dois valores possíveis para $\sin(\gamma)$. Isso acontece porque uma vez definido o ângulo θ_1 para um ponto no espaço existirão duas combinações de θ_2 e θ_3 que satisfarão a configuração desejada:

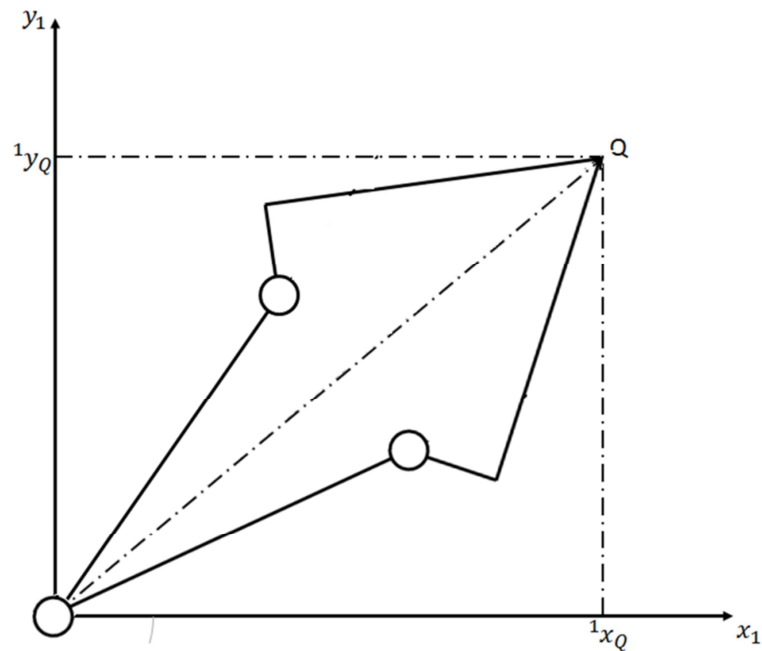


Figura 25 – As duas soluções possíveis para cinemática inversa.

Finalmente, o ângulo θ_3 é determinado a partir da Figura 24.

$$\theta_3 = \arctg\left(\frac{d_4}{a_3}\right) - \gamma \quad (5.23)$$

A lei dos cossenos é novamente utilizada, dessa vez para determinação de $\cos(\delta)$.

$$\cos(\delta) = \frac{d_4^2 + a_3^2 - a_2^2 - {^1x_Q}^2 - {^1y_Q}^2}{-2 \times a_2 \times \sqrt{{^1x_Q}^2 + {^1y_Q}^2}} \quad (5.24)$$

E a lei dos senos para determinação de $\sin(\delta)$.

$$\frac{\sqrt{{^1x_Q}^2 + {^1y_Q}^2}}{\sin(\alpha)} = \frac{\sqrt{d_4^2 + a_3^2}}{\sin(\delta)} \quad (5.25)$$

$$\text{sen}(\delta) = \frac{\sqrt{d_4^2 + a_3^2}}{\sqrt{{}^1x_Q^2 + {}^1y_Q^2}} \times \text{sen}(180 - \gamma) \quad (5.26)$$

Assim,

$$\delta = \text{arctg} \left(\frac{\text{sen}(\delta)}{\text{cos}(\delta)} \right) \quad (5.27)$$

Por último, os ângulos ω e θ_2 são determinados a partir da Figura 24.

$$\omega = \text{arctg} \left(\frac{{}^1y_Q}{{}^1x_Q} \right) \quad (5.28)$$

$$\theta_2 = \omega + \delta - 90^\circ \quad (5.29)$$

Determinados os valores das 3 primeiras juntas a matriz de rotação referente as 3 últimas juntas pode ser calculada segundo equação (2.53), que para um manipulador com 6 juntas será

$${}^3R_6 = {}^0R_3^T {}^0R_6^d \quad (5.30)$$

A matriz de rotação 0R_3 corresponde a parte de rotação da matriz de transformação homogênea 0T_3 .

$${}^0T_3 = A_1 \cdot A_2 \cdot A_3 \quad (5.31)$$

$${}^0T_3 = \begin{bmatrix} {}^0n_x^3 & {}^0s_x^3 & {}^0a_x^3 & {}^0x^3 \\ {}^0n_y^3 & {}^0s_y^3 & {}^0a_y^3 & {}^0y^3 \\ {}^0n_z^3 & {}^0s_z^3 & {}^0a_z^3 & {}^0z^3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.32)$$

Isolando-se 0R_3

$${}^0R_3 = \begin{bmatrix} {}^0n_x^3 & {}^0s_x^3 & {}^0a_x^3 \\ {}^0n_y^3 & {}^0s_y^3 & {}^0a_y^3 \\ {}^0n_z^3 & {}^0s_z^3 & {}^0a_z^3 \end{bmatrix} \quad (5.33)$$

Onde,

$${}^0n_x^3 = -c_1s_{23}$$

$${}^0n_y^3 = -s_1s_{23}$$

$${}^0n_z^3 = c_{23}$$

$${}^0s_x^3 = s_1$$

$${}^0s_y^3 = -c_1$$

$${}^0s_z^3 = 0$$

$${}^0a_x^3 = c_1c_{23}$$

$${}^0a_y^3 = s_1c_{23}$$

$${}^0a_z^3 = s_{23}$$

Assim, a equação (5.30) pode ser reescrita como em (5.34),

$${}^3R_6 = \begin{bmatrix} -c_1s_{23} & -s_1s_{23} & c_{23} \\ s_1 & -c_1 & 0 \\ c_1c_{23} & s_1c_{23} & s_{23} \end{bmatrix} \begin{bmatrix} n_x^d & s_x^d & a_x^d \\ n_y^d & s_y^d & a_y^d \\ n_z^d & s_z^d & z_z^d \end{bmatrix} \quad (5.34)$$

Além disso, 3R_6 pode ser calculado a partir da multiplicação das matrizes A_4, A_5, A_6 , como demonstrado na obtenção de 0R_3 , o que resultará na expressão (5.35).

$${}^3R_6 = \begin{bmatrix} c_4 c_5 c_6 - s_4 s_6 & -s_4 c_6 - c_4 c_5 s_6 & c_4 s_5 \\ s_4 c_5 c_6 - c_4 s_6 & c_4 c_6 - s_4 c_5 s_6 & s_4 s_5 \\ -s_5 c_6 & s_5 s_6 & c_5 \end{bmatrix} \quad (5.35)$$

Finalmente, igualando-se as expressões (5.34) e (5.35) e resolvendo o sistema, os valores das juntas 4, 5 e 6 podem ser determinados a partir dos cálculos dos seus senos e cossenos:

$$c_5 = a_z^d s_{23} + a_x^d c_1 c_{23} \quad (5.36)$$

$$s_5 = \pm \sqrt{1 - c_5^2} \quad (5.37)$$

$$\theta_5 = \text{atan2}\left(\frac{s_5}{c_5}\right) \quad (5.38)$$

$$s_4 = \frac{a_x^d s_1 - a_y^d c_1}{s_5} \quad (5.39)$$

$$c_4 = \frac{a_z^d c_{23} - a_x^d c_1 s_{23} - a_y^d s_1 s_{23}}{s_5} \quad (5.40)$$

$$\theta_4 = \text{atan2}\left(\frac{s_4}{c_4}\right) \quad (5.41)$$

$$s_6 = \frac{s_z^d s_{23} + s_x^d c_1 c_{23} + s_y^d s_1 c_{23}}{s_5} \quad (5.42)$$

$$c_6 = -\frac{n_z^d s_{23} + n_x^d c_1 c_{23} + n_y^d s_1 c_{23}}{s_5} \quad (5.43)$$

$$\theta_6 = \text{atan2}\left(\frac{s_6}{c_6}\right) \quad (5.44)$$

Assim como aconteceu durante a solução do problema de posição inverso, definido θ_5 , existirão duas combinações de θ_4 e θ_6 que corresponderão à orientação desejada. A equação (5.37) comprova a existência de dois valores possíveis para θ_5 .

Se observa também que quando θ_5 for igual a zero as equações (5.39) a (5.44) serão indeterminações matemáticas já que s_5 será igual a zero. Isso acontece porque essa posição constitui uma singularidade, os eixos das juntas 4 e 6 estarão alinhados, e haverá infinitas soluções para a orientação desejada.

Uma possível solução nesse caso será fixar o valor de θ_4 , que aqui será sempre igual a zero, assim o sistema obtido ao se igualar as equações (5.34) e (5.35) pode ser resolvido para esse caso particular, resultando em:

$$s_4 = 0 \quad (5.45)$$

$$c_4 = 1 \quad (5.46)$$

$$s_6 = s_x^d c_1 s_{23} + s_y^d s_1 s_{23} - s_z^d c_{23} \quad (5.47)$$

$$c_6 = n_z^d c_{23} - n_x^d c_1 s_{23} - n_y^d s_1 s_{23} \quad (5.48)$$

Nessa seção utilizou-se o desacoplamento do problema cinemático inverso no problema de posição inverso e no problema de orientação inverso, e dessa maneira foi possível encontrar a solução do problema geral para as seis juntas que compõe o manipulador estudado.

A abordagem aqui utilizada foi completamente algébrica/geométrica, agora serão discutidos a montagem e treinamento de um conjunto de redes neurais como uma solução alternativa ao problema, além da utilização de algoritmos genéticos para auxiliar na otimização das redes.

5.4. Utilização de redes neurais na solução da cinemática inversa do manipulador

Diferentes conjuntos de redes neurais, com diferentes entradas e saídas, foram experimentados afim de se resolver a cinemática inversa do rôbo KR 10 R1420. A

configuração do melhor conjunto de entradas e saídas encontrado está representado nas figuras abaixo.

Na Figura 26 encontram-se as redes utilizadas na solução do problema de posição inverso do manipulador. A partir da postura desejada para o efetuador, representada por ${}^0T_6^d$, as redes 1, 2 e 3 são utilizadas para determinar as coordenadas x_Q^d , y_Q^d e z_Q^d correspondentes para o pulso do manipulador. O seno e o cosseno de θ_1 são determinados algebricamente a partir dos valores de x_Q^d e y_Q^d e são as entradas de rede 4, cuja saída é o valor correspondente de θ_1 .

Como demonstrado na solução algébrica do problema, as coordenadas do ponto Q no referencial $O_1x_1y_1z_1$ são calculadas. Uma vez que o ponto Q sempre pertencerá ao plano x_1y_1 é correto afirmar que 1z_Q será sempre igual a zero, e por isso as entradas da rede 5 são apenas as coordenadas 1x_Q e 1y_Q , a partir das quais a rede 5 calcula os ângulos θ_2 e θ_3 correspondentes.

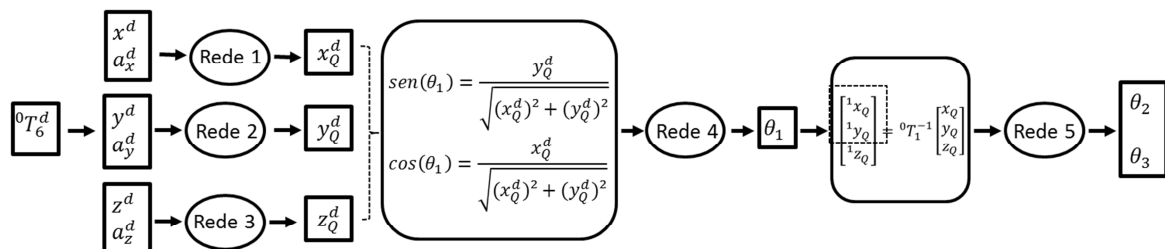


Figura 26 - Entradas e saídas das redes neurais na solução do problema de posição inverso.

Uma vez encontrados os valores das juntas θ_1 , θ_2 e θ_3 calcula-se os valores de s_5 e c_5 segundo as equações (5.36) e (5.37). Esses valores são as entradas da rede 6, responsável por calcular θ_5 . Na prática a rede 4 e a rede 6 são a mesma rede, tendo sido treinada com entradas correspondentes a saídas num intervalo de +/- 170°, o

intervalo de operação da junta 1, e sendo capaz de calcular θ_5 , que corresponde a um intervalo menor.

Conhecido θ_5 determinam-se os valores dos senos e cossenos de θ_4 e θ_6 . A junta 4 possui um intervalo de operação de +/- 185°, e a junta 6 +/- 350°, excendendo o intervalo de ângulos para o qual a rede 4 foi treinada. Por isso duas novas redes foram treinadas, uma para ângulos com senos positivos, de 0° a 180°, e a outra para ângulos com senos negativos, de 0° a -180°. Na Figura 27 es tão representadas as entradas e saídas das redes neurais na solução do problema de orientação inverso.

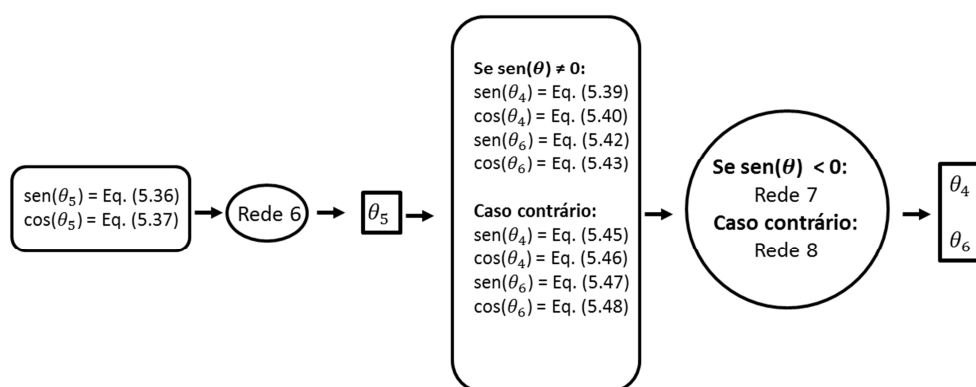


Figura 27 - Entradas e saídas das redes neurais na solução do problema de orientação inverso.

A quantidade de camadas escondidas para cada uma das redes, assim como a quantidade de neurônios por camada foi determinada de forma experimental. A quantidade média de dados de entrada e saída das redes foi de 135000 exemplos, e o número de iterações foi determinado a partir dos resultados apresentados pelas redes durante os respectivos treinamentos.

Para o treinamento de todas as redes foi utilizada a função *trainbr* do Matlab® que utiliza como função de treinamento a backpropagação bayesiana de regularização, onde os pesos e biases da rede neural são atualizados através da otimização de

Levenberg-Marquardt, e uma taxa de aprendizado igual a 0.2. Na Tabela 6 se encontram os parâmetros de treinamento das redes.

Tabela 6 – Parâmetros de treinamento das redes.

| | REDE 1 | REDE 2 | REDE 3 | REDE 4/6 | REDE 5 | REDE 7 | REDE 8 |
|------------------------------|-----------------------|-----------------------|----------------------|------------------------|------------------------|------------------------|------------------------|
| Configuração | [20 20 20] | [20 20 20] | [20 20 20] | [20 20 20] | [25 25 25 25] | [20 20 20] | [20 20 20] |
| Qtd de exemplos | 155925 | 155925 | 155925 | 118683 | 139440 | 125664 | 125664 |
| Nº de iterações | 1000 | 1000 | 1000 | 2000 | 2770 | 2000 | 2000 |
| Erro quadrático médio | $6,15 \times 10^{-8}$ | $1,14 \times 10^{-9}$ | $5,5 \times 10^{-8}$ | $6,05 \times 10^{-14}$ | $7,01 \times 10^{-11}$ | $1,99 \times 10^{-14}$ | $3,03 \times 10^{-15}$ |

Uma vez que todo o conjunto foi treinado foi criada uma trajetória de teste com 300 pontos distintos, e que cuja cinemática inversa deveria ser encontrada pelo conjunto de redes neurais a partir das coordenadas dos pontos.

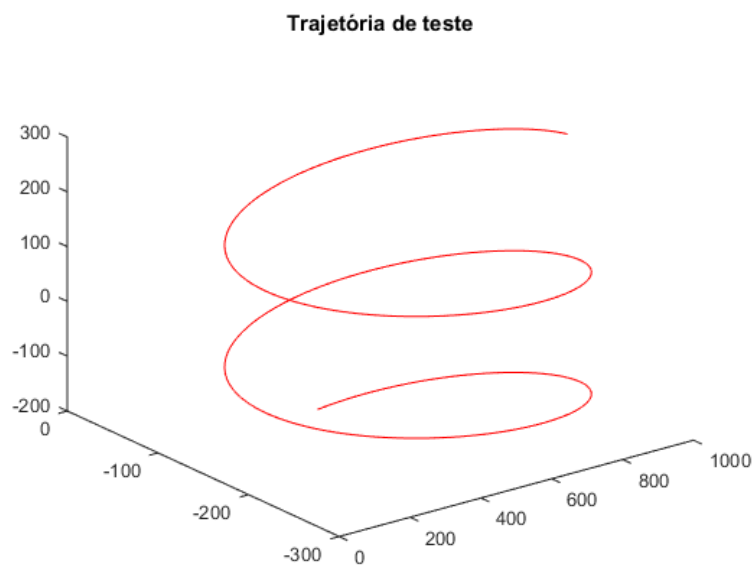


Figura 28 - Trajetória para teste das redes neurais.

A Figura 29 compreende os erros das redes 1, 2 e 3 na previsão das coordenadas do ponto Q (centro do pulso esférico) ao longo da trajetória. E na Figura 30 se encontram os erros referentes aos valores das juntas.

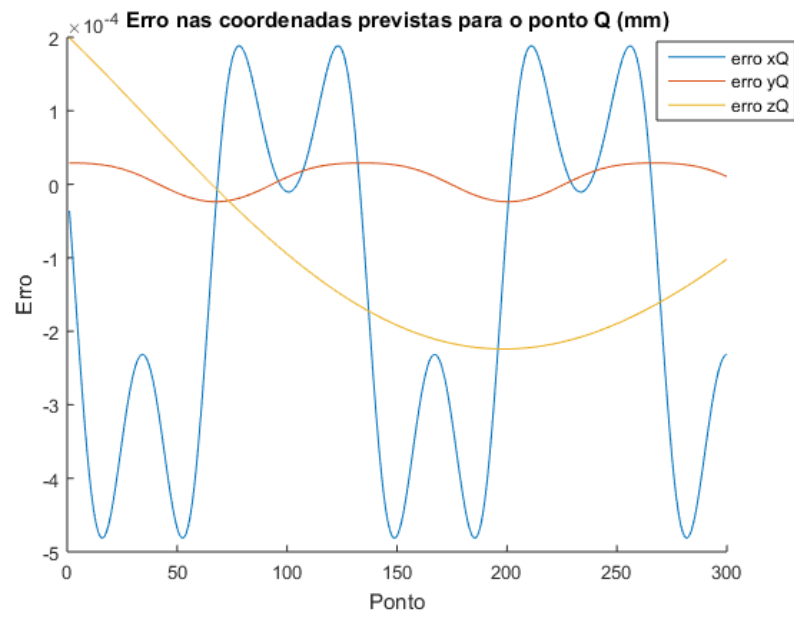


Figura 29 - Erro nas coordenadas previstas para o pulso esférico.

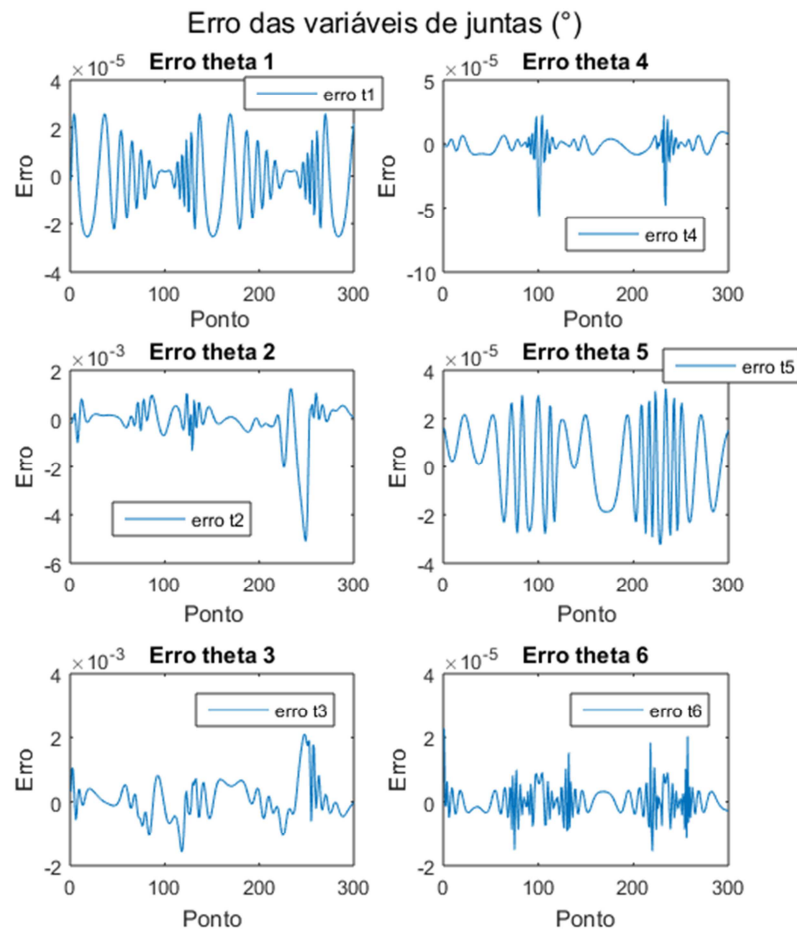


Figura 30 - Erro no valor calculado para as juntas ao longo da trajetória.

O erro na posição do pulso nas coordenadas x, y e z no referencial inercial, e o erro total correspondente a distância entre o ponto desejado na trajetória e o ponto alcançado pelo pulso estão plotados respectivamente nas figuras 31 e 32.

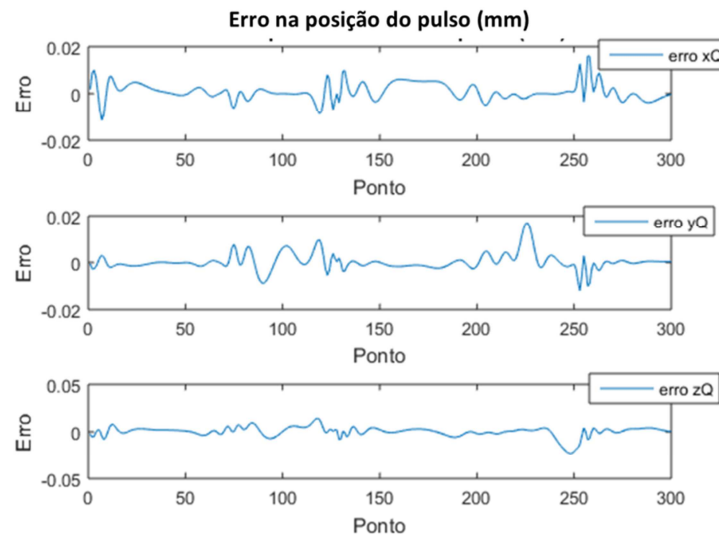


Figura 31 - Erro na posição do pulso nas coordenadas x, y e z.

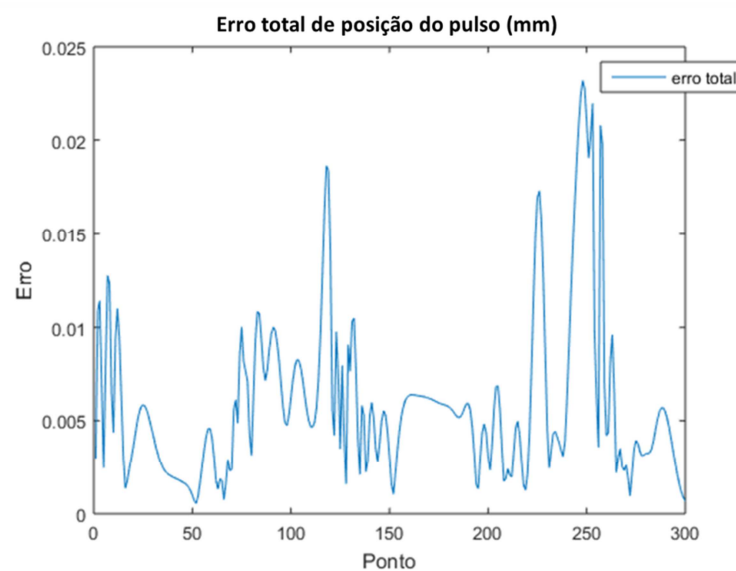


Figura 32 - Erro total de posição do pulso.

Já os erros na posição do efetuador encontram-se nas figuras 33 e 34. Se observa que o erro na posição do pulso é função dos erros das juntas 1, 2 e 3,

enquanto o erro na posição do efetuador é função do erro das 6 juntas. Verifica-se também, que não há diferença apreciável entre os erros do pulso e do efetuador, concluindo-se que uma redução nos erros das juntas 1, 2 e 3 seria o primeiro passo para otimizar ainda mais os resultados.

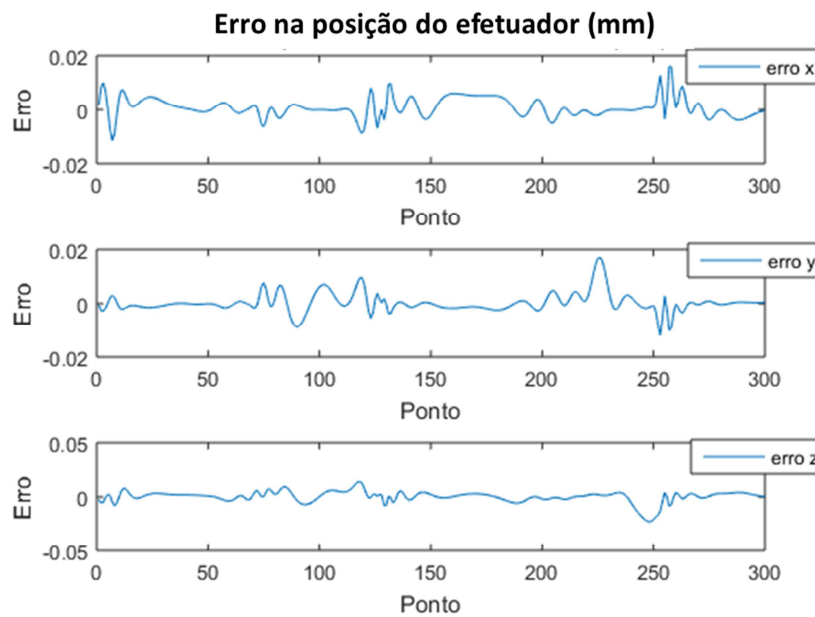


Figura 33 - Erro na posição do efetuador nas coordenadas x, y e z.

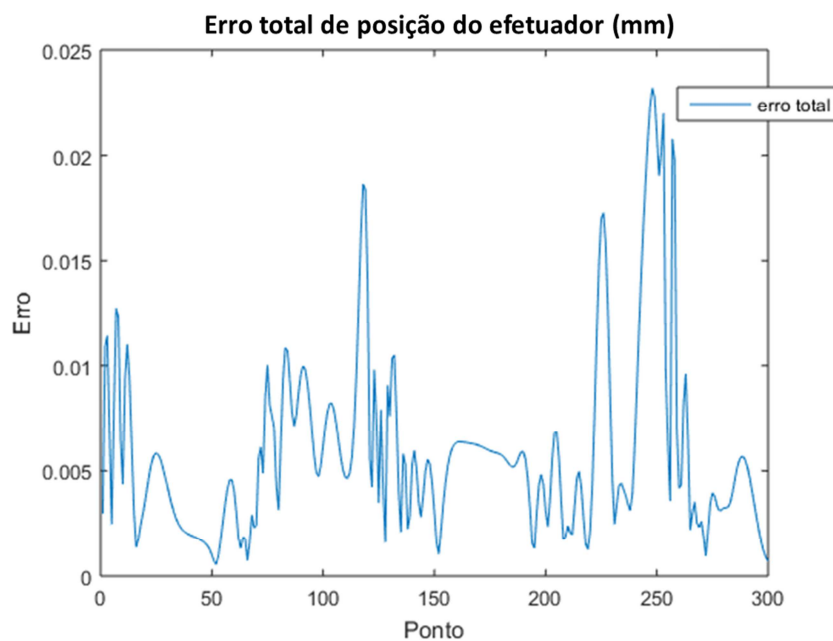


Figura 34 - Erro total de posição do efetuador.

Finalmente, na Figura 35 encontra-se o erro de orientação do efetuador e na Figura 36 a comparação entre a trajetória desejada e aquela realizada pelo efetuador a partir dos valores das juntas encontrados pelo conjunto de redes neurais.

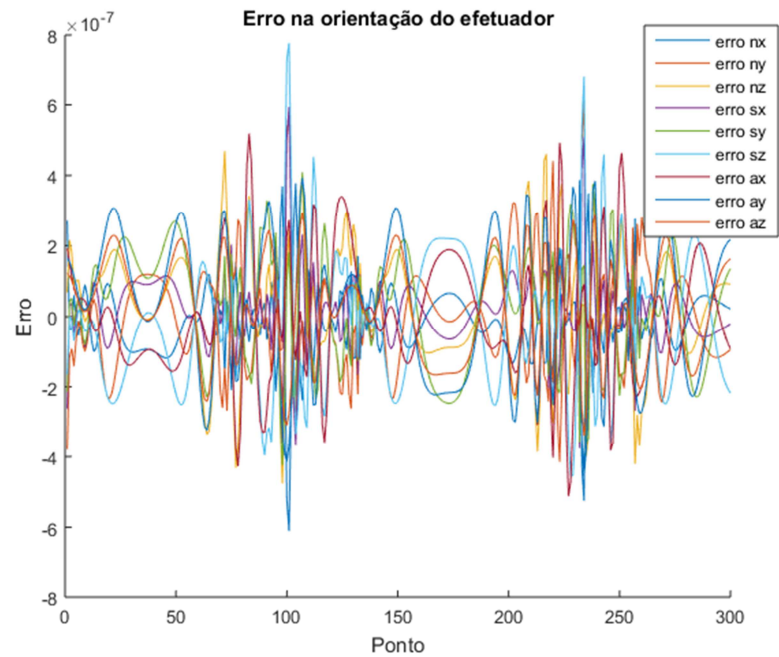


Figura 35 - Erro na orientação do efetuador.

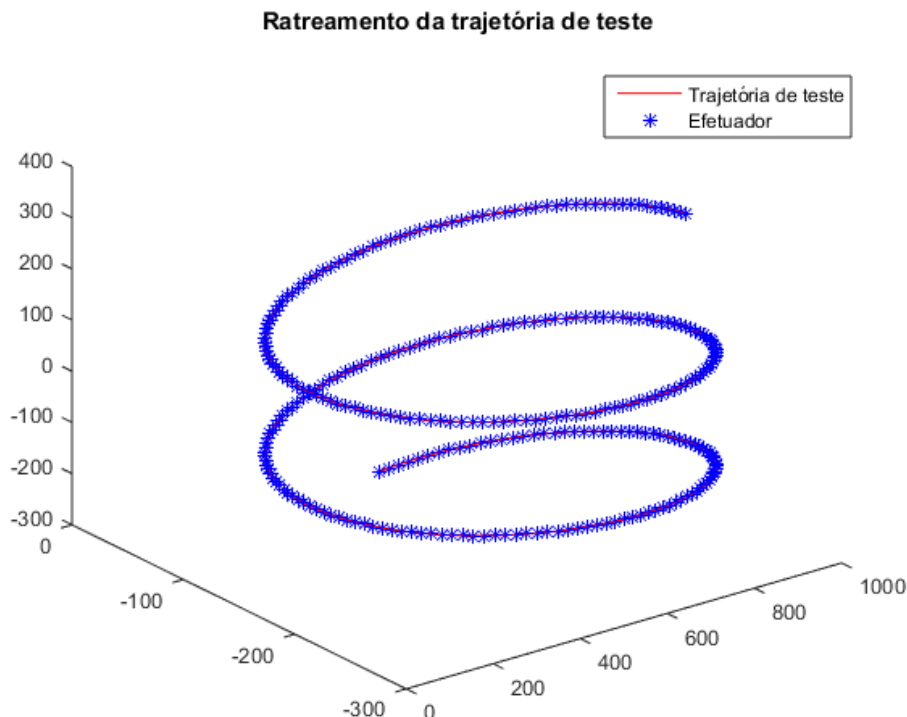


Figura 36 - Rastreamento da trajetória obtido.

A partir dos resultados encontrados se observa que o erro máximo no posicionamento do efetuador, a partir do cálculo das redes neurais, é da ordem de centésimos de milímetros para a posição, sendo o erro médio da ordem de milésimos de milímetros (0,006mm), e da ordem de 10^{-7} para a orientação, tratando-se por isso de resultados promissores.

5.5. Definição da topologia de uma rede neural através de algoritmos genéticos

Como discutido no capítulo 3, o primeiro desafio encontrado a partir da escolha pela utilização de redes neurais artificiais para resolução de um problema trata da definição da estrutura de rede neural, número de camadas escondidas, número de neurônios em cada camada e função de transferência, que otimizará o processo de treinamento da mesma. Apesar de não haver uma metodologia genérica para a

definição da arquitetura ótima de uma rede neural métodos têm sido desenvolvidos na tentativa de se automatizar essa etapa do processo de utilização de uma RNA devido ao seu potencial para as aproximações de funções onde a forma explícita e as relações entre as variáveis são desconhecidas [6].

Uma das ferramentas utilizadas para definição da topologia de uma RNA (e, em alguns casos, também para o cálculo dos seus pesos sinápticos) são os algoritmos genéticos, tendo demonstrado resultados promissores quando as redes neurais treinadas devem resolver problemas de classificação [10][9].

Nesse trabalho, uma abordagem parecida foi utilizada e dada a complexidade intrínseca do problema de posição inverso, foi empregado um algoritmo genético apenas para o estabelecimento da topologia da rede neural, enquanto o treinamento da mesma foi feito através do método de backpropagation utilizando-se a biblioteca fornecida pelo programa Matlab®.

Cada indivíduo da população do algoritmo genético representava uma configuração de rede neural (número de camadas e quantidade de neurônios por camada) candidato a ser o melhor indivíduo da população.

A população inicial (geração 1) foi gerada de forma aleatória, onde seus indivíduos possuíam 2 ou 3 camadas escondidas e entre 8 e 25 neurônios cada. Tratava-se de uma população com 26 habitantes, uma probabilidade de cruzamento de 90% e uma probabilidade de mutação de 5%.

Para a avaliação dos habitantes foi utilizada a soma de duas funções exponenciais, sendo: a primeira uma função do erro quadrático médio obtido durante o treinamento com um valor que variava entre 0 e 5, para erros entre infinito e zero respectivamente; e a segunda, função do gradiente da função de treinamento que ao final do mesmo apresentava valores que variavam também entre 0 e 5, para valores de gradientes entre 0 e infinito respectivamente. Assim, os valores de adaptabilidade dos indivíduos variavam entre 0 e 10. Também foi aplicada uma penalidade no valor de adaptabilidade dos indivíduos em função do seu número de camadas.

Cada habitante foi treinado por 750 gerações utilizando-se a função *traingdm* do Matlab®, onde os pesos das sinapses são atualizados através da técnica de back-propagation com termo de momentum. Essa técnica foi escolhida ao invés da *trainbr*, que apresenta resultados melhores, por se tratar de uma técnica onde as interações acontecem de forma mais rápida, o que é suficiente em termos de comparação de desempenho entre topologias de redes neurais.

A seleção de pais para reprodução foi feita utilizando-se o método de giro de roleta, com a utilização de um método de *scaling* linear sobre o valor de adaptação dos indivíduos. O *scaling* funciona escalonando os valores de adaptação dos indivíduos a fim de prevenir a convergência prematura da população.

E para reprodução foi utilizado o método de cruzamento de um ponto. Assim um indivíduo com n camadas e ponto de cruzamento $i < n$ e um indivíduo com m camadas e ponto de cruzamento $j < m$ gerariam dois filhos, o primeiro com $i + m - j$ camadas e o segundo com $j + n - i$ camadas. Na etapa de reprodução foi definida ainda uma probabilidade de cruzamento de 90%, o que implica em uma probabilidade de 10% dos filhos serem iguais aos pais.

Na etapa de mutação cada camada do filho gerado era sujeita a probabilidade de sofrer mutação (p_m) de 5%. Caso ela ocorresse, haveria uma variação entre 15% e 30% do número de neurônios da camada em questão, para mais ou para menos (mutação creep).

Para atualização da população a cada geração a geração $i + 1$ substituiu a população i completamente, exceto pela preservação do melhor indivíduo da população i .

Como discutido anteriormente, uma otimização da solução geral da cinemática inversa a partir da utilização do conjunto de redes neurais poderia se iniciar diminuindo-se os erros das juntas 1, 2 e 3. Além disso, se observa, na Figura 30, que as juntas 2 e 3, cujos valores são calculados pela rede 5, apresentam erros duas ordens de grandeza maiores que as demais juntas, assim o algoritmo genético

proposto no presente trabalho foi testado na busca de uma topologia alternativa para a rede 5.

No gráfico da Figura 37 está plotada a diversidade da população após a execução do algoritmo genético por 50 gerações. Observam-se quedas bruscas na diversidade principalmente das gerações 44 e 48, provavelmente pelo nascimento de super indivíduos, mas essas quedas são atenuadas graças ao *scaling* e a substituição total da população a cada geração.

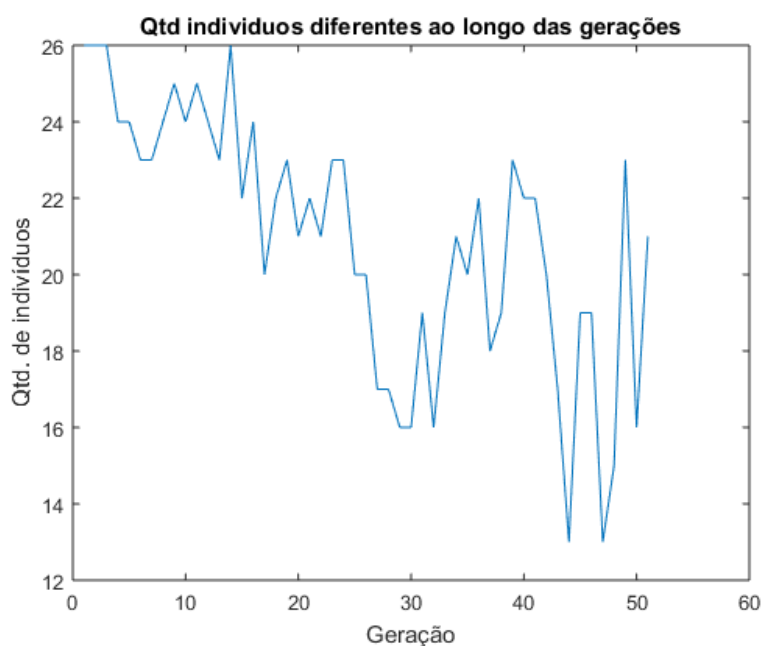


Figura 37 – Diversidade de população após 50 gerações.

Na Figura 38 são comparados os níveis de adaptabilidade do indivíduo melhor adaptado, do pior adaptado e a média da população ao longo das gerações.

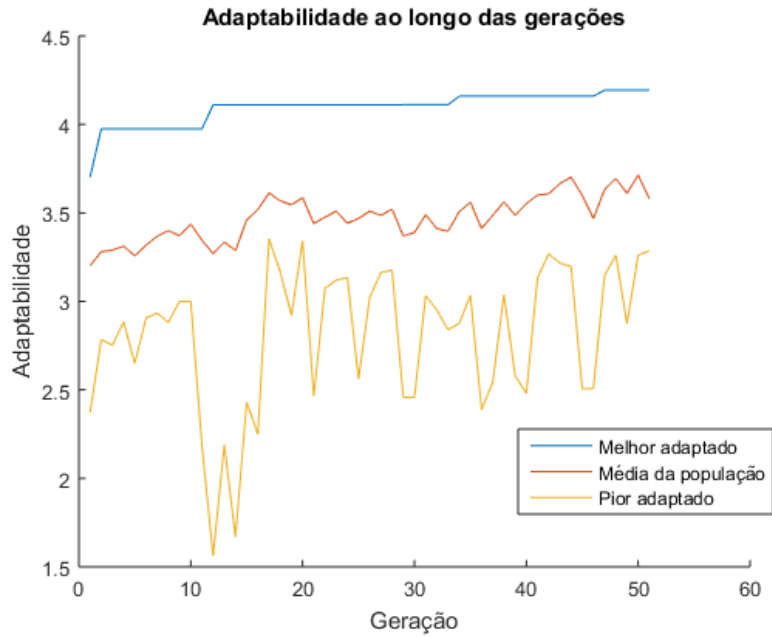


Figura 38 - Grau de adaptabilidade média, do indivíduo melhor adaptado e do indivíduo pior adaptado pelas gerações.

Na Figura 39 acompanha-se o valor do erro quadrático médio do indivíduo melhor adaptado em cada geração. Sempre que há uma variação no valor significa que uma nova topologia ótima foi encontrada pelo algoritmo genético.

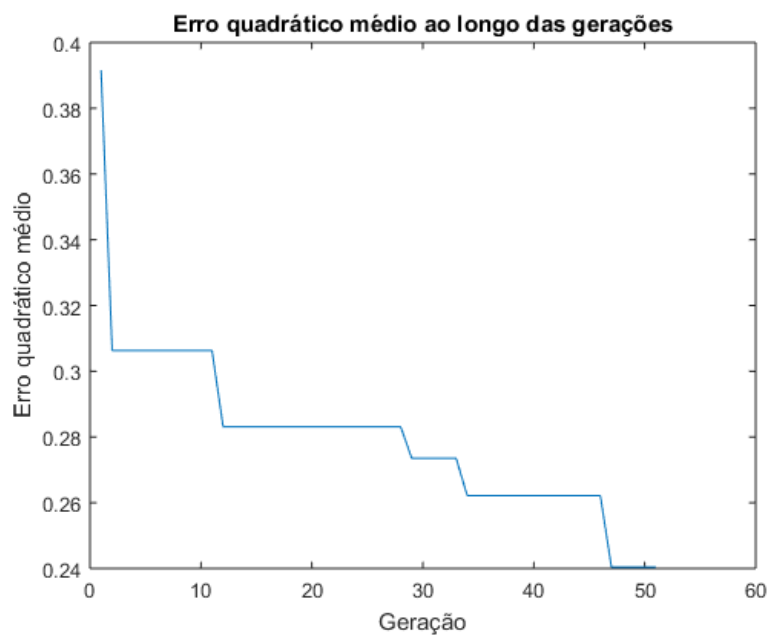


Figura 39 - Erro quadrático médio do habitante melhor adaptado.

A melhor topologia encontrada foi uma rede neural com 4 camadas escondidas com 31, 24, 18, 23 neurônios respectivamente. Se observa que o algoritmo se aproximou da topologia encontrada experimentalmente, de 4 camadas com 25 neurônios, tendo sido capaz, inclusive de identificar a necessidade de se extrapolar os limites impostos nos indivíduos aleatoriamente gerados, de 3 camadas e 25 neurônios. O próximo passo, para validação da topologia definida pelo algoritmo genético, foi treinar a rede neural encontrada por 2770 iterações utilizando-se a função *trainbr* do Matlab®, como foi feito com a primeira rede.

5.5.1. Treinamento da nova rede neural definida

Definida a nova topologia da rede neural o treinamento se deu por 2770 ciclos, utilizando-se a mesma função de treinamento usada na rede original.

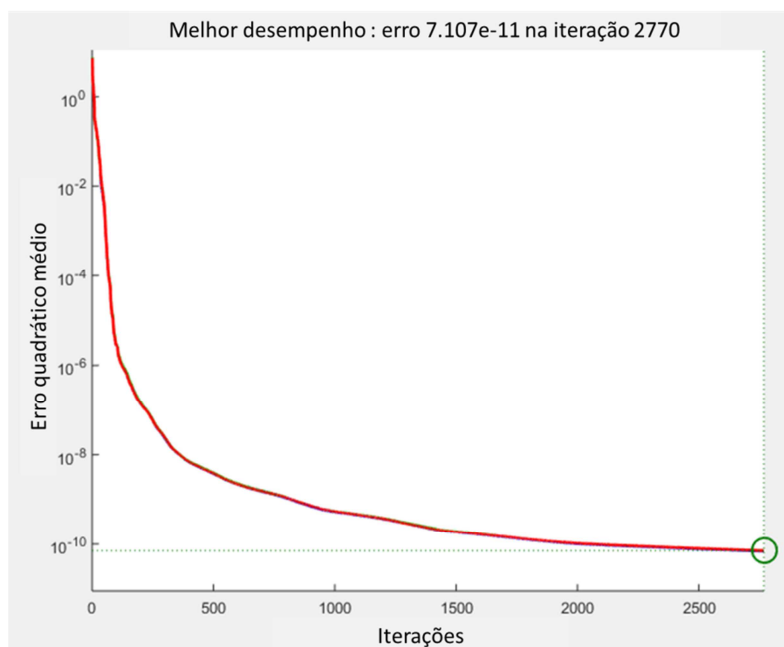


Figura 40 – Avanço do erro quadrático médio durante o treinamento da rede neural original.

Os decaimentos do erro quadrático médio durante o treinamento da rede neural original e aquele obtido durante o treinamento da rede neural com a topologia definida pelo algoritmo genético podem ser visto nas figuras 40 e 41, respectivamente. Através dos gráficos se percebe um desempenho um pouco superior da rede neural cuja topologia foi definida empiricamente.

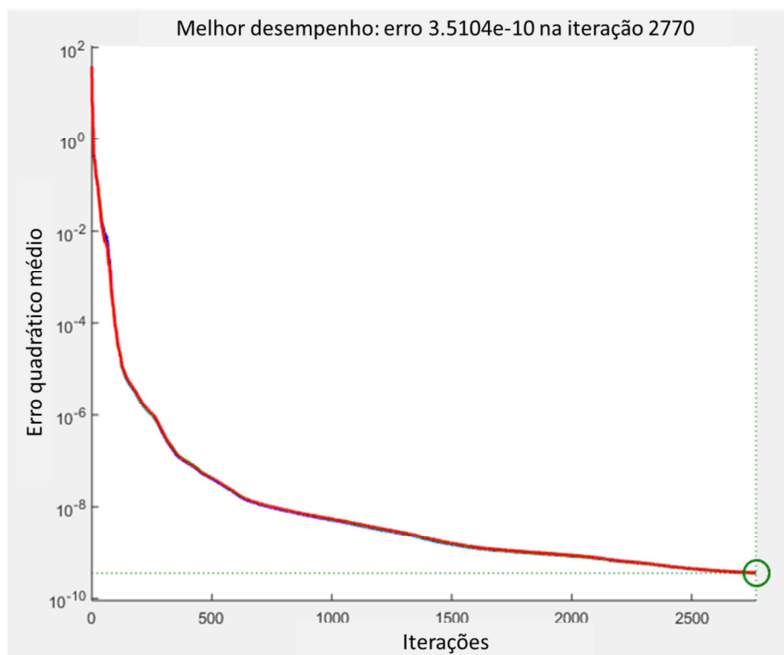


Figura 41 – Avanço do erro quadrático médio durante o treinamento da rede neural com a topologia definida pelo algoritmo genético.

Ao final do seu treinamento, a nova rede neural foi testada, calculando a cinemática inversa da trajetória de teste utilizada na primeira rede neural.

Na Figura 42 estão apresentados os valores dos erros das juntas ao longo dos pontos da trajetória. Observa-se que não houve variação apreciável na ordem de grandeza do valor dos erros das juntas 2 e 3, calculadas diretamente pela rede. Assim como também não foi observada nenhuma variação significativa nos valores dos erros das juntas 4, 5 e 6.

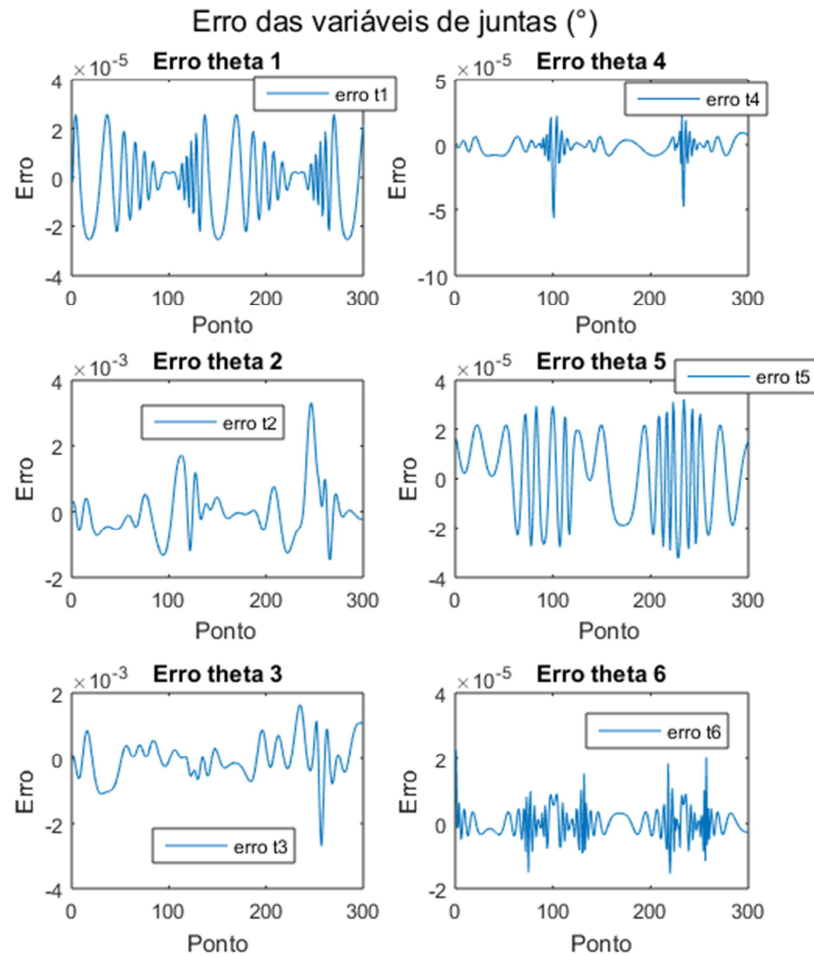


Figura 42 - Erro no valor calculado para as juntas ao longo da trajetória

Dada a pequena variação nos valores dos erros das juntas, espera-se que não hajam variações significativas nos erros de posicionamento do pulso esférico e do efetuador. Os valores dos erros referentes à posição do efetuador encontram-se plotados nas figuras 43 a 46, e na Figura 47 foram plotados os erros referentes à sua orientação.

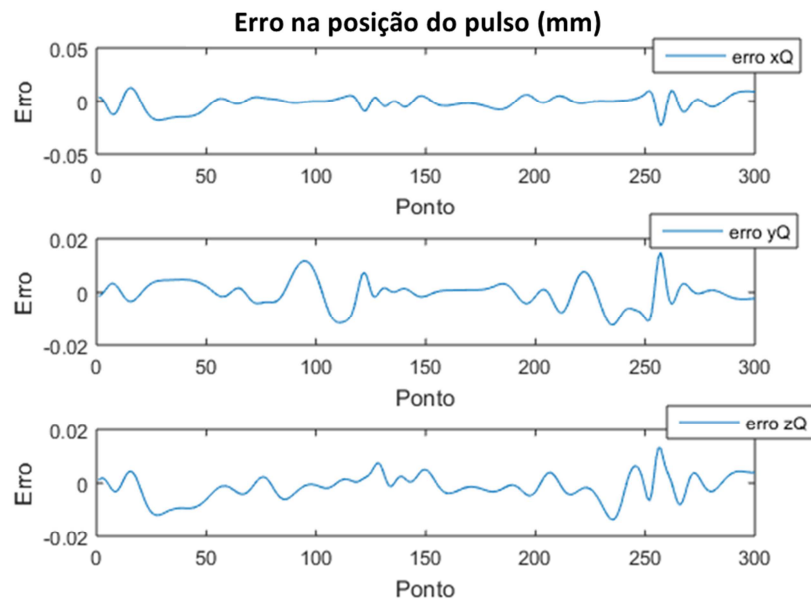


Figura 43 - Erro na posição do pulso nas coordenadas x, y e z

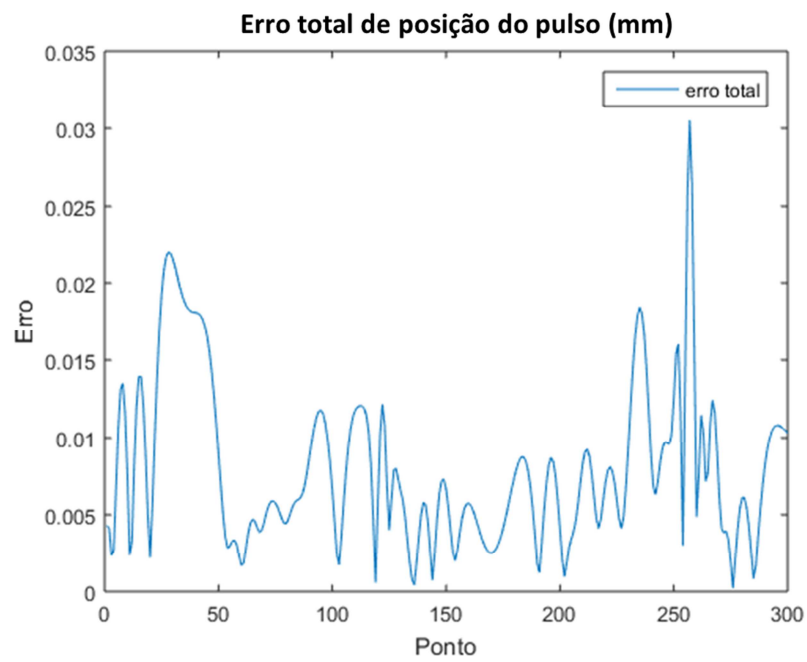


Figura 44 - Erro total de posição do pulso

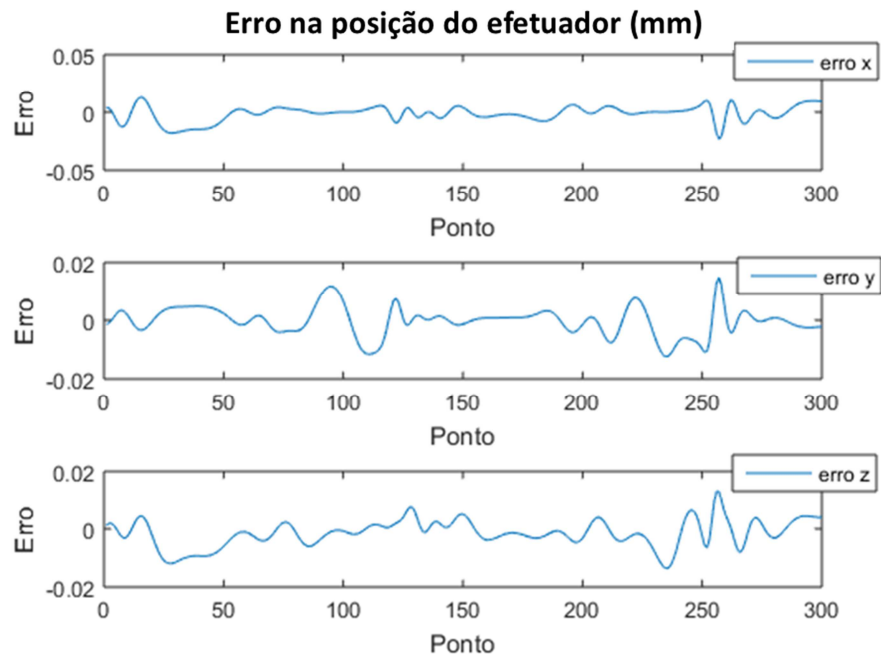


Figura 45 - Erro na posição do efetuador nas coordenadas x, y e z

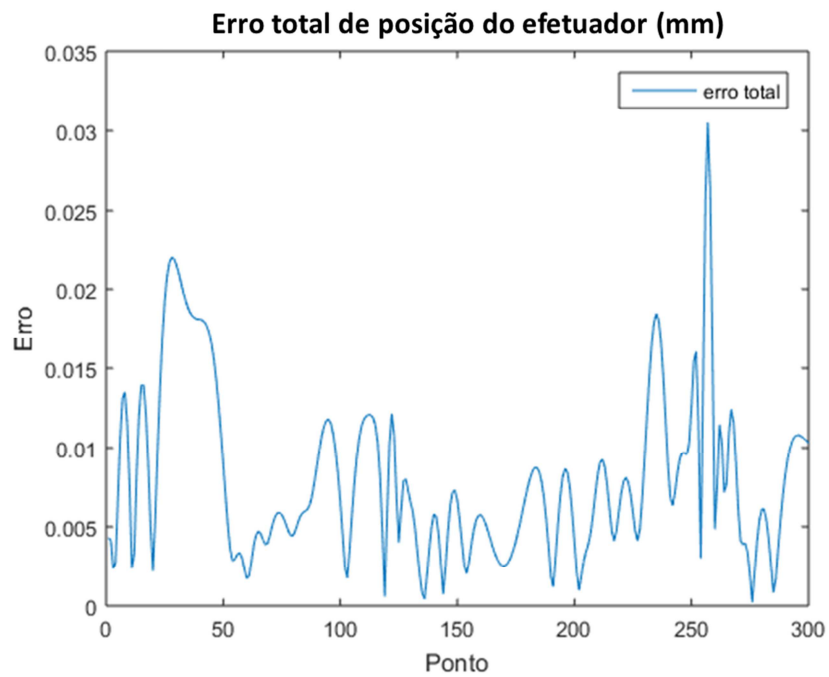


Figura 46 - Erro total de posição do efetuador

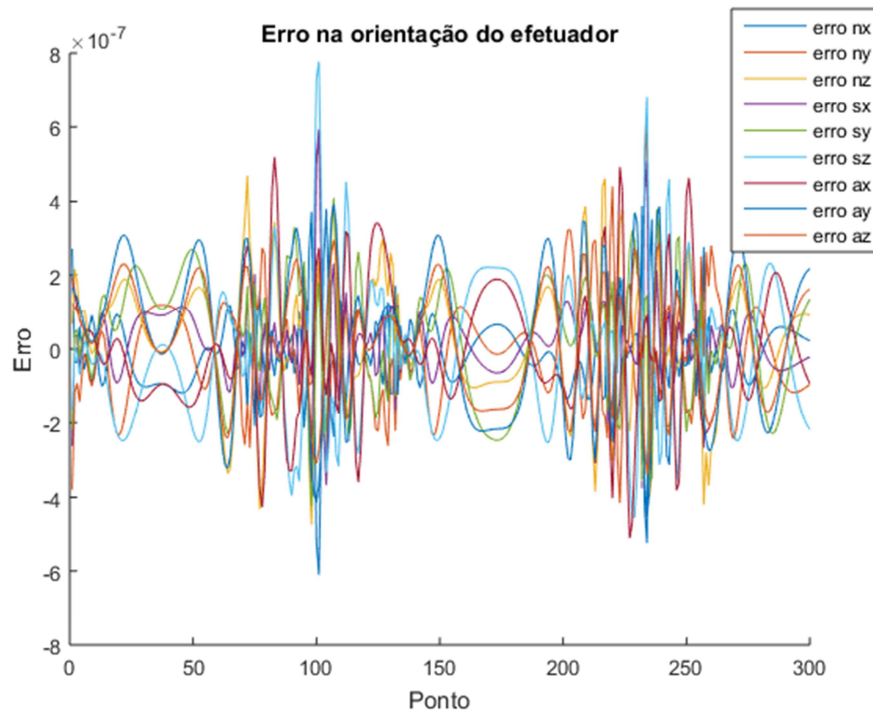


Figura 47 - Erro na orientação do efetuador

A partir da comparação entre os resultados, se verifica um pequeno aumento no valor do erro da posição do efetuador da rede original para a nova rede, com erros médios respectivos de 0,006 e 0,008 milímetros.

Os resultados obtidos através do uso do algoritmo genético são considerados promissores por representarem um método sistemático na definição da topologia de uma rede neural, no lugar dos métodos de tentativa e erro antes utilizados.

Se observa que, em termos de tempo, a definição da topologia da rede neural através do algoritmo genético é feita mais rapidamente e sem a necessidade de supervisão, além disso ela não demanda nenhum conhecimento prévio do utilizador quando o mesmo analisar o desempenho das redes neurais durante o treinamento.

6. CONCLUSÕES

O presente trabalho teve como objetivo apresentar uma alternativa à solução da cinemática inversa de manipuladores robóticos no lugar das ferramentas algébricas já conhecidas, usando como referência o manipulador industrial KR 10 R1420 da empresa Kuka com 6 graus de liberdade. Para isso foi proposta a simplificação do problema através do método apresentado por Pieper [14], subdividindo a busca pela solução da cinemática inversa do manipulador em duas partes (solução da posição inversa e solução da orientação inversa), e em seguida utilizando subconjuntos de redes neurais para resolução de cada uma das partes do problema geral.

Além disso, dado que a definição da topologia ótima de uma rede neural para uma aplicação específica não é uma tarefa simples, sendo feita de maneira experimental, e podendo ser um desafio mesmo para programadores mais experientes, se propôs uma forma de automatizar a busca pela arquitetura da rede neural artificial a ser utilizada através de algoritmos genéticos.

Dois conjuntos de redes neurais foram treinados, o primeiro sem o auxílio de algoritmo genéticos e o segundo com o auxílio de algoritmos genéticos. O primeiro conjunto apresentou um erro máximo de 0,023mm no posicionamento do manipulador, resultado considerado altamente satisfatório, e que comprova a possibilidade de utilização de redes neurais na solução desse tipo de problema. O segundo conjunto, auxiliado por algoritmos genéticos na definição da topologia de suas redes, apresentou um erro máximo de 0,03mm no posicionamento do manipulador, um resultado um pouco inferior ao do conjunto original, mas ainda assim muito promissor.

Ambos os resultados atestam a capacidade de redes neurais na solução de problemas complexos como a cinemática inversa de manipuladores. Além disso, o segundo conjunto comprova uma maneira semi-heurística de se definir a topologia de redes neurais através de algoritmos genéticos, um processo que se provou mais rápido que o típico método de tentativa e erro, e extremamente adequado para

programadores com poucos conhecimentos quando a definição de topologias de redes neurais.

Como possibilidades de novas pesquisas poderia se acrescentar ao algoritmo genético responsável pela definição da topologia da rede o modelo de ilhas, que explora o paralelismo explícito dos algoritmos genéticos e a independência entre seus indivíduos, ampliando o espaço de busca por soluções ótimas. Além disso, um algoritmo genético poderia ser utilizado também para buscar a melhor relação entre os pesos dos neurônios da rede neural, como uma alternativa ao método de treinamento por backpropagation, estratégia que já demonstrou resultados positivos para redes neurais que deveriam resolver problemas de classificação [10].

Já para redução do erro de posicionamento do manipulador há a possibilidade de utilização de lógica fuzzy na construção de redes neuro-fuzzy, assim como a técnica de processamento paralelo, onde redes neurais diferentes tornam-se responsáveis pela solução da cinemática inversa do manipulador em diferentes porções do volume de trabalho, como demonstrado por NUNES [11].

Além disso, a natureza dos problemas de cinemática inversa estudados em si pode ser modificada, focando-se na utilização de redes neurais capazes de solucionar a cinemática inversa de robôs manipuladores redundantes, para os quais não existem soluções algébricas diretas, mas que são de grande interesse prático quanto a precisão de manipulação, e desvio de obstáculos e configurações singulares.

BIBLIOGRAFIA

- [1] BRAGA, A. P.; CARVALHO, A. P. L. & LUDERMIR, T. B. **Redes Neurais Artificiais: Teoria e Aplicações**. Rio de Janeiro: LTC, 2000.
- [2] CRAIG, J. J. **Introduction to Robotics Mechanics and Control**, 3 ed. New Jersey: Pearson Education, 2005.
- [3] CRUZ, G. **A história da robótica até os dias de hoje**. 2013. Disponível em: <<https://goo.gl/enwx7G>>. Acesso em: 6 nov. 2017.
- [4] HUTCHINSON, S.; SPONG, M. W. & VIDYASAGAR, M. **Robot Dynamics and Control**. 2 ed. Illinois: John Wiley & Sons Inc., 2004.
- [5] JARDIM, J. G.; NASCIMENTO, R. S. S. **Reprodução da Vida**, 2 ed. Natal: EDUFRN, 2011.
- [6] JUNIOR, F. E. F. **Estudo e Implementação de Redes Neurais e Algoritmos Genéticos para Resolução de Cinemática Inversa de um Manipulador Robótico com 5 Graus de Liberdade**. Dissertação de Mestrado, UNICAMP, Campinas: 2014.
- [7] LUCAS, D. C., **Algoritmos Genéticos: uma Instrução**. 2002. Disponível em: <<https://goo.gl/2kuMGj>>. Acesso em: 18 out 2017.
- [8] MALAQUIAS, N. G. L. **Uso de Algoritmos Genéticos para p Otimização de Rotas de Distribuição**, Dissertação de Mestrado, UFU, Uberlândia: 2006.
- [9] MATOS, F. D.; NOGUEIRA, M. F. G.; PASSALIA, F. J. & ROCHA, J. Utilização de Algoritmo Genético e Redes Neurais Artificiais para Classificação de Blastocitos Bovinos. In: SBMAC (Org.). **Proceeding Series of the Brazilian Society of Applied and Computational Mathematics**, v. 4, São Paulo: SBMAC, 2016.
- [10] MOTA, T. C. **Aplicação de Algoritmos Genéticos à Definição da Arquitetura e ao Treinamento de Redes Neurais MLP**. Dissertação de Mestrado, UFRJ, Rio de Janeiro: 2007.
- [11] NUNES, R. F. **Mapeamento da Cinemática Inversa de um Manipulador Robótico Utilizando Redes Neurais Artificiais Configuradas em Paralelo**. Dissertação de Mestrado, UNESP, Ilha Solteira:2016.
- [12] **O Projeto Genoma Humano**. Disponível em: <<https://goo.gl/EL5V2a>>. Acesso em: 13 out 2017.
- [13] ORIOLO, G.; SCIAVICCO, L.; SICILIANO, B.; VILLANI, L. **Robotics: Modelling, Planning and Control**. 1 ed. Londres: Springer, 2009.

[14] PIEPER, D. L. **The Kinematics of Manipulators Under Computer Control**. Dissertação de Mestrado, Universidade de Stanford, Califórnia: 1968.

[15] ROMANO, V. F. **Robótica Industrial: Aplicação na Indústria de Manufatura e de Processos**. 1 ed. São Paulo: Edgard Blucher Ltda., 2002.

[16] ROMANO, V. F. **Apostila de Robótica**. 1 ed., Rio de Janeiro, 2002.

[17] SCHIMIDT, F. & SILVA S. R. e, **Redução de Variáveis de Entrada de Redes Neurais Artificiais a Partir de Dados de Análise de Componentes Principais na Modelagem de Oxigênio Dissolvido**. Goiânia: Química Nova, 2016.

[18] **KUKA robots for low payloads from 6kg to 22kg** <https://www.kuka.com/pt-br/services/downloads?terms=product_name:KR%2010%20R1420;Language:en:1;> Acesso em: 26 out. 2017