

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

RUAN VICTOR PELLOSO DUARTE BARROS

**SISTEMAS COMPUTACIONAIS INTELIGENTES APLICADOS AO CONTROLE DE  
UM BRAÇO MANIPULADOR ROBÓTICO SIMULADO**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO  
2017

RUAN VICTOR PELLOSO DUARTE BARROS

**SISTEMAS COMPUTACIONAIS INTELIGENTES APLICADOS AO CONTROLE DE  
UM BRAÇO MANIPULADOR ROBÓTICO SIMULADO**

Trabalho de Conclusão de Curso de graduação do curso de Engenharia de Controle e Automação da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para a obtenção do título de Bacharel.

Orientador: Prof. Dr. Márcio Mendonça

CORNÉLIO PROCÓPIO  
2017



**Universidade Tecnológica Federal do Paraná**  
**Campus Cornélio Procópio**  
**Departamento Acadêmico de Elétrica**  
**Curso de Engenharia de Controle e Automação**



## **FOLHA DE APROVAÇÃO**

**Ruan Victor Peloso Duarte Barros**

### **Sistemas computacionais inteligentes aplicados ao controle de um braço manipulador robótico simulado**

Trabalho de conclusão de curso apresentado às 18:00hs do dia 30/11/2017 como requisito parcial para a obtenção do título de Engenheiro de Controle e Automação no programa de Graduação em Engenharia de Controle e Automação da Universidade Tecnológica Federal do Paraná. O candidato foi arguido pela Banca Avaliadora composta pelos professores abaixo assinados. Após deliberação, a Banca Avaliadora considerou o trabalho aprovado.

---

Prof(a). Dr(a). Márcio Mendonça - Presidente (Orientador)

---

Prof(a). Dr(a). Rodrigo Rodrigues Sumar - (Membro)

---

Prof(a). Me(a). Marco Antonio Ferreira Finocchio - (Membro)

## RESUMO

BARROS, Ruan Victor P. D. **Sistemas computacionais inteligentes aplicados ao controle de um braço manipulador robótico simulado.** 2017. 47 f. Trabalho de Conclusão de Curso (Graduação) – Engenharia de Controle e Automação. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2017.

Este trabalho realiza o controle de um braço manipulador robótico autônomo simulado no *software* MatLab®, esse controle é feito a partir do uso de sistemas computacionais inteligentes. Utiliza-se uma Rede Neural Artificial (Perceptron Multicamadas) relacionada à calibração de câmera. Algoritmo genético, matriz jacobiana e método de Denavit-Hartenberg são usados para que seja programada a cinemática direta e inversa do braço manipulador e para que o mesmo seja simulado. Realiza-se a união entre a câmera e o braço robótico simulados visando a conclusão do objetivo de forma sinérgica. Tem-se assim, a capacidade de realizar a detecção de determinado ponto, por meio da câmera, e alcançá-lo utilizando o braço manipulador robótico. Apresenta-se nos resultados os erros obtidos entre o ponto alcançado e o ponto desejado e o tempo de resposta do sistema. Ao final do trabalho, são exibidas as conclusões e os trabalhos futuros propostos.

**Palavras-chave:** Braço manipulador robótico autônomo. Calibração de câmera. Visão computacional. Sistemas computacionais inteligentes. Cinemática de manipuladores robóticos.

## ABSTRACT

BARROS, Ruan Victor P. D. **Intelligent computer systems applied to the control of a robotic arm.** 2017. 47 f. Monograph – Control and Automation Engineering. Federal University of Technology - Paraná. Cornélio Procópio, 2017.

This work performs the control of a simulated autonomous robotic manipulator arm in the MatLab® software, both simulation and control are performed using intelligent computational systems. An Artificial Neural Network (Multilayer Perceptron) related to camera calibration is used. Genetic algorithm, Jacobian matrix and Denavit-Hartenberg method are used to program the direct and inverse kinematics of the manipulator arm and to simulate it. The union between the camera and the simulated robotic arm is performed in order to conclude the objective in a synergistic way. It has thus the ability to perform the detection of certain point by means of the camera, and achieve it using the robotic manipulator arm. The results presented are the errors obtained between the point reached and the desired point and the system response time. At the end of the paper, the proposed conclusions and future work are presented.

**Keywords:** Autonomous robotic arm. Camera calibration. Computer Vision. Intelligent computational systems. Robotics kinematics.

## LISTA DE TABELAS

|   |    |
|---|----|
| Tabela 1 - Topologias para calibração I .....       | 33 |
| Tabela 2 - Tempo e erros do algoritmo genético..... | 36 |
| Tabela 3 - Topologias para calibração II .....      | 37 |
| Tabela 4 - Método 1 x Método 2.....                 | 40 |

## LISTA DE FIGURAS

|  |    |
|--|----|
| Figura 1 - Exemplo de braço manipulador robótico.....                      | 10 |
| Figura 2 - Exemplo de nuvem de pontos.....                                 | 14 |
| Figura 3 - Esquemático juntas robóticas .....                              | 16 |
| Figura 4 - Funcionamento do algoritmo de Harris .....                      | 20 |
| Figura 5 - Neurônio artificial .....                                       | 21 |
| Figura 6 - Função linear .....   | 22 |
| Figura 7 - Função sigmoide.....  | 22 |
| Figura 8 - Topologia de um PMC .....                                       | 23 |
| Figura 9 - Funcionamento algoritmo genético .....                          | 25 |
| Figura 10 - <i>Grid</i> .....  | 26 |
| Figura 11 - Pontos encontrados pelo algoritmo de Harris .....              | 27 |
| Figura 12 - Pontos de teste .....  | 28 |
| Figura 13 - Simulação do braço manipulador robótico .....                  | 31 |
| Figura 14 - Controle da velocidade .....                                   | 32 |
| Figura 15 - Histograma dos erros I.....                                    | 34 |
| Figura 16 - Teste da RNA I.....  | 35 |
| Figura 17 - Histograma dos erros II.....                                   | 38 |
| Figura 18 - Teste da RNA II.....   | 39 |
| Figura 19 - Representação da simulação do braço manipulador robótico ..... | 40 |

## SUMÁRIO

|  |    |
|--|----|
| <b>1 INTRODUÇÃO</b> .....  | 7  |
| 1.1 Introdução ao tema .....   | 7  |
| 1.2 Objetivos gerais .....   | 8  |
| 1.3 Objetivos específicos.....   | 8  |
| 1.4 Estrutura do trabalho .....  | 8  |
| <b>2 PROBLEMA</b> .....  | 10 |
| 2.1 Problema analisado.....  | 10 |
| 2.2 Justificativa.....   | 11 |
| 2.3 Trabalhos correlatos .....   | 11 |
| <b>3 FUNDAMENTAÇÃO TEÓRICA</b> .....   | 13 |
| <b>3.1 Cinemática de manipuladores robóticos</b> .....   | 13 |
| 3.1.1 Cinemática inversa .....   | 13 |
| 3.1.2 Cinemática direta e Denavit-Hartenberg .....   | 15 |
| 3.1.3 Matriz Jacobiana .....   | 17 |
| <b>3.2 Visão computacional</b> .....   | 18 |
| 3.2.1 Calibração da câmera .....   | 18 |
| 3.2.2 Algoritmo de Harris.....   | 19 |
| <b>3.3 Sistemas computacionais inteligentes</b> .....  | 20 |
| 3.3.1 Redes Neurais Artificiais .....  | 21 |
| 3.3.2 Algoritmos genéticos .....   | 24 |
| <b>4 METODOLOGIA</b> .....   | 26 |
| 4.1 Calibração de câmera e resolução da cinemática inversa .....   | 26 |
| 4.2 Resolução da cinemática direta.....  | 30 |
| <b>5 RESULTADOS</b> .....  | 33 |
| <b>6 CONCLUSÃO</b> .....   | 42 |
| <b>REFERÊNCIAS</b> .....   | 43 |
| <b>ANEXO A – Datasheet do sensor sugerido para obtenção das informações relacionadas ao eixo z</b> ..... | 45 |

## 1 INTRODUÇÃO

Nesta seção, apresenta-se uma introdução sobre os principais assuntos tratados no trabalho, como robótica, manipuladores robóticos e visão computacional. Logo após, define-se os objetivos gerais e os objetivos específicos propostos, que tem como intuito nortear o trabalho e ao seu final permitir uma análise da eficácia do mesmo. Em seguida, são informados os conteúdos que estarão presentes em cada uma das seções seguintes.

### 1.1 Introdução ao tema

A pesquisa na área da robótica tem buscado encontrar soluções para as necessidades técnicas de robôs aplicados em atividades realizadas pelo homem. Robôs médicos, cozinheiros, construtores serão cada vez mais comuns no dia-a-dia das pessoas. A evolução dos campos de aplicação e sua sofisticação tem influenciado tópicos de pesquisa na comunidade de robótica (GARCIA *et al.*, 2007).

De um modo geral, os robôs podem ser úteis, mas precisam ser programados (para a maioria dos robôs empregados na indústria). Nesse contexto, a robótica é um assunto interdisciplinar que se beneficia das engenharias mecânica, elétrica e eletrônica, da ciência da computação, das ciências cognitivas, da biologia, e de muitas outras disciplinas (NIKU, 2013).

Os avanços no campo da robótica têm influenciado fortemente a produtividade e eficiência da indústria de automação. Os robôs são implantados em indústrias para executar vários trabalhos, como cortar, soldar, montar, picar e mover (AJWAD *et al.*, 2015).

A robótica industrial visa aperfeiçoar a performance dos manipuladores robóticos, o mesmo é composto de *links* unidos por atuadores comandados por um sistema de controle (SÁNCHEZ *et al.*, 2016).

Os robôs utilizados nesta área, em sua maioria, são braços manipuladores robóticos, que realizam suas determinadas funções recorrendo somente a programação ou, em casos mais complexos, utilizando como orientação a imagem de uma câmera que deve estar previamente calibrada.

A calibração de câmeras é essencial para a visão computacional, sendo esta uma ciência responsável pelo modo como uma máquina enxerga, ou seja, o jeito

como um computador vê o ambiente a sua volta. É realizada a extração de características importantes através de um processamento digital, principalmente a relação entre as coordenadas globais (reais) e de câmera (imagem) (GONZALEZ; WOODS, 2000).

Visto a importância da aquisição e tratamento de imagens associados a um braço manipulador robótico, este trabalho visa a programação de ambos, em menor escala, porém de maneira que possa ser ampliada para que seja possível a realização das atividades citadas.

## 1.2 Objetivos gerais

Realizar o controle de um braço manipulador robótico autônomo simulado no software MatLab®, utilizando sistemas computacionais inteligentes em suas etapas de desenvolvimento, de forma que o sistema seja capaz de localizar um ponto qualquer dentro um espaço delimitado através de visão computacional e de traçar uma trajetória para que um braço manipulador robótico chegue a este ponto desejado, de maneira eficiente e com velocidade controlada durante todo o trajeto.

## 1.3 Objetivos específicos

- Identificar os pontos de vértice do *grid* (área com quadrados brancos e pretos alternados e com geometria conhecida, que serve como referência para a imagem da câmera);
- Relacionar os pontos da visão computacional com os pontos reais;
- Solucionar o problema da cinemática inversa do braço robótico;
- Modelar a cinemática do braço robótico;
- Traçar a trajetória do braço robótico;
- Realizar o controle da velocidade.

## 1.4 Estrutura do trabalho

A estrutura deste trabalho foi elaborada da seguinte maneira: a Seção 2 define o problema estudado e justifica a escolha do mesmo. A Seção 3 fornece toda a base teórica necessária para o entendimento dos métodos utilizados, isso se faz por

meio da fundamentação de temas como visão computacional, sistemas computacionais inteligentes e cinemática de manipuladores robóticos. A Seção 4 exhibe a metodologia utilizada, definindo os passos seguidos para a resolução do problema apresentado. A Seção 5 mostra e analisa os resultados obtidos com os algoritmos da Seção 4. Por fim, a Seção 6 realiza as conclusões do que foi estudado e indica futuros trabalhos.

## 2 PROBLEMA

Esta seção elucida o problema estudado, em seguida, aborda a justificativa da elaboração deste trabalho, ou seja, a motivação para o início do estudo na área.

### 2.1 Problema analisado

O presente trabalho tem como propósito utilizar uma Rede Neural Artificial (RNA) para detectar a posição de determinados pontos em um *grid*. Por meio do modelo Denavit-Hartenberg (DH) e de um algoritmo genético, guia-se o braço robótico até a posição de um desses pontos com uma velocidade controlada. O ponto pode representar algum parafuso que deve ser rosqueado, alguma peça que deve ser movida ou quaisquer outros objetivos que possam ser cumpridos por essa ferramenta. Um exemplo de braço manipulador robótico é exposto na Figura 1.

Figura 1 - Exemplo de braço manipulador robótico



Fonte: <[https://www.jaycar.com.au/medias/sys\\_master/images/8948347011102/robot-arm-kit-with-controllerImageMain-515.jpg](https://www.jaycar.com.au/medias/sys_master/images/8948347011102/robot-arm-kit-with-controllerImageMain-515.jpg)> acesso em 24/10/2017

Desse modo, busca-se detectar a posição de um objeto utilizando uma câmera e alcançá-lo com um braço robótico, ou seja, o desenvolvimento de um braço manipulador robótico autônomo.

## 2.2 Justificativa

O que motivou a realização desse trabalho foi o crescimento da área industrial e a constante substituição do trabalho humano pelo trabalho realizado por máquinas, além dos possíveis usos domésticos de robôs. Essas aplicações, em muitos casos, utilizam a relação entre um manipulador robótico e a visão computacional para que se tornem viáveis ou mais eficientes.

Na indústria, os braços robóticos são indispensáveis, sem eles a grande demanda de produtos pelos consumidores não poderia ser cumprida. A tarefa dos engenheiros é criar um sistema inteligente, mas simples, que possa ser usado para controlar os braços robóticos (JUSOH *et al.*, 2012).

A calibração de câmeras é primordial em visão computacional, e possui diversas aplicações nas áreas de fabricação, inspeção submarina, medicina etc. (MENDONÇA, 2003).

Além das possíveis aplicações, este trabalho engloba diferentes áreas do conhecimento estudadas durante o curso de Engenharia de Controle e Automação, com ênfase nas vistas nas disciplinas de sistemas inteligentes, robótica e dispositivos de manipulação robótica.

## 2.3 Trabalhos correlatos

Buscando a validação deste trabalho, de forma a mostrar a relevância dos temas tratados, apresenta-se nesta seção alguns trabalhos correlatos, que utilizam técnicas similares para calibração de câmera e/ou desenvolvimento da cinemática de manipuladores robóticos.

Shheibia (2001) desenvolve um braço robótico com três graus de liberdade, unido à um tabuleiro de xadrez móvel que auxilia a movimentação do braço, desta forma é possível a movimentação das peças. A abordagem utilizada na concepção do sistema foi orientada a Agentes Inteligentes baseados em lógica *Fuzzy*.

Mashhadany (2012) utiliza uma arquitetura cognitiva para gerar o movimento do robô PUMA 560. Foi utilizada uma ANFIS (*Adaptive Network Based Fuzzy Inference System*), união de conceitos de lógica *Fuzzy* e RNA. A simulação do projeto foi feita no ambiente virtual.

Nunes (2016) mapeia a cinemática inversa de um braço manipulador robótico de 5 graus de liberdade, utilizando Redes Neurais Artificiais do tipo Perceptron Multicamadas. Além disso, é realizado o planejamento da trajetória baseado na posição do efetor.

### 3 FUNDAMENTAÇÃO TEÓRICA

Nesta seção é apresentada a fundamentação teórica utilizada no desenvolvimento do trabalho, os temas são apresentados na seguinte sequência: cinemática de manipuladores robóticos, visão computacional e sistemas computacionais inteligentes.

#### 3.1 Cinemática de manipuladores robóticos

Um manipulador possui diversos propósitos, é formado por  $n$  graus de liberdade que devem ter seus movimentos controlados de forma sincronizada, esses graus de liberdade determinam a cinemática do robô. A relação entre a orientação e localização do efector e os valores de seus *links* constituem a cinemática de um manipulador (CARDONA, 2015).

##### 3.1.1 Cinemática inversa

Cinemática inversa, é um problema que visa obter os parâmetros das juntas para uma certa orientação e posição do efector nas coordenadas globais, ou seja, encontrar os ângulos necessários para que o manipulador robótico alcance a posição pretendida (GUEZ; AHMAD, 1988).

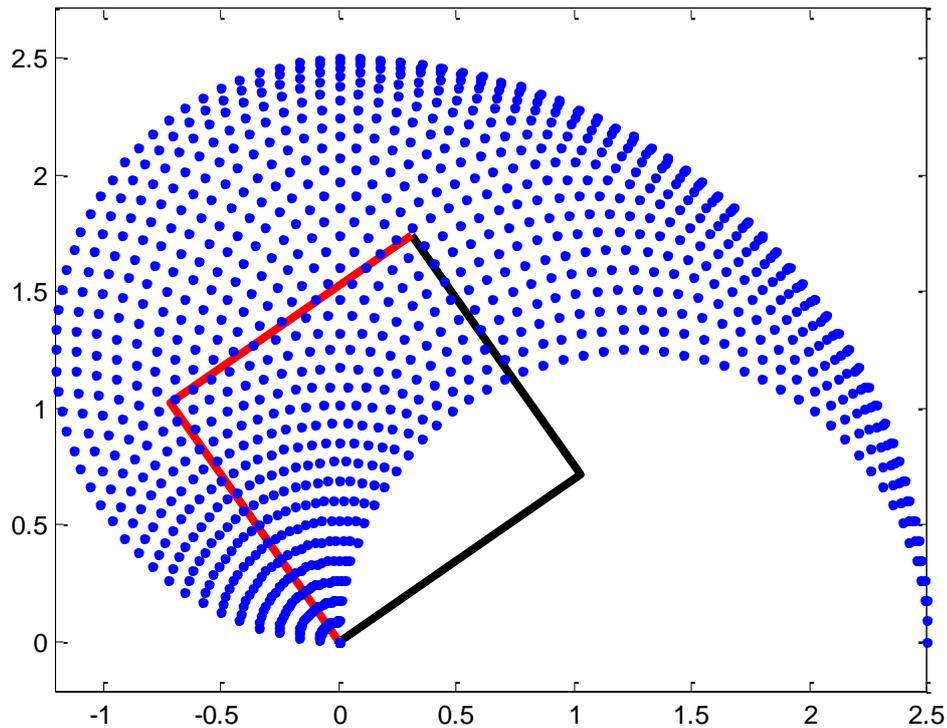
Pode-se utilizar a solução da cinemática inversa no controle do manipulador para determinar ou alterar sua posição conforme necessário (CRAIG, 1986).

Com a solução da cinemática inversa, por exemplo, podem ser calculados os parâmetros das juntas de forma que seja possível posicionar o braço robótico para pegar uma peça ou realizar uma função em qualquer local que esteja ao seu alcance, mesmo que o local seja alterado.

A Figura 2, obtida através do MatLab®, mostra um exemplo de braço robótico simulado com dois graus de liberdade e sua nuvem de pontos (discretização dos pontos que podem ser alcançados pelo braço). Para a criação da nuvem, limitou-se o ângulo do primeiro braço de  $0^\circ$  a  $90^\circ$  e o do segundo de  $0^\circ$  a  $180^\circ$ , os braços têm tamanho adimensional de 1,25 cada. O braço representado pelos traços vermelhos e o representado pelos traços pretos atingem o mesmo ponto, mostrando que mesmo

em um caso com apenas duas dimensões pode existir mais de uma solução para a cinemática inversa. Nesse caso, conforme aumenta-se os graus de liberdade, o número de soluções também cresce (CRAIG, 1986).

**Figura 2 - Exemplo de nuvem de pontos**



Fonte: autoria própria.

Há diversas formas de se solucionar a cinemática inversa de um manipulador, entre as soluções mais usadas estão os métodos analíticos e os métodos numéricos iterativos. Os métodos analíticos buscam encontrar todas as soluções, eles são complexos e não existe garantia de que o resultado possa ser encontrado. Por esse motivo são indicados, em geral, para robôs simples, ou seja, aqueles que possuem um grande número de parâmetros de DH nulos. Um exemplo é a função *solve* do MatLab®.

Os métodos numéricos iterativos convergem para uma solução possível, podendo existir diversas outras, além disso, não atingem com exatidão a solução, realizando uma aproximação. Por outro lado, possuem complexidade computacional mais baixa do que os métodos analíticos e, mesmo que não exista uma solução possível, é buscada a mais próxima disso. Um exemplo é o uso de algoritmos genéticos.

### 3.1.2 Cinemática direta e Denavit-Hartenberg

Um problema dos braços robóticos é: dados os parâmetros das juntas e os deslocamentos dos braços, achar a posição final do manipulador. Este problema é chamado de cinemática direta (GUEZ; AHMAD, 1988).

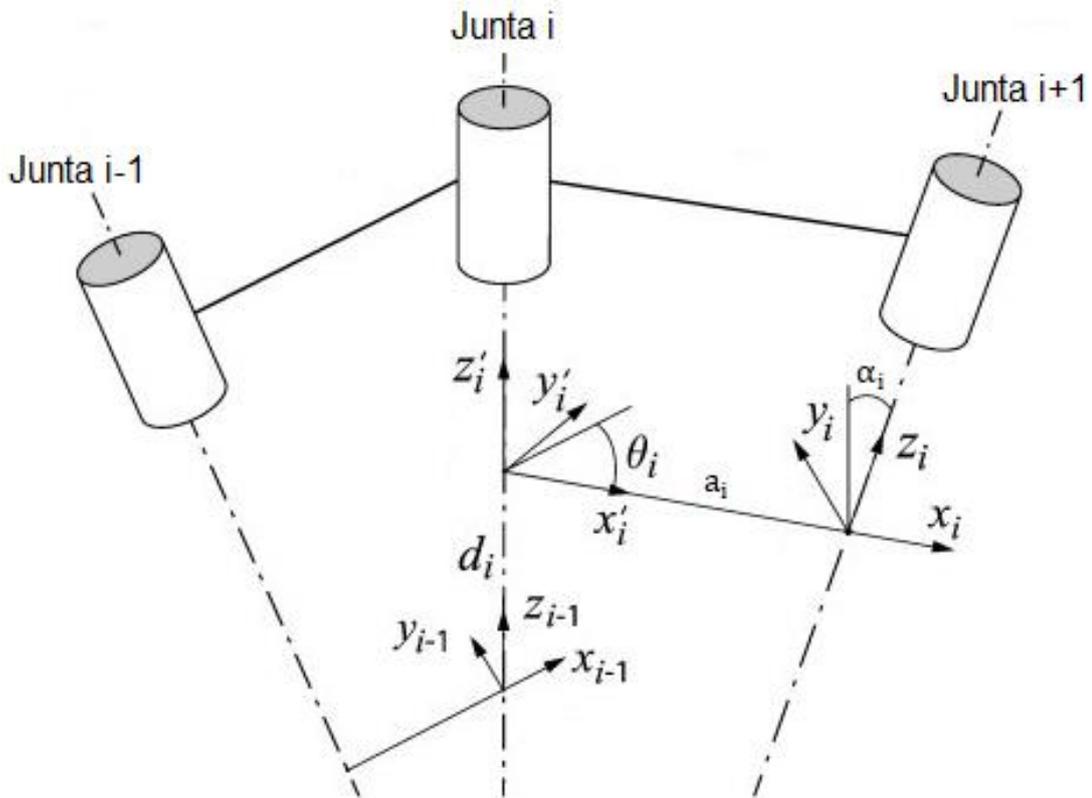
O modelo de Denavit-Hartenberg é uma ferramenta muito utilizada para solucionar a cinemática de sistemas articulados com  $n$  graus de liberdade (DENAVID, 1955).

Soluções através da cinemática direta podem ser obtidas através do modelo de DH, que é baseado em quatro parâmetros ( $d$ ,  $a$ ,  $\alpha$ ,  $\theta$ ), utilizando os mesmos em transformações matriciais homogêneas consecutivas. Desta forma, possível que seja encontrada a posição do efector de um braço robótico, por exemplo.

Cada movimento independente que uma articulação pode fazer relacionado à anterior é denominado grau de liberdade. A quantidade de graus de liberdade de um manipulador é o número mínimo de variáveis de posição necessárias para localizar todas as partes do mecanismo (CRAIG, 1986).

A notação de Denavit-Hartenberg é um método para descrever a posição da extremidade do manipulador baseado nas características do próprio braço. A Figura 3 representa dois braços de um manipulador robótico no qual  $i-1$  é anterior a  $i$  e a cinemática é dada pelas características apresentadas.

Figura 3 - Esquemático juntas robóticas



Fonte: adaptado de <<https://www.quora.com/Robotics-What-is-the-best-resource-to-understand-Denavit%E2%80%93Hartenberg-parameters>> acesso em 24/10/17.

Analisando a figura acima e as variáveis nela contidas, tem-se que (CRAIG, 1986):

- $a_i$  refere-se à distância entre  $z_{i-1}$  e  $z_i$  percorrendo o eixo  $x_i$ ;
- $\alpha_i$  refere-se ao ângulo entre os  $z_{i-1}$  e  $z_i$  tendo como referência o eixo  $x_i$ ;
- $d_i$  refere-se à distância entre  $x_{i-1}$  e  $x_i$  percorrendo o eixo  $z_{i-1}$ ;
- $\theta_i$  refere-se ao ângulo entre  $x_{i-1}$  e  $x_i$  tendo como referência o eixo  $z_{i-1}$ .

A partir das equações (1) e (2) é obtida a matriz de transformação para um braço do manipulador, enquanto que utilizando a equação (3) tem-se a matriz de transformação para o manipulador completo.

$$A_{i-1}^i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{i-1}^i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cdot \cos\alpha_i & \sin\theta_i \cdot \sin\alpha_i & a_i \cdot \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cdot \cos\alpha_i & -\cos\theta_i \cdot \sin\alpha_i & a_i \cdot \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$A_0^n = A_0^1 \cdot A_1^2 \cdot A_{n-1}^n \quad (3)$$

Desta forma os três primeiros termos da última coluna da matriz  $A_0^n$  são, respectivamente, as posições da extremidade do manipulador nos eixos x, y e z. Desta forma é possível solucionar a cinemática direta (CRAIG, 1986).

### 3.1.3 Matriz Jacobiana

Uma especificação da trajetória do braço é um pré-requisito para o sucesso na aplicação de um manipulador para muitas tarefas. Soldagem, pintura, escolha de peças em esteiras e colagem são algumas tarefas que necessitam tanto das especificações temporais quanto das espaciais de uma trajetória (HOLLERBACH; SAHAR, 1984).

A matriz jacobiana é a derivada parcial obtida a partir das equações geométricas que definem a posição final de um manipulador. Tomando como exemplo um manipulador robótico simples, com dois graus de liberdade e em duas dimensões é calculada a matriz jacobiana referente ao mesmo.

$$x_f = l_1 \cdot \cos\theta_1 + l_2 \cdot \cos(\theta_1 + \theta_2); y_f = l_1 \cdot \sin\theta_1 + l_2 \cdot \sin(\theta_1 + \theta_2) \quad (4)$$

Calculando a derivada de ambas:

$$\begin{aligned} dx_f &= -l_1 \cdot d\theta_1 \cdot \sin\theta_1 - l_2 \cdot (d\theta_1 + d\theta_2) \cdot \sin(\theta_1 + \theta_2); \\ dy_f &= l_1 \cdot d\theta_1 \cdot \cos\theta_1 + l_2 \cdot (d\theta_1 + d\theta_2) \cdot \cos(\theta_1 + \theta_2) \end{aligned} \quad (5)$$

Passando para a forma matricial:

$$\begin{bmatrix} dx_f \\ dy_f \end{bmatrix} = \begin{bmatrix} -l_1 \cdot \text{sen}\theta_1 - l_2 \cdot \text{sen}(\theta_1 + \theta_2) & -l_2 \cdot \text{sen}(\theta_1 + \theta_2) \\ l_1 \cdot \text{cos}\theta_1 + l_2 \cdot \text{cos}(\theta_1 + \theta_2) & l_2 \cdot \text{cos}(\theta_1 + \theta_2) \end{bmatrix} \begin{bmatrix} d\theta_1 \\ d\theta_2 \end{bmatrix} \quad (6)$$

No qual:

- $dx_f, dy_f$ : o movimento diferencial da extremidade do manipulador;
- $l_1, l_2$ : o tamanho de cada braço;
- $\theta_1, \theta_2$ : o ângulo de cada grau de liberdade;
- $d\theta_1, d\theta_2$ : o movimento diferencial das articulações.

Através das equações (4), (5) e (6) é possível traçar a trajetória do braço manipulador robótico do seu ponto atual em direção ao ponto desejado, portanto a matriz jacobiana indica o caminho a ser utilizado na movimentação do manipulador (NIKU, 2013).

### 3.2 Visão computacional

A visão computacional é inspirada na própria visão humana e seu processamento pelo cérebro, que agrupa e organiza os estímulos visuais. Ela é definida como o estudo das imagens obtidas por câmeras e do tratamento fornecido a elas, tendo como objetivo extrair determinada informação dessa imagem. É uma das principais áreas de pesquisa em robótica e possui diversas aplicações, como por exemplo, diferenciação de grãos em uma esteira seletora, controle de processos, detecção de eventos, controle de robôs autônomos e robôs industriais.

#### 3.2.1 Calibração da câmera

A calibração de câmera é a elaboração de um modelo da imagem em relação à câmera em um sistema digital, desta forma, confirma-se a correlação entre pontos correspondentes na imagem e no mundo real, e atinge-se os parâmetros da câmera (YUAN *et al.*, 2013).

A calibração, objetiva o estabelecimento de uma relação entre as coordenadas 3D e a suas correspondentes no plano (2D), maneira como o computador realiza a análise (LENZ; TSAI, 1987).

A câmera utilizada para a obtenção das imagens é do tipo *pin-hole*, uma câmera ideal sem nenhuma distorção (GONZALEZ; WOODS, 2000).

Há a necessidade de atenuar-se as diferenças entre a imagem obtida pela câmera e a realidade, para isso são usadas RNA, que são redes de neurônios com a capacidade de aprendizado, mesmo quando se trata de uma relação não linear entre as coordenadas de mundo e as coordenadas de câmera (LYNCH *et al.*, 1999).

O ajuste dos pesos sinápticos da RNA durante a fase de treino possibilita a associação entre os pontos, ou seja, correlaciona de forma correta os dados de entrada a partir de dados já conhecidos. Desta forma, sabe-se que uma rede deste tipo é capaz de aproximar qualquer função contínua arbitrária (JIN; RUI, 2011).

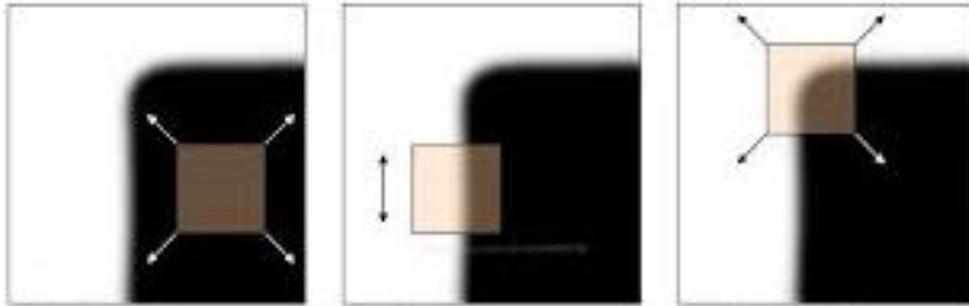
### 3.2.2 Algoritmo de Harris

Para auxiliar a calibração da câmera tem-se o algoritmo de Harris ou Harris *Corner Detector*, que foi desenvolvido em 1988 por Chris Harris e Mike Stephens através do aprimoramento do detector de canto de Moravec (HARRIS; STEPHENS, 1988).

O objetivo desse algoritmo é localizar os pontos da imagem que podem ser definidos como vértices, é comumente associado a um *grid*, metodologia usualmente utilizada em calibração de câmeras (OUYANG *et al.*, 2005).

A abordagem utilizada por Harris segmenta a imagem de forma a criar uma pequena janela utilizando apenas uma parte da imagem original, esta janela pode ser deslocada em todas as direções na parte interna da imagem, como mostra a Figura 4.

**Figura 4 - Funcionamento do algoritmo de Harris**



Fonte: <<http://twixar.me/6033>> acesso em 24/10/2017.

No primeiro caso, a janela pode se mover levemente em quaisquer direções que continua em uma região completamente preta ou com pequenas alterações no tom da cor, isso define que a região atual é uniforme e, portanto, não é um vértice.

No segundo, se movendo para a esquerda e para a direita as cores de dentro da janela serão alteradas, entretanto, caso a movimentação seja para cima ou para baixo, como apontada pelas setas, não ocorre alteração, portanto, é uma aresta, mas não um vértice.

No último caso, a janela se encontra em determinada posição que não importa a direção do movimento, sempre altera-se o padrão de cores de dentro da janela, neste caso, então, podemos definir que naquela região há um vértice.

Basicamente o algoritmo encontra a diferença de intensidade ao redor de um ponto da imagem em todas as direções e através de uma análise determina se aquele ponto é ou não um vértice. No MatLab® é possível utilizar essa ferramenta a partir do comando *detectHarrisFeatures*.

Um programa baseado no algoritmo de detecção de cantos de Harris é usado para obter as coordenadas 2D dos cantos no sistema de coordenadas de uma imagem (JIN; RUI, 2011).

### **3.3 Sistemas computacionais inteligentes**

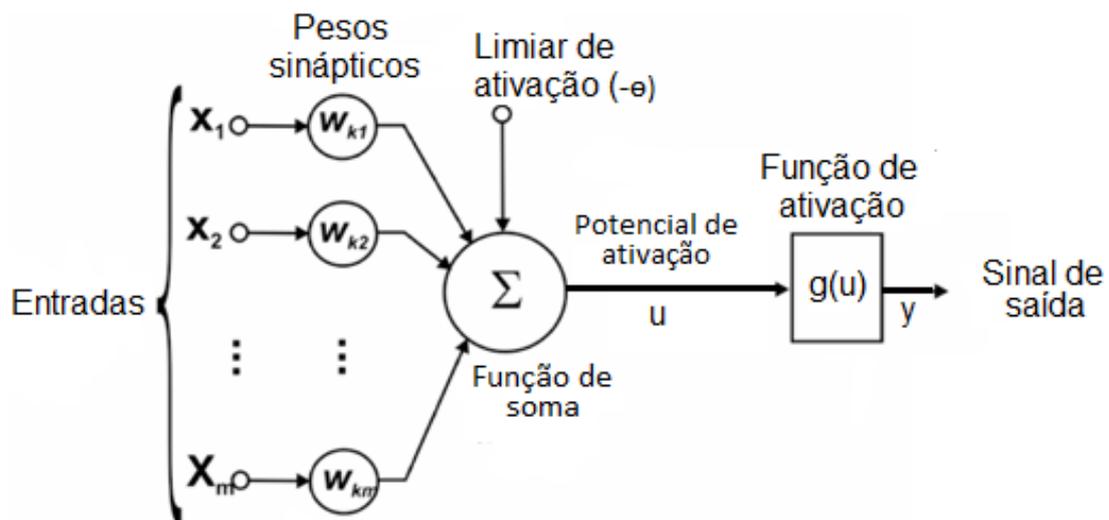
Sistemas computacionais inteligentes estão relacionados a inteligência artificial e estudam os processos envolvidos no pensamento humano, buscando recriá-los utilizando máquinas, portanto um sistema inteligente visa imitar ou multiplicar a inteligência humana para resolver problemas em geral.

### 3.3.1 Redes Neurais Artificiais

Uma RNA é um processador massivo de distribuição paralela feito de unidades de processamento simples, possuindo uma propensão natural para armazenar conhecimento empírico e torná-lo disponível para uso. As RNAs se assemelham com o cérebro em dois aspectos: o conhecimento é adquirido pela rede a partir do seu ambiente através de um processo de aprendizagem; a força de conexão entre neurônios (pesos sinápticos) é usada para armazenar o conhecimento adquirido (HAYKIN, 1999).

São constituídas de neurônios artificiais, que buscam imitar o comportamento de um neurônio biológico e, embora seja uma ferramenta bem mais simples, são estruturas lógico-matemáticas que visam a realização das mesmas funções do seu análogo orgânico, a Figura 5 mostra o seu funcionamento (DA SILVA *et al.*, 2010).

**Figura 5 - Neurônio artificial**



Fonte: adaptado de Haykin (1999).

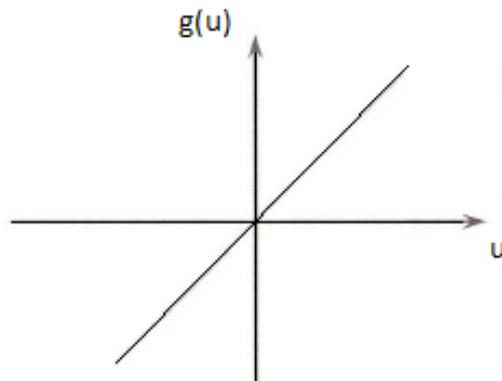
Podendo ser representado por:

$$u = \sum_{i=1}^n w_i \cdot x_i - \theta \quad (7)$$

$$y = g(u) \quad (8)$$

As funções de ativação que serão utilizadas no caso analisado são a função linear (Figura 6 e equação 8) e a função tangente sigmoide (Figura 7 e equação 9).

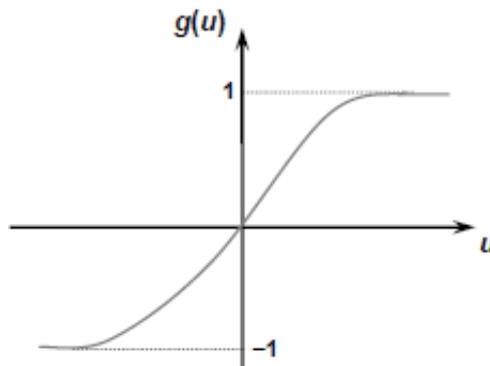
**Figura 6 - Função linear**



Fonte: Da Silva *et al.* (2010).

$$g(u) = \frac{2}{1+e^{-2.u}} + 1 \quad (9)$$

**Figura 7 - Função sigmoide**



Fonte: Da Silva *et al.* (2010).

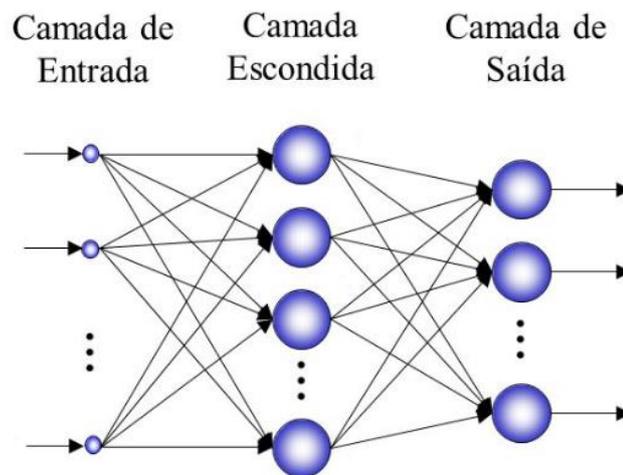
As principais características da RNA relevantes para a aplicação proposta são: a capacidade de generalizar o conhecimento adquirido, podendo realizar estimativas de soluções; o poder de relacionar as variáveis que o compõem, extraíndo assim, o relacionamento entre elas; alta tolerância a falhas (DA SILVA *et al.*, 2010).

Uma RNA é uma estrutura distribuída paralelamente na forma de um grafo direcionado com os nós sendo os elementos de processamento, e suas arestas as

conexões, funcionando como caminhos de condução instantânea de sinais em uma única direção (MENDONÇA, 2011).

A Figura 8 explica a topologia de uma RNA do tipo Perceptron Multicamadas (PMC). A camada de entrada é onde os padrões são apresentados à rede, as camadas escondidas realizam o processamento da informação e absorvem as características do sistema e a camada de saída apresenta os resultados finais.

**Figura 8 - Topologia de um PMC**



Fonte: adaptado de <[http://images.slideplayer.com.br/2/356311/slides/slide\\_29.jpg](http://images.slideplayer.com.br/2/356311/slides/slide_29.jpg)> acesso em 24/10/17.

Redes PMC têm como característica a presença de pelo menos uma camada intermediária de neurônios. Essas redes podem ser utilizadas em problemas de aproximação universal de funções, classificação de padrões, previsão de séries temporais, etc. O número de camadas intermediárias deve ser corretamente parametrizado para melhor desempenho da RNA (DA SILVA *et al.*, 2010).

Para o processo de treinamento de um PMC é utilizado o algoritmo *backpropagation*. Esse processo é feito através de aplicações sucessivas de duas fases bem específicas: *forward* (inserção das amostras na rede para obter o avanço na mesma) e *backward* (baseado no erro entre a primeira fase e a resposta desejada ajusta-se os parâmetros da rede), no treinamento usa-se o algoritmo de Levenberg-Marquardt, que se trata de um método de otimização que acelera a convergência do *backpropagation* (DA SILVA *et al.*, 2010).

Desta forma, redes neurais podem ser utilizadas para atenuar as discordâncias entre a realidade e a imagem obtida pela câmera, realizando, desse modo, a calibração.

### 3.3.2 Algoritmos genéticos

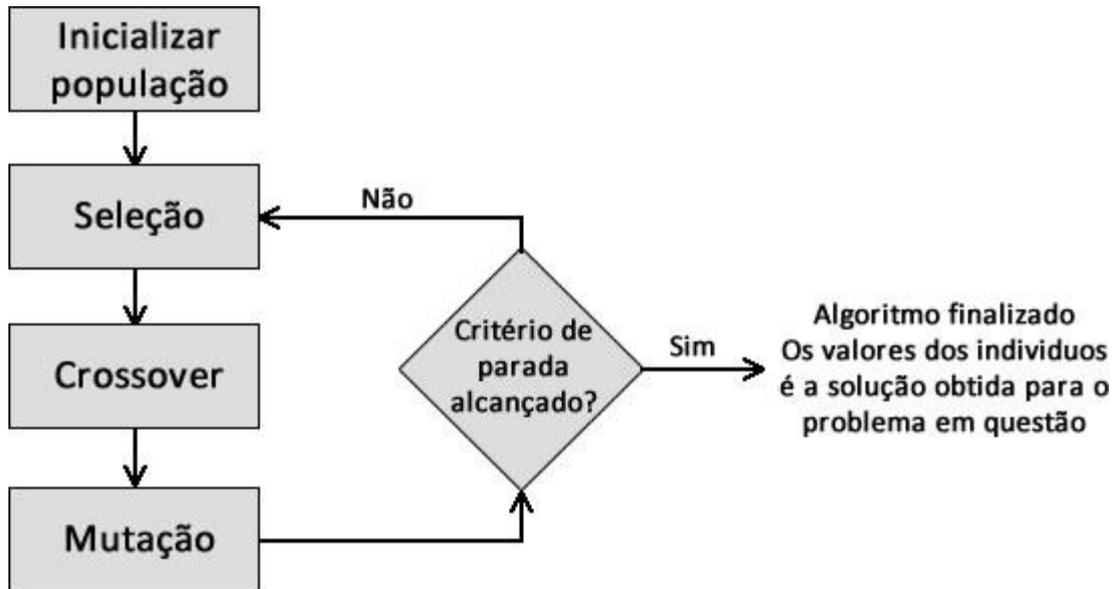
Os algoritmos genéticos são métodos heurísticos para a solução de problemas que se baseiam na teoria da evolução de Charles Darwin, foram propostos por John Holland, em 1975, utilizando o conceito de cromossomos como estrutura básica para representar possíveis soluções de determinado problema (SILVA, 2008).

Para o melhor entendimento de algoritmos genéticos, é necessário compreender alguns conceitos que formam sua estrutura, são eles:

- **Indivíduo:** possui o código genético que é uma representação do espaço de busca do problema, na forma de sequências de *bits* (GOLDBERG, 1989);
- **Seleção:** ordenação dos indivíduos, de acordo com o objetivo do problema são atribuídas probabilidades decrescentes de serem escolhidos, proporcionais à razão entre adaptação desse indivíduo e a soma da adaptação de todos os outros da população, e assim, a escolha é realizada de acordo com essa probabilidade (LINDEN, 2008);
- **Reprodução:** os descendentes recebem metade de seu código genético da mãe e metade do pai, desta forma os melhores indivíduos são capazes de reproduzir entre si e dar origem a gerações cada vez mais aptas (*crossing-over*). Neste processo ocorrem mutações, que permitem maior variabilidade genética por meio de modificações aleatórias em *bits* elementares, impedindo assim a estagnação da população (GOLDBERG, 1989);
- **População:** conjunto de todos os indivíduos que são cogitados como possível solução (GOLDBERG, 1989).

A Figura 9 mostra um fluxograma para melhor apreensão.

Figura 9 - Funcionamento algoritmo genético



Fonte: adaptado de <<http://www.computacaointeligente.com.br/wp-content/uploads/2016/08/fluxogramaGA.jpg>> acesso em 24/10/17.

A ideia básica é de buscar a solução ótima para um problema começando de uma população inicial. A partir da reprodução destes indivíduos, chega-se a uma nova geração e a seleção dos que estiverem mais próximos da solução ótima, ou seja, mais aptos, permite levar a uma nova geração e assim por diante, até que o critério de parada seja atendido.

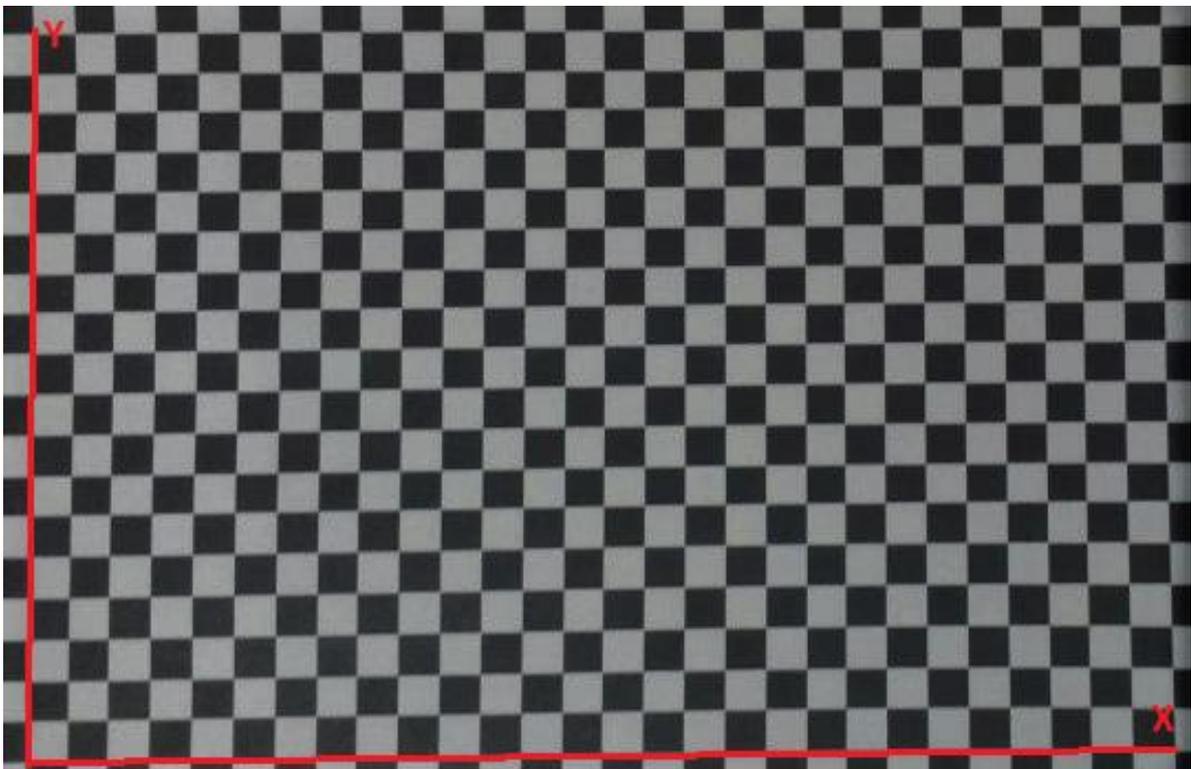
## 4 METODOLOGIA

Esta seção contém os procedimentos realizados nas simulações através do software MatLab®, desde a criação do *grid* de calibração até a modelagem da cinemática do braço manipulador robótico. São abordadas todas as etapas do seu desenvolvimento, definindo os passos a serem seguidos e mostrando como realizou-se o trabalho.

### 4.1 Calibração de câmera e resolução da cinemática inversa

Inicialmente, para a calibração de uma câmera utilizando RNAs, é necessário a obtenção dos dados que serão fornecidos para rede em seu processo de treinamento. Visto isso, criou-se um *grid* de calibração comumente empregado em situações como esta, sendo utilizado para relacionar as coordenadas de câmera com as coordenadas reais. A Figura 10 mostra o *grid* utilizado com os eixos x e y marcados, sendo que cada quadrado tem a medida de 1 cm.

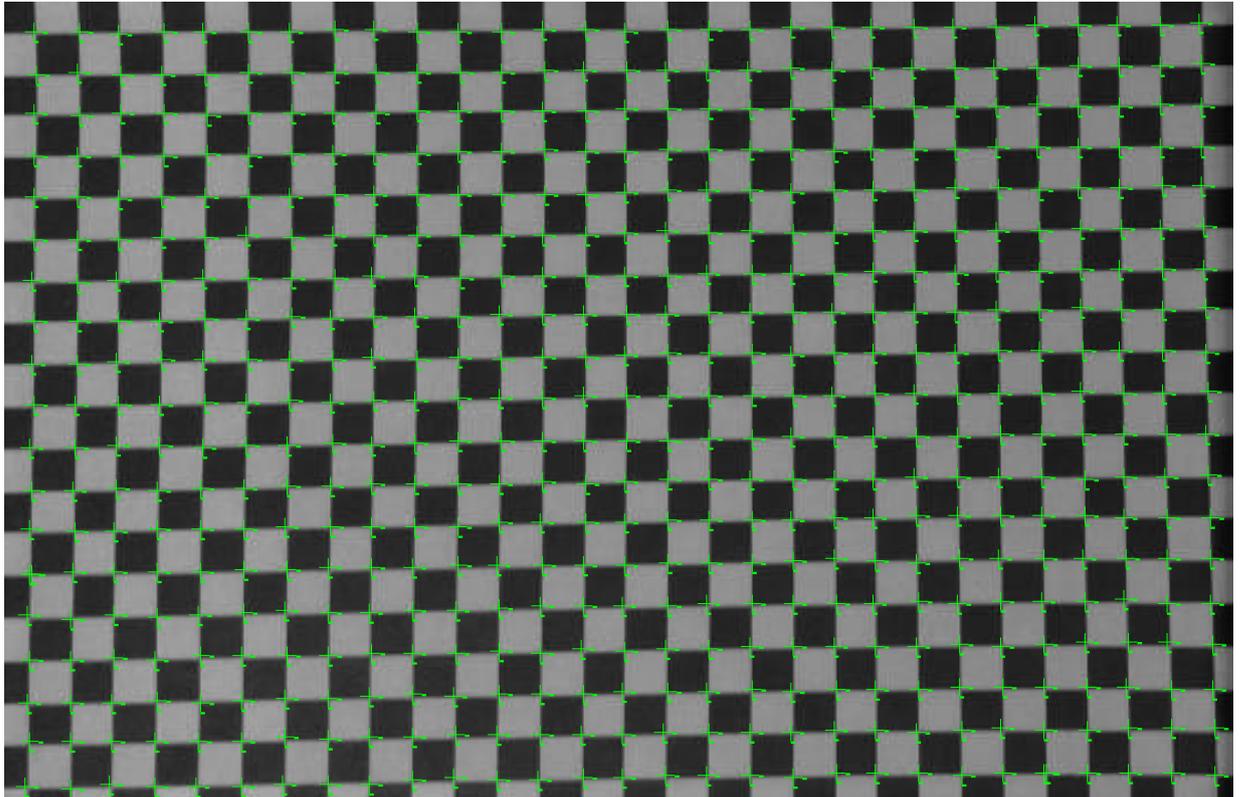
Figura 10 - *Grid*



Fonte: autoria própria.

Possuindo a imagem do *grid*, utiliza-se o algoritmo de Harris no ambiente computacional para realizar a detecção de cada um dos vértices. Na Figura 11, as marcações em verde mostram os pontos encontrados pelo algoritmo.

**Figura 11 - Pontos encontrados pelo algoritmo de Harris**



**Fonte: autoria própria.**

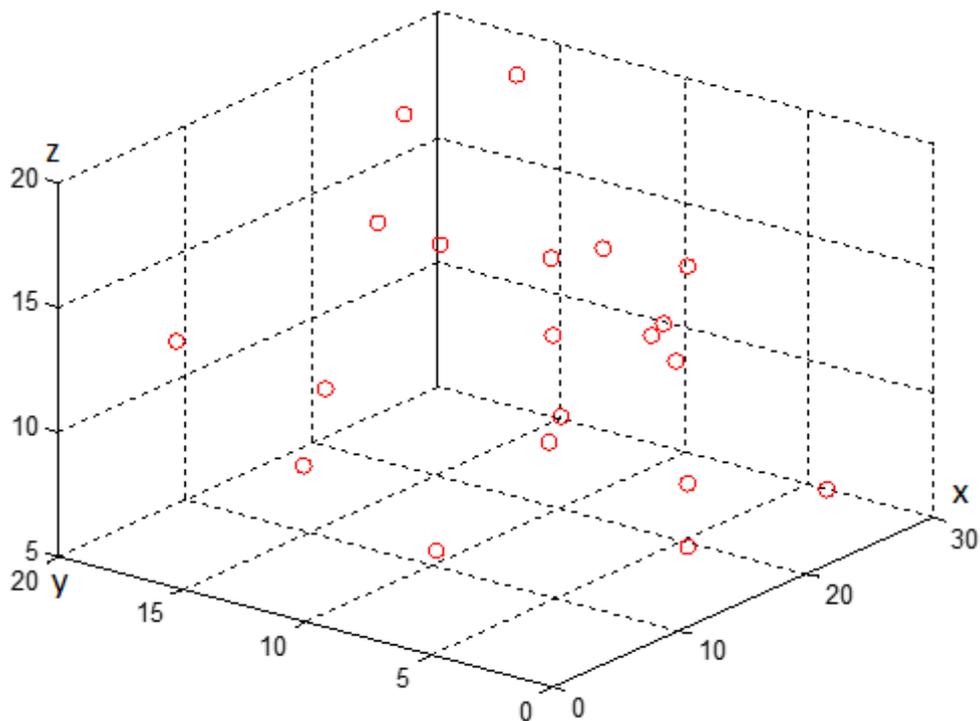
A área estipulada é de 28 cm x 18 cm, ao todo, são encontrados 551 pontos, como a aquisição desses pontos ocorre fora de ordem, é criada uma rotina de agrupamento para facilitar a manipulação dos mesmos. Aliado a isso, para completar os pontos utilizados como entrada da RNA, define-se um limite no eixo z de 20 cm, que culmina em um total de 11571 pontos de entrada. O alto número de pontos deve-se à complexidade do problema e ao esforço para obter um erro pequeno ao final das simulações.

A localização do ponto que se deseja atingir é obtida pela câmera em relação aos eixos x e y, e por um sensor em relação ao eixo z, desta forma é possível realizar a calibração em três dimensões com uma maior precisão. Um modelo de sensor que poderia ser aplicado é mostrado no Anexo A, mas há a possibilidade de

serem utilizados quaisquer sensores que cumpram a função de detectar um ponto com precisão dentro da área proposta.

Com os pontos a serem utilizados como entrada da RNA já conhecidos, estipula-se com quais a RNA deve ser testada. Utilizou-se 20 pontos escolhidos arbitrariamente para o teste, sendo eles representados pelos círculos vermelhos e estão espalhados no espaço já determinado como mostra a Figura 12, as medidas estão em centímetros.

**Figura 12 - Pontos de teste**



**Fonte: autoria própria.**

Os cálculos a seguir, tanto os relacionados à cinemática inversa quanto os relacionados à cinemática direta, são possíveis devido ao uso do modelo DH nos códigos aplicados. A matriz do modelo foi criada considerando os quatro graus de liberdade do braço manipulador, sendo dois na base (um de rotação e um linear), e mais dois lineares ao longo do manipulador como mostra a Figura 13.

A Equação 10 mostra a matriz DH do braço manipulador robótico considerado,  $\theta_{1,2,3,4}$  são os ângulos das juntas e  $a_{1,2,3,4}$  são os tamanhos dos segmentos do braço, em ordem crescente da base ao efetor.

$$\begin{aligned}
 A &= \begin{bmatrix} \cos\theta_1 * \cos(\theta_2 + \theta_3 + \theta_4) & -\cos\theta_1 * \sin(\theta_2 + \theta_3 + \theta_4) & & \\ \sin\theta_1 * \cos(\theta_2 + \theta_3 + \theta_4) & -\sin\theta_1 * \sin(\theta_2 + \theta_3 + \theta_4) & & \\ & \sin(\theta_2 + \theta_3 + \theta_4) & \cos(\theta_2 + \theta_3 + \theta_4) & \dots \\ & 0 & 0 & \dots \end{bmatrix} \\
 A &= \begin{bmatrix} \sin\theta_1 & \cos\theta_1 * (a_1 * \cos\theta_2 + a_2 * \cos(\theta_2 + \theta_3) + a_3 * \cos(\theta_2 + \theta_3 + \theta_4)) \\ \dots & -\cos\theta_1 * (a_1 * \cos\theta_2 + a_2 * \cos(\theta_2 + \theta_3) + a_3 * \cos(\theta_2 + \theta_3 + \theta_4)) \\ & 0 & a_1 * \sin\theta_2 + a_2 * \sin(\theta_2 + \theta_3) + a_3 * \sin(\theta_2 + \theta_3 + \theta_4) \\ & 0 & 1 \end{bmatrix} \quad (10)
 \end{aligned}$$

A partir desse ponto há a divisão em duas possíveis estratégias para continuar a resolução do problema, elas serão chamadas de Método 1 e Método 2. O Método 1 tem seu foco na precisão enquanto o Método 2 é focado na velocidade, ainda assim os dois métodos prezam por ambos os fatores.

- **Método 1**

Neste método, utiliza-se somente os 551 pontos de câmera obtidos em relação aos eixos x e y para o treinamento da RNA, e como saída da mesma tem-se as coordenadas, também em x e y, dos seus correspondentes reais. Os critérios de parada da rede são 1000 épocas ou um erro de  $10^{-6}$ .

Inicia-se a resolução da cinemática inversa do braço manipulador robótico. De agora em diante, aborda-se as partes *online* do código, ou seja, o tempo gasto pelos programas a seguir refletem no tempo de resposta do sistema, enquanto que as partes anteriores não têm influência nesse escopo.

Com a resposta da RNA, aliada à informação do sensor, sabe-se as coordenadas reais do ponto alvo (pontos de teste). Essas informações são fornecidas a um algoritmo genético capaz de realizar o cálculo dos quatro ângulos necessários para que o ponto seja atingido. O erro referente ao critério de parada pode ser ajustado, entretanto, quanto menor o erro maior o tempo que o algoritmo leva para encontrar a resposta final. Terminado o código, tem-se os valores dos ângulos para que determinado ponto seja atingido.

- **Método 2**

Para esse método são utilizados todos os 11571 pontos obtidos com a câmera e o sensor para o treinamento da RNA, contudo, desta vez, tem-se como saída da rede os ângulos necessários para que esses pontos sejam atingidos. Para que isso seja possível, é essencial que o algoritmo genético detecte os quatro ângulos necessários para que sejam atingidos cada um dos 11571 pontos e a rede seja treinada com essas informações. Nesse método, o critério de parada é 300 épocas.

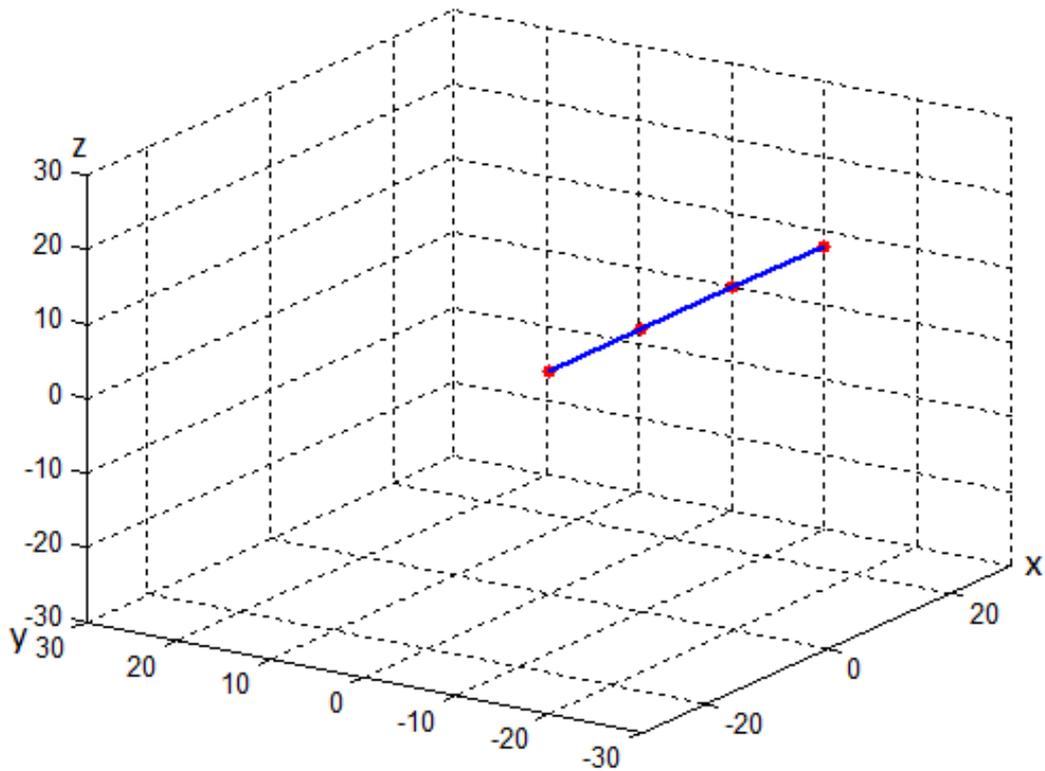
Esse processo é demorado, principalmente quando se busca um baixo erro, todavia, isso é realizado de maneira *offline*, não tendo importância o tempo decorrido. Desta forma, depois do treinamento, quando os valores dos ângulos para que determinado ponto seja atingido são solicitados à RNA, os mesmos são fornecidos de maneira rápida.

#### **4.2 Resolução da cinemática direta**

A Figura 13 mostra a simulação do manipulador proposto. Os pontos vermelhos são as juntas e ele encontra-se na posição inicial com todos os ângulos iguais a zero, para cada segmento do braço robótico foi adotado o tamanho de 15 cm.

Ambos os métodos apresentados culminam nos ângulos que devem ser aplicados ao braço manipulador robótico para que o ponto desejado seja alcançado. Esses ângulos são aplicados na simulação da Figura 13, e através de um código inspirado na matriz jacobiana, cria-se uma breve animação simulando a trajetória do ponto inicial até o ponto desejado. Em todas as figuras que representam o espaço analisado, as medidas dos eixos estão em centímetros.

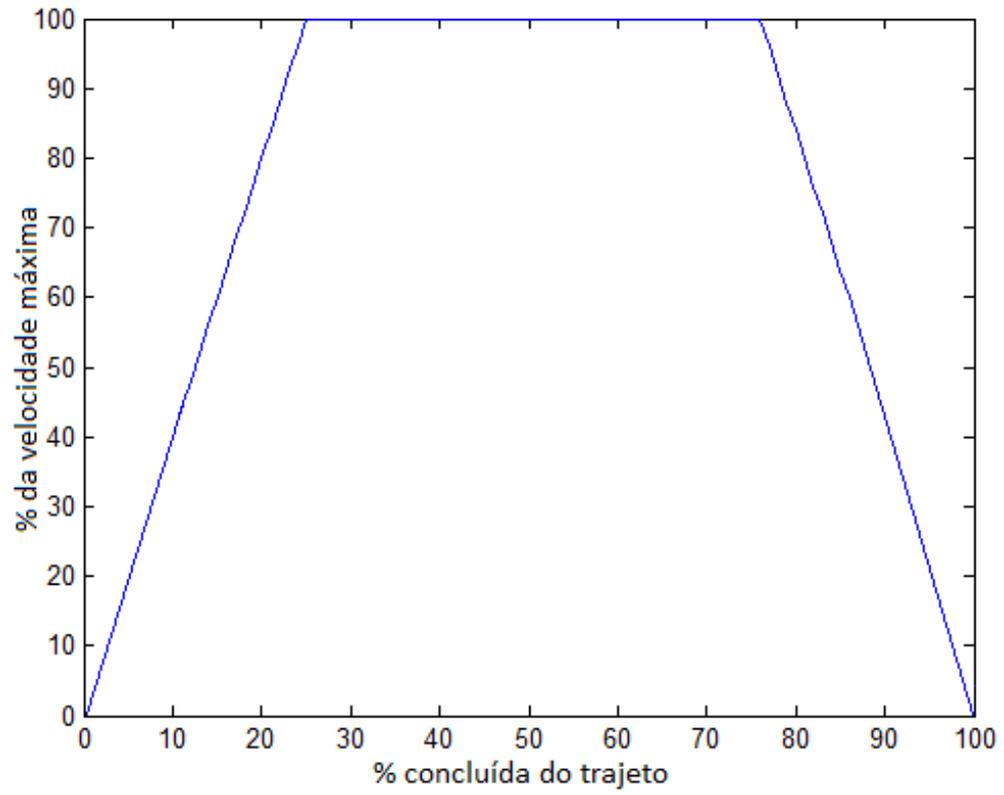
**Figura 13 - Simulação do braço manipulador robótico**



**Fonte: autoria própria.**

Também na animação, aplica-se o controle da velocidade do movimento, que foi feito através de uma relação entre a distância do efetor ao ponto final ou inicial, de forma que tanto no início, quanto no fim do movimento, as velocidades sejam reduzidas.

Em outras palavras, a velocidade deve aumentar gradualmente atingindo seu valor máximo em aproximadamente 25% do trajeto, mantendo essa velocidade até 75% do percurso, a partir daí ela é reduzida à medida que se aproxima do alvo. A Figura 14 mostra o gráfico da velocidade em função do trajeto.

**Figura 14 - Controle da velocidade**

Fonte: autoria própria.

## 5 RESULTADOS

Nessa seção são exibidos e discutidos os resultados das simulações propostas na seção anterior. Cada uma das estratégias adotadas é analisada, em seguida há uma comparação entre ambas e apresenta-se os resultados da cinemática direta (igual para os dois métodos).

- **Método 1**

Para a calibração, realizaram-se diversos testes afim de que, empiricamente, fosse decidido qual a melhor topologia para a resolução do problema, ou seja, efetua-se o treinamento da RNA variando o número de neurônios e a quantidade de camadas intermediárias, registrando as tentativas para que seja possível inferir qual a melhor escolha de acordo com o teste realizado. Na Tabela 1 tem-se os resultados obtidos.

**Tabela 1 - Topologias para calibração I**

| Quantidade de neurônios em cada camada | Épocas | Tempo (s) | Erro médio em x (cm) | Erro médio em y (cm) | Erro máximo em x (cm) | Erro máximo em y (cm) |
|--|--------|-----------|----------------------|----------------------|-----------------------|-----------------------|
| 5                                      | 1000   | 16        | 0,0653               | 0,0878               | 0,0767                | 0,0462                |
| 10                                     | 1000   | 12        | 0,0353               | 0,0283               | 0,083                 | 0,0593                |
| 20                                     | 583    | 14        | 0,0202               | 0,0243               | 0,0706                | 0,0762                |
| 35                                     | 502    | 16        | 0,0265               | 0,0335               | 0,0678                | 0,0708                |
| 50                                     | 234    | 11        | 0,0458               | 0,0346               | 0,0657                | 0,0706                |
| 75                                     | 337    | 25        | 0,0361               | 0,363                | 0,0809                | 0,07                  |
| 5 5                                    | 1000   | 23        | 0,0664               | 0,0587               | 0,0804                | 0,0623                |
| 10 10                                  | 1000   | 32        | 0,0441               | 0,0377               | 0,0784                | 0,0599                |
| 15 15                                  | 836    | 47        | 0,0286               | 0,0368               | 0,0731                | 0,073                 |

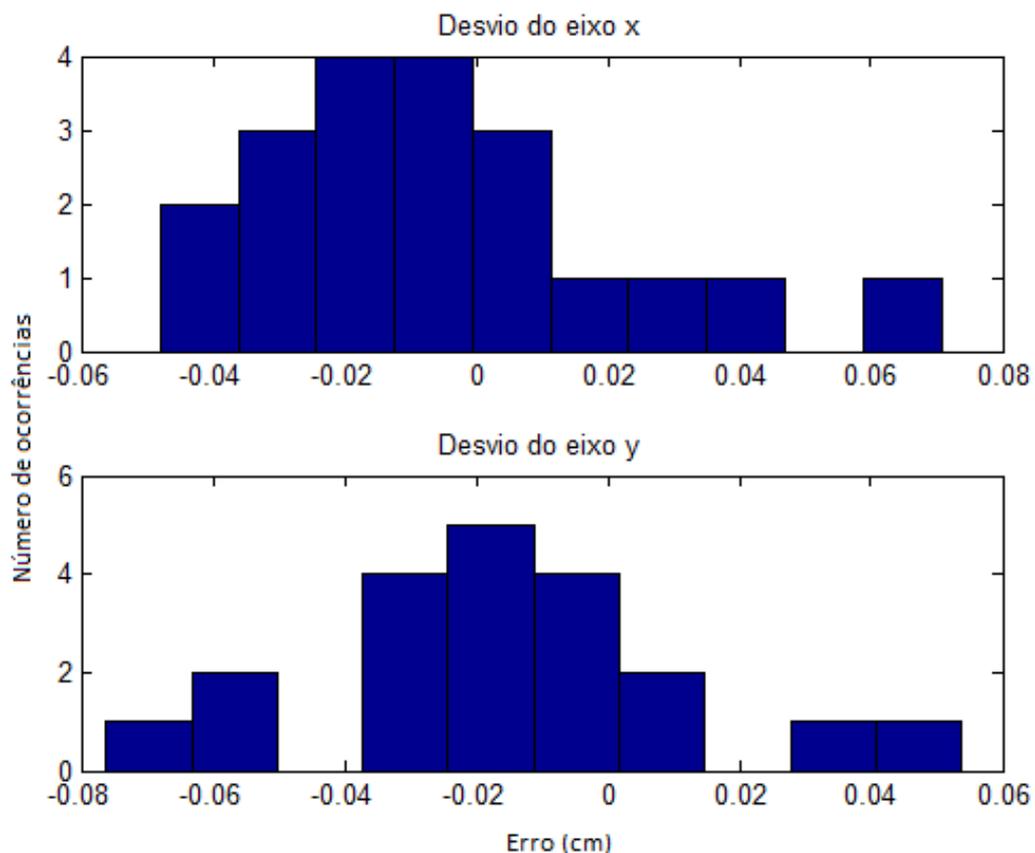
Fonte: autoria própria.

Com a análise dos resultados e com base nos valores obtidos, observa-se que a RNA foi capaz de calibrar a câmera. Sendo assim, foi escolhida a opção com 20 neurônios, que apresentou os menores erros médios, apesar de não ter os menores erros máximos. Essa escolha foi realizada devido aos erros médios terem maior impacto no resultado final. Embora o tempo de treinamento seja apresentado, ele não influi no tempo de resposta do sistema, pois este treinamento é realizado *offline*.

Considerando o método abordado, os erros do eixo z nesta etapa dependem exclusivamente da precisão do sensor utilizado, uma sugestão de sensor encontra-se no Anexo A.

Para confirmar a funcionalidade da RNA, a Figura 15 mostra os histogramas dos erros nos eixos x e y da topologia escolhida. Espera-se uma projeção que se aproxime de uma curva gaussiana, comprovando dessa forma que a metodologia utilizada na RNA foi correta.

**Figura 15 - Histograma dos erros I**

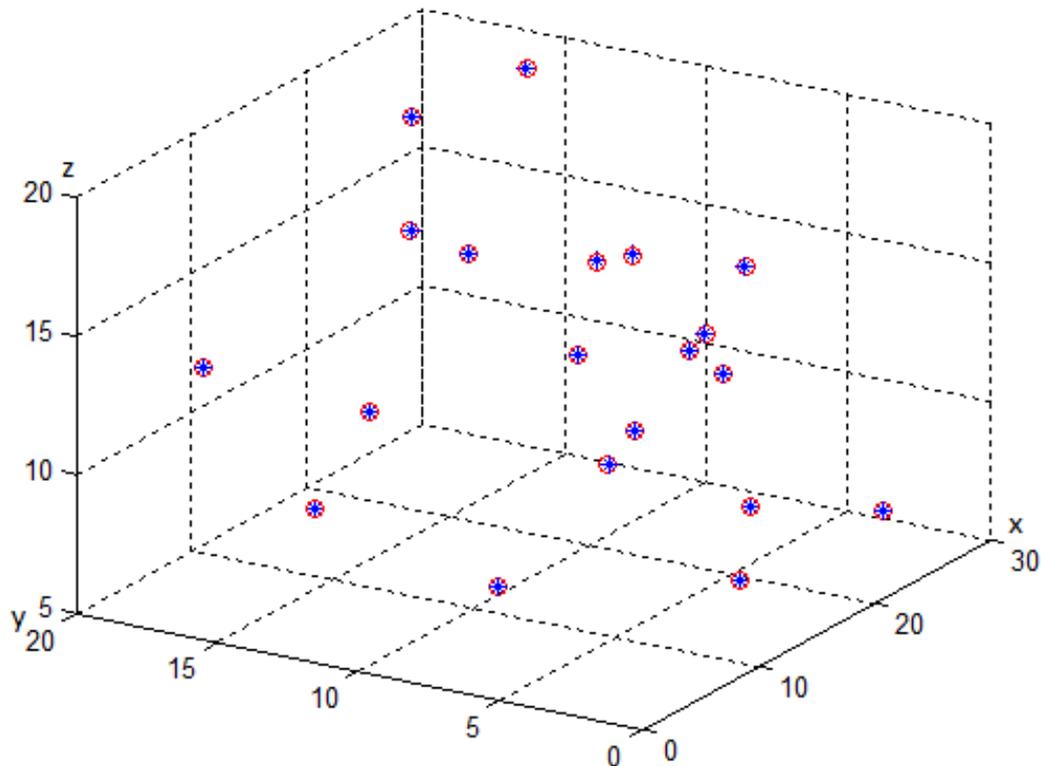


Fonte: autoria própria.

Como pode-se observar, os desvios possuem maior concentração nos valores próximos de 0, como o esperado de uma RNA com capacidade de generalização.

Ainda referente à calibração, a Figura 16 exibe a projeção dos pontos de teste (círculos vermelhos) juntamente com os valores fornecidos pela RNA (asteriscos azuis).

**Figura 16 - Teste da RNA I**



**Fonte: autoria própria.**

A precisão da RNA corresponde ao esperado, possuindo um baixo erro e sendo capaz de generalizar o espaço estudado. Como já mostrado na Tabela 1, os erros médios em x e y são, respectivamente, 0,0202 cm e 0,0243 cm.

Possuindo as coordenadas do ponto que deve ser alcançado, esses valores são inseridos em um algoritmo genético que define quais os ângulos necessários para que este ponto seja atingido. O algoritmo genético pode ter seu critério de parada ajustado, com isso é possível escolher qual o erro desejado para o procedimento, contudo, quanto menor o erro leva-se mais tempo para a conclusão do objetivo.

A Tabela 2 mostra o tempo decorrido para a finalização do algoritmo, levando em conta três pontos diferentes e alterando o erro desejado. O código foi rodado diversas vezes para cada situação e apresenta-se o resultado de um teste intermediário, não sendo o melhor nem o pior obtido.

**Tabela 2 - Tempo e erros do algoritmo genético**

| Pontos (x;y;z)<br>(cm) | Erro máximo permitido (cm) | Erro em x (cm) | Erro em y (cm) | Erro em z (cm) | Tempo (s) |
|------------------------|----------------------------|----------------|----------------|----------------|-----------|
| <b>(8;5;6)</b>         | 1                          | 0,4999         | 0,7198         | 0,8933         | 0,317     |
|                        | 0,3                        | 0,0804         | 0,1306         | 0,291          | 1,105     |
|                        | 0,1                        | 0,0675         | 0,0147         | 0,0735         | 25,53     |
| <b>(16;10;12)</b>      | 1                          | 0,7383         | 0,3421         | 0,5485         | 0,282     |
|                        | 0,3                        | 0,2184         | 0,1587         | 0,135          | 1,455     |
|                        | 0,1                        | 0,0398         | 0,0586         | 0,0694         | 15,087    |
| <b>(24;15;18)</b>      | 1                          | 0,8796         | 0,5576         | 0,4622         | 0,423     |
|                        | 0,3                        | 0,086          | 0,2833         | 0,1302         | 1,073     |
|                        | 0,1                        | 0,0621         | 0,0999         | 0,086          | 20,95     |

Fonte: autoria própria.

Nota-se que é factível conseguir um erro baixo e uma velocidade aceitável, da mesma forma que é concebível obter uma ação veloz com um erro admissível. Observa-se também que as velocidades seguem um padrão que depende do erro proposto. O erro total do processo pode ser, em seu caso máximo, a soma dos erros da calibração com os erros derivados do algoritmo genético, que variam de acordo com a intenção do operador.

- **Método 2**

A calibração da câmera, nesse caso, é realizada em conjunto com a resolução da cinemática inversa do braço manipulador robótico, por conseguinte, o erro apresentado é referente ao processo todo. Assim como no Método 1, realiza-se diversos teste afim de encontrar a topologia da RNA que melhor resolva o problema

analisado. A Tabela 3 apresenta as características da RNA, juntamente com os erros obtidos.

**Tabela 3 - Topologias para calibração II**

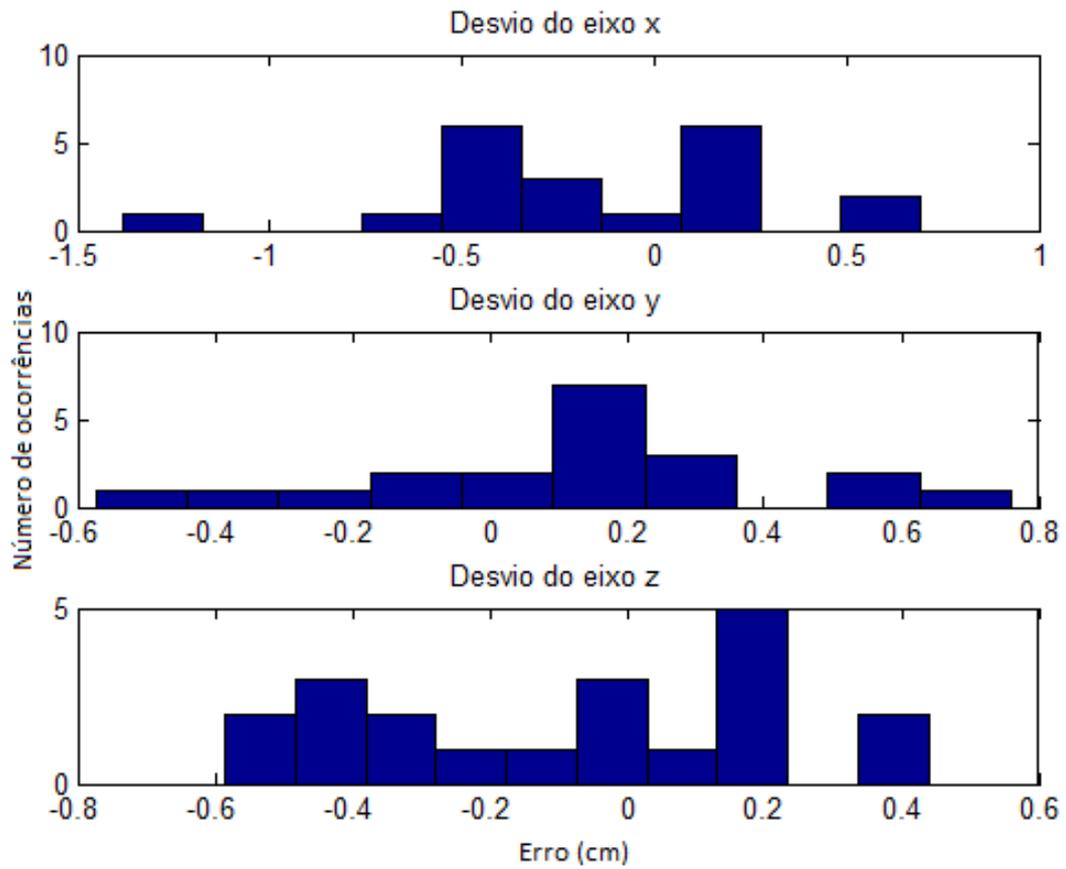
| Quantidade de neurônios em cada camada | Tempo (s) | Erro médio em x (cm) | Erro médio em y (cm) | Erro médio em z (cm) | Erro máximo em x (cm) | Erro máximo em y (cm) | Erro máximo em z (cm) |
|--|-----------|----------------------|----------------------|----------------------|-----------------------|-----------------------|-----------------------|
| 10                                     | 108       | 1,3176               | 0,9683               | 0,86                 | 2,427                 | 1,53                  | 1,11                  |
| 20                                     | 188       | 0,5646               | 0,4085               | 0,57                 | 1,658                 | 1,659                 | 1,32                  |
| 35                                     | 325       | 0,385                | 0,272                | 0,274                | 1,38                  | 0,759                 | 0,585                 |
| 50                                     | 523       | 0,5677               | 0,3288               | 0,45                 | 1,149                 | 1,196                 | 1,01                  |
| 75                                     | 892       | 0,6364               | 0,4102               | 0,57                 | 2,624                 | 1,532                 | 1,97                  |
| 5 5                                    | 117       | 1,3235               | 1,4921               | 0,79                 | 2,724                 | 3,636                 | 3,55                  |
| 10 10                                  | 237       | 0,7341               | 0,6014               | 0,71                 | 2,576                 | 2,86                  | 1,3                   |
| 20 20                                  | 716       | 0,5872               | 0,3963               | 0,52                 | 0,955                 | 1,189                 | 0,96                  |
| 30 30                                  | 1535      | 1,1446               | 1,8112               | 1,17                 | 2,038                 | 1,943                 | 2,13                  |

**Fonte: autoria própria.**

A RNA calibrou a câmera com eficácia, tendo como melhor opção a topologia com 35 neurônios, que apresentou os melhores resultados dentre os testes. Novamente o erro médio foi o fator determinante para a escolha dessa opção. Além do tempo apresentado, também decorreu-se 38109.073 segundos (aproximadamente 10,5 horas) referentes a aquisição dos dados para o treinamento, isso não influi no tempo de execução da ação, todavia expõe a grande demanda computacional do problema analisado.

A Figura 17 exibe os histogramas dos erros nos eixos x, y e z, para validação da metodologia utilizada, espera-se que as barras se aproximem da forma de uma gaussiana.

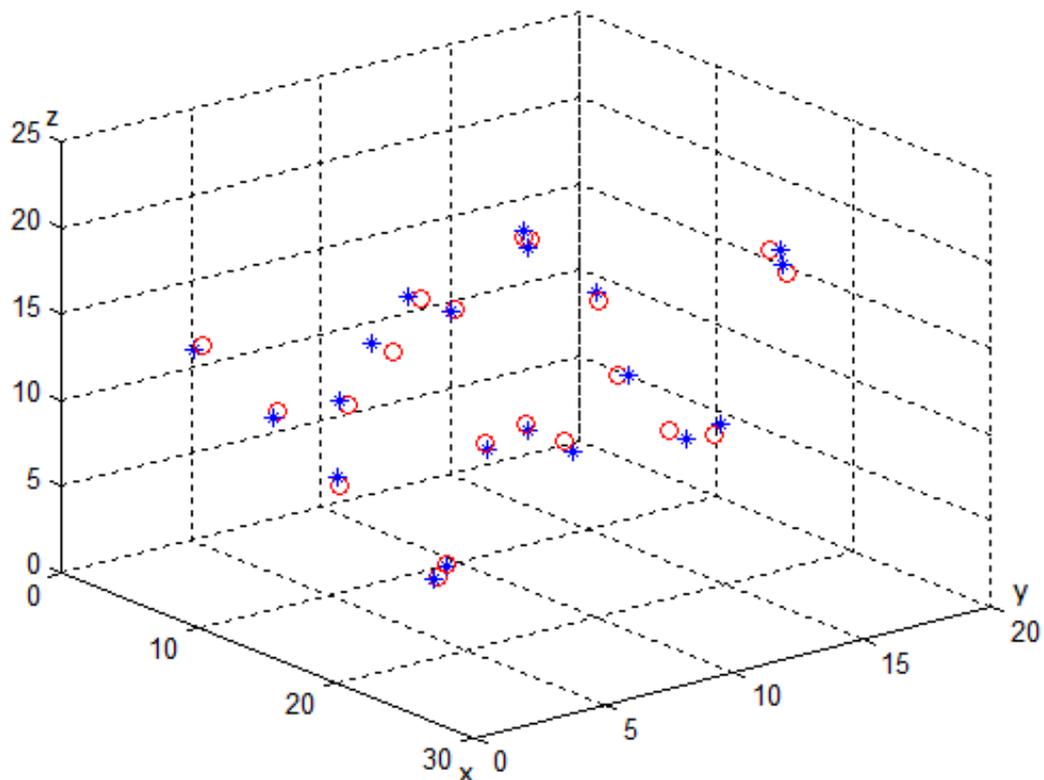
**Figura 17 - Histograma dos erros II**



Fonte: autoria própria.

O histograma corresponde ao esperado para a calibração. Demonstra-se na Figura 18 a projeção dos pontos de teste e dos valores fornecidos pela RNA, respectivamente representados pelos círculos vermelhos e asteriscos azuis.

**Figura 18 - Teste da RNA II**

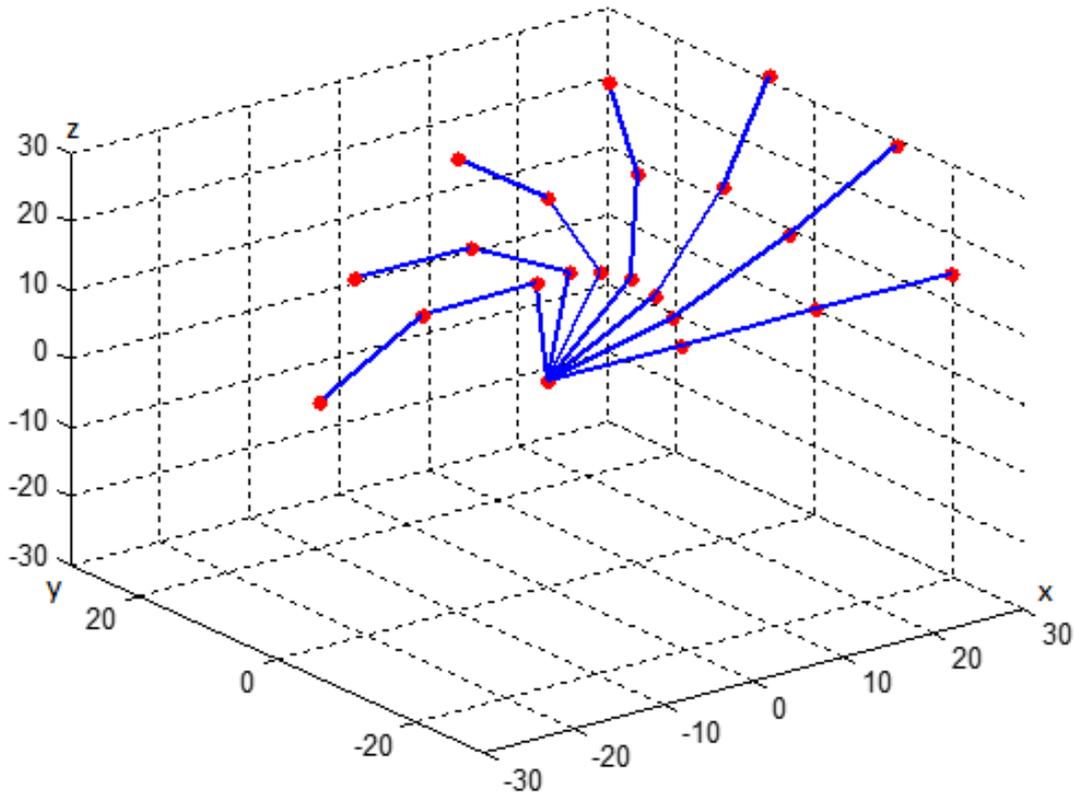


**Fonte: autoria própria.**

Novamente a RNA mostrou-se capaz de generalizar o espaço proposto, conseguindo se aproximar dos pontos desejados e apresentando um erro médio, já mostrado na Tabela 3, de 0,385 cm para o eixo x, 0,272 cm para o y e 0,274 cm para o z. Neste caso, a velocidade de resposta da rede (velocidade do processo *online*) foi 0,191 s.

A partir deste ponto, os métodos seguem pelo mesmo caminho, em ambos são fornecidos, como resultado, os ângulos necessários para que determinado ponto seja atingido. Tendo esta informação, aplica-se o controle de velocidade já mencionado e realiza-se a simulação do braço manipulador robótico, partindo de um ponto inicial até que o ponto pretendido seja alcançado. Uma gravação de uma das simulações realizadas encontra-se no *link*: <<https://www.youtube.com/watch?v=C9TJEq1a7gA>>, nela o braço manipulador parte da posição indicada na Figura 13 e atinge o ponto  $x = -30$  cm,  $y = -5$  cm e  $z = 10$  cm, uma representação da simulação encontra-se na Figura 19.

Figura 19 - Representação da simulação do braço manipulador robótico



Fonte: autoria própria.

A Tabela 4 compara o Método 1 e com o Método 2, analisando quesitos abordados durante o trabalho. Para o Método 1 foi considerado o caso mais preciso, porém, mais lento e, foram considerados os erros médios.

Tabela 4 - Método 1 x Método 2

| Método   | Tempo do processo <i>offline</i> (s) | Tempo do processo <i>online</i> (s) | Erro total em x (cm) | Erro total em y (cm) | Erro total em z (cm) |
|----------|--------------------------------------|-------------------------------------|----------------------|----------------------|----------------------|
| Método 1 | 583                                  | 20                                  | 0,06                 | 0,0829               | 0,0994               |
| Método 2 | 38434                                | 0,191                               | 0,385                | 0,272                | 0,274                |

Fonte: autoria própria.

Após a análise dos dados acima, percebe-se que realmente o Método 1 possui um foco na precisão em detrimento da velocidade, enquanto o Método 2 prioriza as variáveis de forma contrária, ainda assim, os dois métodos possuem bons resultados em ambos os itens.

Para que seja realizada a escolha de algum dos dois, é necessário que seja analisada a aplicação empregada e o modo de implementação, ou até mesmo sejam testadas ambas, pois para cada caso pode haver mudança de qual irá obter um melhor desempenho.

## 6 CONCLUSÃO

A RNA estudada generalizou o espaço analisado permitindo que fosse encontrada a relação entre as coordenadas reais e as coordenadas da imagem. A partir dos resultados obtidos na fase de teste e validação da RNA, pode-se afirmar que ela é capaz de solucionar o problema proposto de forma precisa e eficaz, visto que para problemas relacionados à localização de objetos por câmeras, o erro obtido se encontra dentro do aceitável.

Já o algoritmo genético, por sua vez, resolveu o problema da cinemática inversa com efetividade. Além disso as outras técnicas utilizadas responderam bem à simulação.

Todos os códigos foram feitos de modo que seja simples a modificação, não só dos tamanhos do braço, como também suas características, ou diversas outras características do sistema. Desta forma é possível realizar a aplicação dos métodos estudados tanto para um pequeno braço didático, quanto para um manipulador industrial.

Portanto, foram analisadas no trabalho duas possibilidades para a resolução do problema de braços manipuladores robóticos aliados à calibração de câmeras, cada um dos métodos apresenta vantagens e desvantagens, e os dois podem ser utilizados. Como futuros desafios tem-se a possibilidade da implementação de algum dos métodos estudados, além disso é viável a realização de melhorias em ambos os casos.

## REFERÊNCIAS

- AJWAD, S. A.; IQBAL, J.; KHAN, A. A.; MEHMOOD, A. **Disturbance-observer-based robust control of a serial-link robotic manipulator using SMC and PBC techniques**, *Studies in Informatics and Control*. Studies in Informatics and Control – SIC, vol. 24, pp. 401–408, 2015.
- CARDONA, M. **A new approach for the forward kinematics of general stewart-gough platforms**. 2015 IEEE Thirty Fifth Central American and Panama Convention (CONCAPAN XXXV), Tegucigalpa, pp. 1-6, 2015.
- CRAIG, J. J. **introduction to robotics mechanics and control**. 3ª edição, Prentice Hall, 1986.
- DA SILVA, I. N.; SPATTI, D. H.; FLAUZINO, R. A. **Redes neurais artificiais para engenharia e ciências aplicadas: curso prático**. 1 ed. São Paulo: ARTLIBER, 2010.
- DENAVIT, J. **Description and displacement analysis of mechanics based on the 2x2 dual matrices**. 1 ed. University of Northwestern, Evanston, 1955.
- GARCIA, E.; JIMENEZ, M. A.; DE SANTOS, P. G.; ARMADA, M. **The evolution of robotics research**. IEEE Robotics & Automation Magazine, vol. 14, no. 1, pp. 90-103, 2007.
- GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**. EUA: Addison-Wesley, 1989.
- GONZALEZ, R. C.; WOODS, R. E. **Processamento de Imagens Digitais**. Ed. Edgard Blücher, São Paulo, SP, 2000.
- GUEZ, A.; AHMAD, Z. **Solution to the inverse kinematics problem in robotics by neural networks**. IEEE 1988 International Conference on Neural Networks, San Diego, CA, USA, pp. 617-624 vol.2, 1988.
- HARRIS C.; STEPHENS M. **A combined corner and edge detector**. Proceedings of the 4th Alvey Vision Conference. pp. 147–151, 1988.
- HAYKIN, S. S. **Neural networks: a comprehensive foundation**. 2 ed. Pearson Prentice Hall, 823 p. 1999.
- HOLLERBACH, J.; SAHAR, G. **Wrist-partitioned inverse kinematic accelerations and manipulator dynamics**. 1984 IEEE International Conference on Robotics and Automation, pp. 152-161, 1984.
- JIN, L. G.; RUI, L. G. **Camera calibration for monocular vision system based on Harris corner extraction and neural network**, 2011 International Conference on, XianNing, pp. 1-4, 2011.
- JUSOH, N. F.; HARON, M. A.; SULAIMAN, F. **An FPGA implementation of shift converter block technique on FIFO for RS232 universal serial bus converter**

**control and system graduate research colloquium**, pp. 219-224, Seremban, 2012.

LENZ, R.; TSAI, R. **Techniques for calibration of the scale factor and image center for high accuracy 3D machine vision metrology**. Robotics and Automation. Proceedings. 1987 IEEE International Conference on, pp. 68-75, 1987.

LINDEN, R. **Algoritmos genéticos: uma importante ferramenta da inteligência computacional**. 2ª Ed. BR: Brasport, 2008.

LYNCH, M. B. H.; DAGLI, C. H.; VALLENKI, M. **The use of feedforward neural networks for machine vision calibration**. International Journal Production Economics, 60-61(1), pp. 479-489, 1999.

MASHHADANY Y. I. **ANFIS-Inverse-Controlled PUMA 560 Workspace Robot with Spherical Wrist**. International Symposium on Robotics and Intelligent Sensors, 2012.

MENDONÇA, M. **Redes neurais artificiais aplicadas à visão computacional**. Universidade Estadual Paulista, Bauru, SP, 2003.

MENDONÇA, M. **Uma contribuição ao desenvolvimento de sistemas inteligentes utilizando redes cognitivas dinâmicas**. 177 f. Tese (Doutorado em Engenharia de Automação e Sistemas) – Universidade Tecnológica Federal do Paraná: Curitiba, 2011.

NIKU, S. B. **Introdução à robótica: Análise, Controle, Aplicações**, 2 ed. LTC, VitalBook, 2013.

NUNES R. F. **Mapeamento da cinemática inversa de um manipulador robótico utilizando redes neurais artificiais configuradas em paralelo**. Dissertação de mestrado. Universidade Estadual de São Paulo– Ilha Solteira, 2016.

SÁNCHEZ, A.; TERÁN, A.; IBARRA, A.; ABATTA, L.; ALULEMA, D.; MOROCHO, D.; ENCALADA, F. **Design and construction of an anthropomorphic robotic arm of seven degrees of freedom with kinematic and dynamic analysis based on genetic algorithms**. Automatica (ICA-ACCA), IEEE International Conference, 2016.

SHHEIBIA, T. A. A. **Controle de um braço robótico utilizando uma abordagem de agente inteligente**. Dissertação de mestrado. UFPB – Universidade Federal da Paraíba, 2001.

YUAN, M.; HU, H.; JIANG Y.; HANG, S. **A new camera calibration based on neural network with tunable activation function in intelligent space**. Computational Intelligence and Design (ISCID), 2013 Sixth International Symposium on, Hangzhou, pp. 371-374, 2013.

**ANEXO A – *Datasheet* do sensor sugerido para obtenção das informações relacionadas ao eixo z**

**SHARP**

GP2Y0A21YK0F

# GP2Y0A21YK0F

**Distance Measuring Sensor Unit**  
**Measuring distance: 10 to 80 cm**  
**Analog output type**



## ■ Description

GP2Y0A21YK0F is a distance measuring sensor unit, composed of an integrated combination of PSD (position sensitive detector), IRED (infrared emitting diode) and signal processing circuit.

The variety of the reflectivity of the object, the environmental temperature and the operating duration are not influenced easily to the distance detection because of adopting the triangulation method.

This device outputs the voltage corresponding to the detection distance. So this sensor can also be used as a proximity sensor.

## ■ Features

1. Distance measuring range : 10 to 80 cm
2. Analog output type
3. Package size : 29.5×13×13.5 mm
4. Consumption current : Typ. 30 mA
5. Supply voltage : 4.5 to 5.5 V

## ■ Agency approvals/Compliance

1. Compliant with RoHS directive (2002/95/EC)

## ■ Applications

1. Touch-less switch  
(Sanitary equipment, Control of illumination, etc.)
2. Robot cleaner
3. Sensor for energy saving  
(ATM, Copier, Vending machine)
4. Amusement equipment  
(Robot, Arcade game machine)

**Notice** The content of data sheet is subject to change without prior notice.

In the absence of confirmation by device specification sheets, SHARP takes no responsibility for any defects that may occur in equipment using any SHARP devices shown in catalogs, data books, etc. Contact SHARP in order to obtain the latest device specification sheets before using any SHARP device.

Sheet No.: E4-A00201EN

Date Dec.01.2006

©SHARP Corporation

### ■ Absolute Maximum Ratings ( $T_a=25^{\circ}\text{C}, V_{CC}=5\text{V}$ )

| Parameter               | Symbol    | Rating               | Unit               |
|-------------------------|-----------|----------------------|--------------------|
| Supply voltage          | $V_{CC}$  | -0.3 to +7           | V                  |
| Output terminal voltage | $V_O$     | -0.3 to $V_{CC}+0.3$ | V                  |
| Operating temperature   | $T_{opr}$ | -10 to +60           | $^{\circ}\text{C}$ |
| Storage temperature     | $T_{stg}$ | -40 to +70           | $^{\circ}\text{C}$ |

### ■ Electro-optical Characteristics ( $T_a=25^{\circ}\text{C}, V_{CC}=5\text{V}$ )

| Parameter                   | Symbol       | Conditions   | MIN. | TYP. | MAX. | Unit |
|-----------------------------|--------------|--|------|------|------|------|
| Average supply current      | $I_{CC}$     | L=80cm (Note 1)  | —    | 30   | 40   | mA   |
| Distance measuring          | $\Delta L$   | (Note 1)   | 10   | —    | 80   | cm   |
| Output voltage              | $V_O$        | L=80cm (Note 1)  | 0.25 | 0.4  | 0.55 | V    |
| Output voltage differential | $\Delta V_O$ | Output voltage difference between L=10cm and L=80cm (Note 1) | 1.65 | 1.9  | 2.15 | V    |

\* L : Distance to reflective object

Note 1 : Using reflective object : White paper (Made by Kodak Co., Ltd. gray cards R-27·white face, reflectance; 90%)

### ■ Recommended operating conditions

| Parameter      | Symbol   | Rating     | Unit |
|----------------|----------|------------|------|
| Supply voltage | $V_{CC}$ | 4.5 to 5.5 | V    |

Note 1. The dimensions marked \* are described the dimensions of lens center position.

Note 2. Unspecified tolerances shall be  $\pm 0.3$  mm.

Note 3. The dimensions in parenthesis are shown for reference.