

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

MATEUS CABRAL DOS SANTOS

**MANIPULADOR ROBÓTICO AUTÔNOMO EMPREGANDO SISTEMAS
COMPUTACIONAIS INTELIGENTES**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO
2019

MATEUS CABRAL DOS SANTOS

**MANIPULADOR ROBÓTICO AUTÔNOMO EMPREGANDO SISTEMAS
COMPUTACIONAIS INTELIGENTES**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina Trabalho de Conclusão de Curso II, 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
2019



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



FOLHA DE APROVAÇÃO

Mateus Cabral dos Santos

Manipulador Robótico Autônomo Empregando Sistemas Computacionais Inteligentes

Trabalho de conclusão de curso apresentado às 14:00hs do dia 22/11/2019 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). Wagner Fontes Godoy - (Membro)

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

AGRADECIMENTOS

Agradeço a meus pais por proporcionar a oportunidade de cursar engenharia e realizar o sonho de construir robôs.

Agradeço ao meu orientador Prof. Dr. Márcio Mendonça, por me aceitar como estudante de iniciação científica, juntamente com meus colegas do Laboratório de Automação Avançada e Robótica pelo auxílio no desenvolvimento desse projeto, desde a concepção até a execução.

Aos meus amigos, Thalita, Rafael, Gabriella, Aron e Anderson, por estarem juntos comigo nessa jornada.

“Não entre em Pânico”
(ADAMS, Douglas; **Guia do mochileiro das Galáxias**, 1979.)

RESUMO

SANTOS, Mateus Cabral. **Manipulador Robótico Autônomo Empregando Sistemas Computacionais Inteligentes**. 2019. 62 f. Trabalho de Conclusão de Curso (Graduação) – Engenharia de Controle e Automação. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2019.

Neste trabalho é realizado o desenvolvimento de um protótipo de braço manipulador autônomo aplicando visão computacional e Redes Neurais Artificiais, noções de robótica como cinemática direta e inversa, planejamento de trajetória e sistemas microcontrolados. Os subsistemas desenvolvidos neste trabalho foram integrados. Assim, obteve-se um sistema capaz de localizar um objeto no espaço tridimensional e move-lo para um ponto pré-selecionado. Neste contexto, foram utilizadas Redes Neurais Artificiais para a solução da cinemática inversa e método de Denavit-Hartenberg para a solução da cinemática direta, bem como técnicas de processamento de imagem para identificação do objeto. O protótipo de manipulador robótico autônomo montado e os subsistemas desenvolvidos, apresentaram resultados satisfatórios de acordo com os parâmetros estabelecidos, validando a viabilidade da aplicação. Para futuros trabalhos é sugerido a implementação do sistema desenvolvido em um robô manipulador industrial.

Palavras-chave: Robótica. Redes Neurais Artificiais. Visão Computacional.

ABSTRACT

SANTOS, Mateus Cabral. **Autonomous Robotic Manipulator Applying Intelligent Computation system**. 2019. 62 f. Trabalho de Conclusão de Curso (Graduação) – Engenharia de Controle e Automação. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2019.

This paper proposes the development of an autonomous manipulator arm prototype applying computational vision and artificial neural networks, with notions of robotics such as direct and inverse kinematics, path planning and microcontroller systems. The subsystems developed in this paper were integrated. Thus, we obtained a system capable of locating an object in three-dimensional space and moving to a preselected point. In this context, artificial neural networks were used inverse kinematics solution and Denavit-Hartenberg method for the direct kinematics solution, as well as image processing techniques for object identification. The built prototype and the subsystems present satisfactory results according to the specified parameters, validating the application viability. For future work, the implementation of the system developed in an industrial manipulator robot is suggested.

Keywords: Robotics. Artificial Neural Networks. Computer Vision.

LISTA DE FIGURAS

Figura 1 – Coordenadas robóticas	13
Figura 2 - Etapas para processamento de imagens digitais.....	15
Figura 3 – Representação do modelo DH	17
Figura 4 – Neurônio <i>Perceptron</i>	20
Figura 5 – Rede <i>Perceptron</i> multicamadas	21
Figura 6 – Sentido dos sinais de propagação e retropropagação	22
Figura 7 – Planejamento de Trajetória – Polinômio de 5ª Ordem	23
Figura 8 - Ciclos PWM	25
Figura 9 - Fluxograma: visão macro do sistema.....	26
Figura 10 – Manipulador Robótico utilizado	27
Figura 11 – Etapas de Processamento abordadas.	28
Figura 12 – Esquema de Montagem do Sistema.	29
Figura 13 – Modelo do Suporte de Câmeras.	29
Figura 14 – Webcam Logitech C270	30
Figura 15 – Par de Imagem para Calibração	30
Figura 16 – Imagens de Entrada	31
Figura 17 – Imagens de Entrada em Preto e Branco	32
Figura 18 – Imagem distorcida	32
Figura 19 - Imagem com o objeto identificado.....	33
Figura 20 - Modelo D-H.....	34
Figura 22 - Resultado da Rede Neural Escolhida	36
Figura 23 – Comparativo entre Trajetórias Contínua e Discretas.....	38
Figura 24 – Arduino UNO	39
Figura 25 - Servo Motor JX PDI-6221MG	40
Figura 26 – Gráfico de movimento do atuador.	41
Figura 27 - Histograma de Erro de Calibração.	42
Figura 28 - Imagens Reconhecimento de Objeto.....	43
Figura 21 - Nuvem de pontos	45
Figura 29 - Posição Simulada do Manipulador.....	47
Figura 30 - Trajetória $T_f = 2s$, $T_d = 0,4s$	48
Figura 31 - Trajetória $T_f = 2$, $T_d = 0.1$	48
Figura 32 - Trajetória $T_f = 5$, $T_d = 0.5$	49
Figura 33 – Protótipo desenvolvido.	51
Figura 34 – Suporte de Câmeras.	51
Figura 35 – Área de trabalho do Manipulador robótico.	52
Figura 36 – Sistema em funcionamento.....	53
Figura 37 – Sugestão de posicionamento das câmeras.....	54
Figura 38 – Motor de Passo NEMA 17.....	55
Figura 39 – Manipulador Robótico com 6 GDL.	56

LISTA DE TABELAS

Tabela 1: Parâmetros do modelo D-H.....	34
Tabela 2: Primeira etapa de validação cruzada	35
Tabela 3: Validação cruzada da topologia selecionada	36
Tabela 4: Resultado de validação da RNA.....	37
Tabela 5: Coordenadas Calculadas x Coordenadas Reais.....	44
Tabela 6: Validação do modelo D-H.....	45
Tabela 7: Resultados do teste da RNA.	46
Tabela 8: Testes Atuadores Trajeto Início-Objeto.....	49
Tabela 9: Testes Atuadores Trajeto Objeto-Destino.	50
Tabela 10: Testes Atuadores Trajeto Destino-Início.	50

SUMÁRIO

1 INTRODUÇÃO	9
1.1 Motivação.....	10
1.2 Objetivos.....	10
1.2.1 Objetivo Geral	10
1.2.2 Objetivos Específicos	10
2 REVISÃO BIBLIOGRAFICA.....	12
2.1 Manipulador Robótico.....	12
2.2 Sistema de Visão Computacional	14
2.2.1 Processamento de imagens	15
2.2.2 Identificação de Objetos	16
2.3 Análise Cinemática de um Robô.....	16
2.3.1 Cinemática Direta e Inversa	16
2.3.2 Modelo Denavit-Hartenberg (D-H).....	17
2.3.3 Cinemática Inversa.....	18
2.4 Redes Neurais Artificiais	19
2.5 Planejamento de Trajetória.....	22
2.6 Acionamento dos atuadores	24
2.6.1 Modulação por Largura de Pulso (PWM)	24
3 MATERIAIS E METODOS	26
3.1 Manipulador Robótico.....	27
3.2 Sistema de Visão Computacional	28
3.2.1 Posicionamento das Câmeras.....	28
3.2.2 Aquisição de Imagens	30
3.3 Analise Cinemática	34
3.4 Cinemática inversa.....	34

1 INTRODUÇÃO

A robótica é uma das áreas de pesquisas que mais cresce, e envolve assunto multidisciplinares das engenharias mecânica, elétrica, eletrônica, ciência da computação, biologia e outras áreas (NIKU, 2011). Após os seus primeiros passos na década de 1960, a robótica chamou a atenção do setor industrial, com a promessa de aumentar a produtividade e diminuir os custos (ROSARIO, 2005).

Segundo o Instituto Americano de Robótica (RIA), robôs podem ser dispositivos que executam etapas sucessivas de uma tarefa de acordo com um determinado método, fáceis de modificar e dispositivos com meios de compreender seu ambiente e capazes de concluir uma tarefa com êxito, mesmo que haja mudanças no meio de atuação. Os robôs são elementos empregados na indústria atual, capazes de realizar diversas tarefas e operações com precisão e não requerem os elementos comuns de segurança e conforto que seres humanos demandam (NIKU, 2011). Entre os diversos robôs existentes, os manipuladores robóticos estão presentes em diversos tipos de indústria, executando funções como cortar, soldar, montar, picar e mover (AJWAD et al., 2015).

Para a montagem de um manipulador autônomo, inicialmente é necessário o reconhecimento de um objeto por meio de visão computacional para o reconhecimento de um objeto, localizando um alvo no espaço tridimensional. Posteriormente é necessário a solução da cinemática inversa, obtendo um vetor de ângulos para que o braço atinja uma determinada posição no espaço. A solução da cinemática inversa não é trivial, devido ao clássico problema das múltiplas soluções, onde conforme o aumento do número de graus de liberdade, mais soluções podem ser aplicadas, para um mesmo objetivo (NUNES, 2016).

A etapa final do desenvolvimento do sistema é o acionamento dos atuadores, conforme o planejamento de trajetória realizado anteriormente. Tal acionamento depende dos aspectos construtivos do manipulador, como tipo dos atuadores e o controlador utilizado.

O desenvolvimento de um manipulador autônomo exige que sejam solucionados alguns problemas clássicos da robótica como a cinemática inversa e o problema das múltiplas soluções. Também é necessário levar em conta fatores físicos

1 INTRODUÇÃO

A robótica é uma das áreas de pesquisas que mais cresce, e envolve assunto multidisciplinares das engenharias mecânica, elétrica, eletrônica, ciência da computação, biologia e outras áreas (NIKU, 2011). Após os seus primeiros passos na década de 1960, a robótica chamou a atenção do setor industrial, com a promessa de aumentar a produtividade e diminuir os custos (ROSARIO, 2005).

Segundo o Instituto Americano de Robótica (RIA), robôs podem ser dispositivos que executam etapas sucessivas de uma tarefa de acordo com um determinado método, fáceis de modificar e dispositivos com meios de compreender seu ambiente e capazes de concluir uma tarefa com êxito, mesmo que haja mudanças no meio de atuação. Os robôs são elementos empregados na indústria atual, capazes de realizar diversas tarefas e operações com precisão e não requerem os elementos comuns de segurança e conforto que seres humanos demandam (NIKU, 2011). Entre os diversos robôs existentes, os manipuladores robóticos estão presentes em diversos tipos de indústria, executando funções como cortar, soldar, montar, picar e mover (AJWAD et al., 2015).

Para a montagem de um manipulador autônomo, inicialmente é necessário o reconhecimento de um objeto por meio de visão computacional para o reconhecimento de um objeto, localizando um alvo no espaço tridimensional. Posteriormente é necessário a solução da cinemática inversa, obtendo um vetor de ângulos para que o braço atinja uma determinada posição no espaço. A solução da cinemática inversa não é trivial, devido ao clássico problema das múltiplas soluções, onde conforme o aumento do número de graus de liberdade, mais soluções podem ser aplicadas, para um mesmo objetivo (NUNES, 2016).

A etapa final do desenvolvimento do sistema é o acionamento dos atuadores, conforme o planejamento de trajetória realizado anteriormente. Tal acionamento depende dos aspectos construtivos do manipulador, como tipo dos atuadores e o controlador utilizado.

O desenvolvimento de um manipulador autônomo exige que sejam solucionados alguns problemas clássicos da robótica como a cinemática inversa e o problema das múltiplas soluções. Também é necessário levar em conta fatores físicos

3.5 Planejamento de trajetória.....	37
3.6 Acionamento dos Atuadores.....	38
3.6.1 Arduino UNO	39
3.6.2 Servo-motores.....	40
4 RESULTADOS.....	42
4.1 Visão Computacional	42
4.2 Modelo de Denavit-Hartenberg (D-H).....	45
4.3 Planejamento de Trajetória.....	47
4.4 Sistema Final	51
5 Discussão de Resultados	54
5.1 Sistema de Visão Computacional	54
5.2 Sistema de Solução da Cinemática Inversa.....	55
5.3 Manipulador Robótico.....	55
6 CONCLUSÃO	57
REFERÊNCIAS.....	58

1 INTRODUÇÃO

A robótica é uma das áreas de pesquisas que mais cresce, e envolve assunto multidisciplinares das engenharias mecânica, elétrica, eletrônica, ciência da computação, biologia e outras áreas (NIKU, 2011). Após os seus primeiros passos na década de 1960, a robótica chamou a atenção do setor industrial, com a promessa de aumentar a produtividade e diminuir os custos (ROSARIO, 2005).

Segundo o Instituto Americano de Robótica (RIA), robôs podem ser dispositivos que executam etapas sucessivas de uma tarefa de acordo com um determinado método, fáceis de modificar e dispositivos com meios de compreender seu ambiente e capazes de concluir uma tarefa com êxito, mesmo que haja mudanças no meio de atuação. Os robôs são elementos empregados na indústria atual, capazes de realizar diversas tarefas e operações com precisão e não requerem os elementos comuns de segurança e conforto que seres humanos demandam (NIKU, 2011). Entre os diversos robôs existentes, os manipuladores robóticos estão presentes em diversos tipos de indústria, executando funções como cortar, soldar, montar, picar e mover (AJWAD et al., 2015).

Para a montagem de um manipulador autônomo, inicialmente é necessário o reconhecimento de um objeto por meio de visão computacional para o reconhecimento de um objeto, localizando um alvo no espaço tridimensional. Posteriormente é necessário a solução da cinemática inversa, obtendo um vetor de ângulos para que o braço atinja uma determinada posição no espaço. A solução da cinemática inversa não é trivial, devido ao clássico problema das múltiplas soluções, onde conforme o aumento do número de graus de liberdade, mais soluções podem ser aplicadas, para um mesmo objetivo (NUNES, 2016).

A etapa final do desenvolvimento do sistema é o acionamento dos atuadores, conforme o planejamento de trajetória realizado anteriormente. Tal acionamento depende dos aspectos construtivos do manipulador, como tipo dos atuadores e o controlador utilizado.

O desenvolvimento de um manipulador autônomo exige que sejam solucionados alguns problemas clássicos da robótica como a cinemática inversa e o problema das múltiplas soluções. Também é necessário levar em conta fatores físicos

que afetam o funcionamento do robô, como aquecimento dos motores e instabilidade no movimento.

Os problemas da cinemática inversa e múltiplas soluções podem ser resolvidos através da aplicação de um sistema inteligente com Redes Neurais Artificiais (SOUZA et al.,2018). Para os problemas físicos, o planejamento da trajetória deve ser realizado para que não haja um desgaste da estrutura e dos atuadores.

1.1 Motivação

O desenvolvimento de um manipulador robótico capaz de identificar um ou mais pontos no espaço permite a detecção de superfícies no espaço e consequentemente a possível seleção de amostras de produtos defeituosos ou abaixo dos padrões de qualidade necessários para um processo ou produto final. Sendo assim, um manipulador autônomo poderia reconhecer uma peça que apresenta risco para a qualidade do processo. Após a etapa de reconhecimento, o manipulador poderia remover da esteira de transporte um objeto, auxiliando no desempenho e qualidade de uma linha de produção.

1.2 Objetivos

Nesta seção do trabalho serão apresentados os objetivos gerais e específicos do projeto proposto.

1.2.1 Objetivo Geral

O objetivo do trabalho desenvolvido foi realizar a montagem de um protótipo de manipulador robótico autômato capaz de reconhecer um ponto no espaço e movê-lo para um espaço pré-determinado, utilizando sistemas computacionais inteligentes.

1.2.2 Objetivos Específicos

- Determinar a posição de um objeto no espaço tridimensional utilizando visão computacional.

1 INTRODUÇÃO

A robótica é uma das áreas de pesquisas que mais cresce, e envolve assunto multidisciplinares das engenharias mecânica, elétrica, eletrônica, ciência da computação, biologia e outras áreas (NIKU, 2011). Após os seus primeiros passos na década de 1960, a robótica chamou a atenção do setor industrial, com a promessa de aumentar a produtividade e diminuir os custos (ROSARIO, 2005).

Segundo o Instituto Americano de Robótica (RIA), robôs podem ser dispositivos que executam etapas sucessivas de uma tarefa de acordo com um determinado método, fáceis de modificar e dispositivos com meios de compreender seu ambiente e capazes de concluir uma tarefa com êxito, mesmo que haja mudanças no meio de atuação. Os robôs são elementos empregados na indústria atual, capazes de realizar diversas tarefas e operações com precisão e não requerem os elementos comuns de segurança e conforto que seres humanos demandam (NIKU, 2011). Entre os diversos robôs existentes, os manipuladores robóticos estão presentes em diversos tipos de indústria, executando funções como cortar, soldar, montar, picar e mover (AJWAD et al., 2015).

Para a montagem de um manipulador autônomo, inicialmente é necessário o reconhecimento de um objeto por meio de visão computacional para o reconhecimento de um objeto, localizando um alvo no espaço tridimensional. Posteriormente é necessário a solução da cinemática inversa, obtendo um vetor de ângulos para que o braço atinja uma determinada posição no espaço. A solução da cinemática inversa não é trivial, devido ao clássico problema das múltiplas soluções, onde conforme o aumento do número de graus de liberdade, mais soluções podem ser aplicadas, para um mesmo objetivo (NUNES, 2016).

A etapa final do desenvolvimento do sistema é o acionamento dos atuadores, conforme o planejamento de trajetória realizado anteriormente. Tal acionamento depende dos aspectos construtivos do manipulador, como tipo dos atuadores e o controlador utilizado.

O desenvolvimento de um manipulador autônomo exige que sejam solucionados alguns problemas clássicos da robótica como a cinemática inversa e o problema das múltiplas soluções. Também é necessário levar em conta fatores físicos

que afetam o funcionamento do robô, como aquecimento dos motores e instabilidade no movimento.

Os problemas da cinemática inversa e múltiplas soluções podem ser resolvidos através da aplicação de um sistema inteligente com Redes Neurais Artificiais (SOUZA et al.,2018). Para os problemas físicos, o planejamento da trajetória deve ser realizado para que não haja um desgaste da estrutura e dos atuadores.

1.1 Motivação

O desenvolvimento de um manipulador robótico capaz de identificar um ou mais pontos no espaço permite a detecção de superfícies no espaço e consequentemente a possível seleção de amostras de produtos defeituosos ou abaixo dos padrões de qualidade necessários para um processo ou produto final. Sendo assim, um manipulador autônomo poderia reconhecer uma peça que apresenta risco para a qualidade do processo. Após a etapa de reconhecimento, o manipulador poderia remover da esteira de transporte um objeto, auxiliando no desempenho e qualidade de uma linha de produção.

1.2 Objetivos

Nesta seção do trabalho serão apresentados os objetivos gerais e específicos do projeto proposto.

1.2.1 Objetivo Geral

O objetivo do trabalho desenvolvido foi realizar a montagem de um protótipo de manipulador robótico autômato capaz de reconhecer um ponto no espaço e movê-lo para um espaço pré-determinado, utilizando sistemas computacionais inteligentes.

1.2.2 Objetivos Específicos

- Determinar a posição de um objeto no espaço tridimensional utilizando visão computacional.

- Solucionar o problema da cinemática inversa utilizando sistemas inteligentes.
- Planejar a trajetória do manipulador robótico utilizando polinômios para posição, velocidade e aceleração.
- Montar um controle para o manipulador robótico que seja capaz de realizar o movimento planejado.

2 REVISÃO BIBLIOGRAFICA

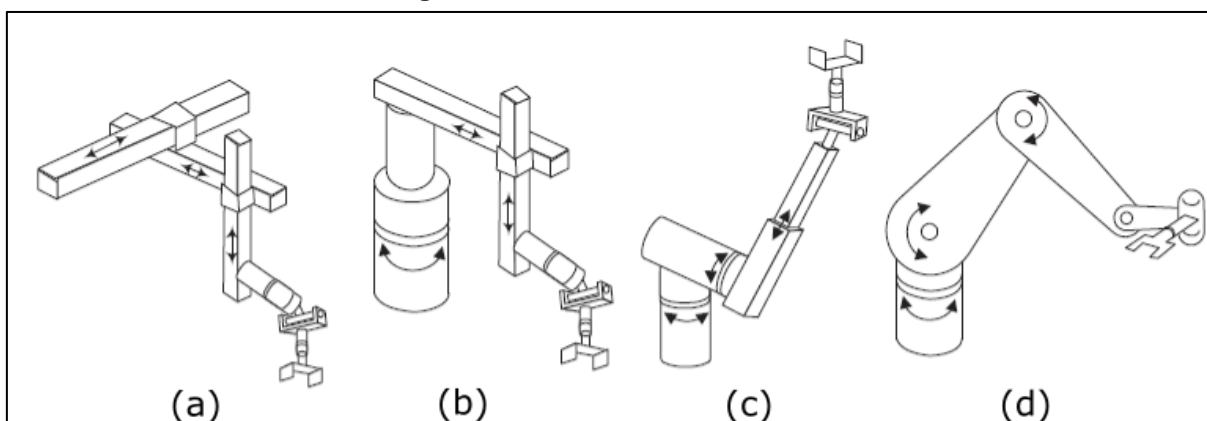
Nessa seção, são apresentados aspectos teóricos e trabalhos correlatos desenvolvidos para validar a relevância do tema abordado. Os temas abordados envolvem assuntos como manipuladores robóticos, visão computacional, estudo de cinemática de manipuladores, Redes Neurais Artificiais, estudos de trajetórias dos manipuladores robóticos.

2.1 Manipulador Robótico

Um manipulador pode ser considerado também como a integração de elementos distintos que juntos formam um sistema capaz de realizar as funções desejadas. Dentre os elementos que compõem um robô, estão: Manipulador, Atuadores, Sensores, Controlador, Processador e Software (NIKU, 2011).

Um manipulador robótico é composto por ligações e juntas. As ligações são componentes físicas dos robôs. As juntas são elementos que conectam os membros do robô e impõem restrições aos movimentos entre duas ligações. As juntas restringem movimentos das ligações e são medidos em Graus de Liberdade (GDL). Por padrão, uma ligação possui 6 GDL, translação e rotação ao longo das três coordenadas cartesianas. Robôs são frequentemente classificados pelo número de Graus de Liberdade (HEXMOOR, 2003).

Os manipuladores são divididos em anatomias, de acordo com suas coordenadas robóticas, definidas pelos diferentes tipos de juntas aplicadas. As configurações mais comuns são: Cartesiana (a), Cilíndrica (b), Esférica (c), Articulada (d) (NIKU, 2011). A Figura 1 exemplifica as configurações mais comuns.

Figura 1 – Coordenadas robóticas

Fonte: NIKU (2011).

A junta de um robô pode ser de diferentes tipos, como lineares, rotativas, deslizantes ou esféricas, sendo a maioria das aplicações feitas com juntas lineares ou rotativas. No caso das juntas lineares, não há rotação envolvida, sendo comum a utilização de cilindros hidráulicos, pneumáticos ou atuadores elétricos lineares. Embora juntas rotativas possam ser hidráulicas e pneumáticas, é mais comum a utilização de juntas acionadas eletricamente, como motores de passo ou servomotores (NIKU, 2011).

Para localizar um ponto no espaço é necessário especificar três coordenadas (como as coordenadas cartesianas X, Y, Z). Para objetos bidimensionais ou tridimensionais, mesmo com as três coordenadas (em 3-D) serem suficiente para indicar a posição do ponto, não são capazes de indicar a orientação do mesmo. Para isso são necessárias mais informações dos pontos de referência. Esse conceito de posição e orientação, os ângulos das juntas, são capazes de especificar a localização das juntas ou ligações do manipulador.

Outros conceitos ou parâmetros de desenvolvimento a serem considerados na construção de um manipulador robótico são a repetibilidade e a precisão. Nas quais, a precisão é definida como a acurácia com que o manipulador alcança determinado ponto no espaço, e a repetibilidade referente a precisão do movimento após muitas repetições (NIKU, 2011).

2.2 Sistema de Visão Computacional

A pesquisa no campo da visão tem sido foco de pesquisadores desde o início da computação e se estende até os dias atuais, compreendendo aplicações nas áreas de automação industrial, robótica e processamento de documentos (MOLZ, 2001).

A visão computacional representa os meios de transformação de um sinal de imagem para um meio digital, para que esse possa ser processado com o uso de computador (ZHAO *et al.*, 2005). De tal modo, a ciência da visão computacional estuda técnicas de processamento de imagens para extrair e processar características de objetos e ambientes (GONZALEZ e WOODS, 2000).

O intuito da aplicação de um sistema de visão computacional em um manipulador autônomo é permitir que o sistema seja capaz de identificar o ambiente de trabalho, tomando decisões autônomas. Através de um conjunto de câmeras, o robô é capaz de identificar um alvo ou obstáculos na região de trabalho do manipulador robótico e enviar as coordenadas globais do ponto para que seja realizado o planejamento de trajetória após a etapa de solução da cinemática inversa.

Alguns trabalhos influenciaram as decisões tomadas no desenvolvimento do trabalho apresentado. Jong-Kyu Oh e Chan-Ho Lee (2007) desenvolveram um sistema de visão computacional estéreo de propósito generalizado aplicado em robôs industriais. O sistema utiliza pontos específicos do objeto de trabalho para identificar a posição no espaço.

Xinjian Fan et al. (2014) apresenta uma técnica de visão computacional em robôs industriais com a finalidade de pegar objetos de trabalho em posições randômicas. No trabalho desenvolvido é aplicado uma combinação de visão computacional 2D e 3D. Na etapa 2D o objeto de trabalho é detectado e utilizando a visão 3D, as coordenadas do objeto são identificadas.

Pedra et al. (2013) realiza a calibração de um conjunto de câmeras estéreo. No trabalho apresentado foi desenvolvida uma rede neural artificial que utiliza como valores de entrada o par de coordenadas (u , v , w) das câmeras e obtém como saída as coordenadas globais do sistema (X , Y , Z).

Lestriandoko e Sadikin (2016), utilizam técnicas de processamento para identificar círculos em imagens. O processo envolve a detecção de bordas utilizando

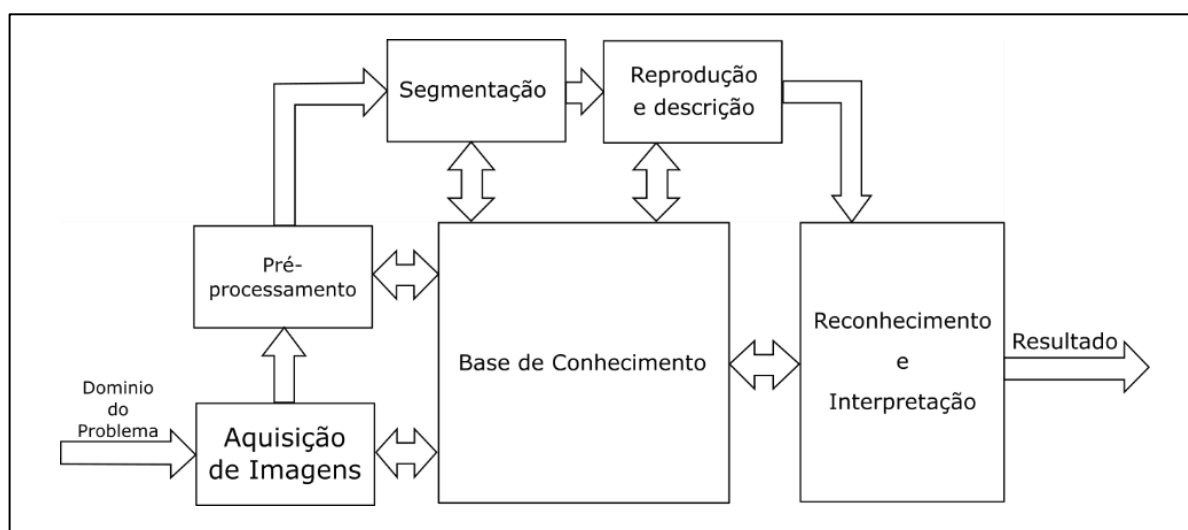
o filtro de Sobel, Transformada de Hough para a detecção de formas circulares e o Filtro Chapéu Mexicano para realce das bordas.

2.2.1 Processamento de imagens

A identificação de um objeto e extração de suas coordenadas exige uma série de etapas de processamento que devem ser realizadas. O processamento de imagens digitais abrange uma escala de hardware, software e fundamentos teóricos (GONZALEZ e WOODS, 2000).

Segundo Gonzalez e Woods, o processamento de imagens seguem as etapas exibidas na Figura 2.

Figura 2 - Etapas para processamento de imagens digitais



Fonte: Adaptado de GONZALEZ e WOODS (2000).

Algumas das etapas como aquisição de imagens, necessitam de equipamentos físicos. Outras são puramente software, como pré-processamento e segmentação. Por fim, algumas etapas necessitam de um embasamento teórico acerca do problema a ser solucionado, como Base de Conhecimento, Reconhecimento e Interpretação das imagens.

2.2.2 Identificação de Objetos

Como abordado por Lestriandoko e Sadikin (2016), uma das maneiras de identificar círculos em imagens é através da Transformada Circular de Hough, que considera três parâmetros que descrevem um círculo matematicamente, as coordenadas do centro e Raio da circunferência. Com esses parâmetros é possível descrever um círculo através da Equação 1.

$$r^2 = (X - a)^2 + (Y - b)^2 \quad (1)$$

O algoritmo da Transformada Circular de Hough consiste em selecionar pixels que correspondem a borda de objetos na imagem. Utilizando as coordenadas dos pixels escolhidos como centro (a, b) , o algoritmo gera círculo com raio fixo r . O local no qual houver a maior concentração de bordas de círculos gerados é considerado como centro do círculo da imagem.

2.3 Análise Cinemática de um Robô

A fim de executar uma tarefa, o manipulador robótico precisa se deslocar da sua posição inicial de descanso, até a área de atuação de forma controlada. Esse deslocamento entre pontos é definido como trajetória. A análise cinemática de um robô é realizada a partir de um modelo geométrico ou de um protótipo que representam a evolução temporal da posição, velocidade e aceleração de cada junta do robô (NIKU, 2011).

Um modelo cinemático ou dinâmico é capaz de representar o posicionamento do robô no espaço tridimensional. Entre as formas de representação, os modelos de cinemática direta e inversa se destacam.

2.3.1 Cinemática Direta e Inversa

Através do modelo cinemático direto é possível obter a posição e orientação final do efector do robô, sabendo os ângulos de controle de cada junta. Em contrapartida, o modelo cinemático inverso retorna os ângulos de cada junta para que o robô seja posicionado em determinada posição. Conforme o número de Graus de

liberdade de um robô aumenta, também crescem o número de possíveis soluções para a cinemática direta e inversa. Esse problema clássico da robótica é descrito como problema das múltiplas soluções (HEXMOOR, 2013).

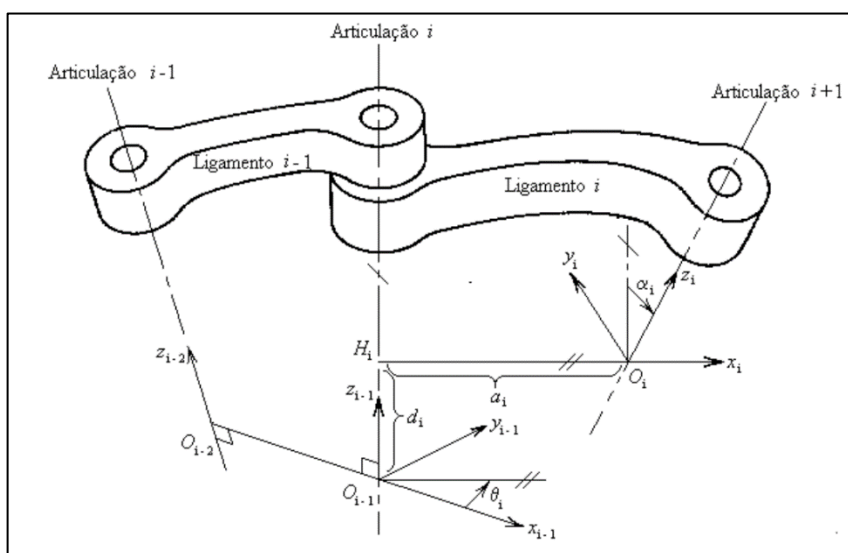
A partir das informações da cinemática direta e inversa é possível planejar a trajetória do robô manipulador. Neste contexto, a forma com que um robô é movido de um ponto inicial a um final, de modo controlado (NIKU, 2011).

Dentre as diversas possibilidades de descrever um modelo cinemático de um robô estão o Modelo de Denavit-Hartenberg (D-H) para a Cinemática Direta e aplicação de sistemas inteligentes para a solução da Cinemática Inversa.

2.3.2 Modelo Denavit-Hartenberg (D-H)

O modelo de Denavit-Hartenberg é uma ferramenta utilizada para solucionar a cinemática direta de sistemas articulados com n graus de liberdade (DENAVIT, 1955). Baseado em quatro parâmetros (d , a , α , Θ), a notação de Denavit-Hartenberg descreve a posição do efector em relação as características do próprio manipulador. A Figura 3 exibe um braço manipulador de 2-GDL, nos parâmetros do modelo D-H.

Figura 3 – Representação do modelo DH



Fonte: NIKU (2011).

A partir da análise da Figura 3, tem-se que (NIKU, 2011):

- a_i refere-se à distancia entre as juntas no eixo X ;
- α_i refere-se ao ângulo entre as juntas com referência ao eixo X ;
- d_i refere-se a distância entre os ligamentos, com referência ao eixo Z ;
- θ_i refere-se ao ângulo entre os ligamentos, com referência ao eixo Z ;

O movimento que pode ser realizado por um dos ligamentos de um robô é descrito pela multiplicação de matrizes de rotação e translação para cada ligamento. Essa multiplicação gera a matriz (3), que após ser multiplicada pela matriz que descreve o movimento das outras juntas obtém-se a matriz de transformação do manipulador (4), que descreve a posição da extremidade do manipulador no plano cartesiano.

$$A_{i-1}^i = Rot(z, \theta_{n+1}) \times Trans(0,0, d_{n+1}) \times Trans(a_{n+1}, 0,0) \times Rot(x, \alpha_{n+1}) \quad (2)$$

$$A_{i-1}^i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cdot \cos\alpha_i & \sin\theta_i \cdot \sin\alpha_i & \alpha_i \cdot \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cdot \cos\alpha_i & -\cos\theta_i \cdot \sin\alpha_i & \alpha_i \cdot \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$A_0^n = A_0^1 \cdot A_1^2 \cdot A_{n-1}^n \quad (4)$$

2.3.3 Cinemática Inversa

Com o objetivo de o manipulador alcançar um determinado ponto no espaço, será necessária solução da cinemática inversa. Esse cálculo é um problema constante no estudo da robótica e pode ser realizado de diversas formas como solução numérica, algoritmos genéticos, ANFIS e Redes Neurais Artificiais (SOUZA et al., 2018).

A solução numérica, se torna inviável para sistemas com mais de 3-GDL, apresentando múltiplas soluções. O uso de sistemas inteligentes na solução da cinemática inversa oferece uma alternativa para o problema das múltiplas soluções.

Para a solução do problema da cinemática inversa, algumas considerações devem ser aplicadas, como por exemplo, velocidades e acelerações iniciais e finais e a posição de descanso do braço manipulador é fixa, ou seja, inicia a trajetória sempre do mesmo ponto inicial. Desse modo, é preciso reconhecer a posição do alvo no espaço para determinar apenas a posição final da trajetória. Neste trabalho, optou-se pela solução do problema da calibração e da solução da cinemática inversa por meio da aplicação de Redes Neurais Artificiais.

Souza et al. (2018), em seu trabalho, fazem uma comparação entre três métodos de obtenção da solução da cinemática inversa e um estudo sobre o planejamento de trajetória de um manipulador.

Nunes (2016) utiliza uma Rede Neural Artificial Multicamada para cálculo da cinemática inversa de um manipulador robótico com 5 Graus de Liberdade. Também é realizado um planejamento de trajetória com base na posição final do efector.

2.4 Redes Neurais Artificiais

As Redes Neurais Artificiais são modelos computacionais inspirados no sistema nervoso de seres vivos. Possuem capacidade de aquisição e manutenção de conhecimento e podem ser definidas como conjuntos de unidades de processamento, caracterizadas por neurônios artificiais, interligados por muitas interconexões ou sinapses artificiais, sendo essas representadas por vetores e matrizes de pesos sinápticos (SILVA, SPATTI E FLAUZINO, 2010).

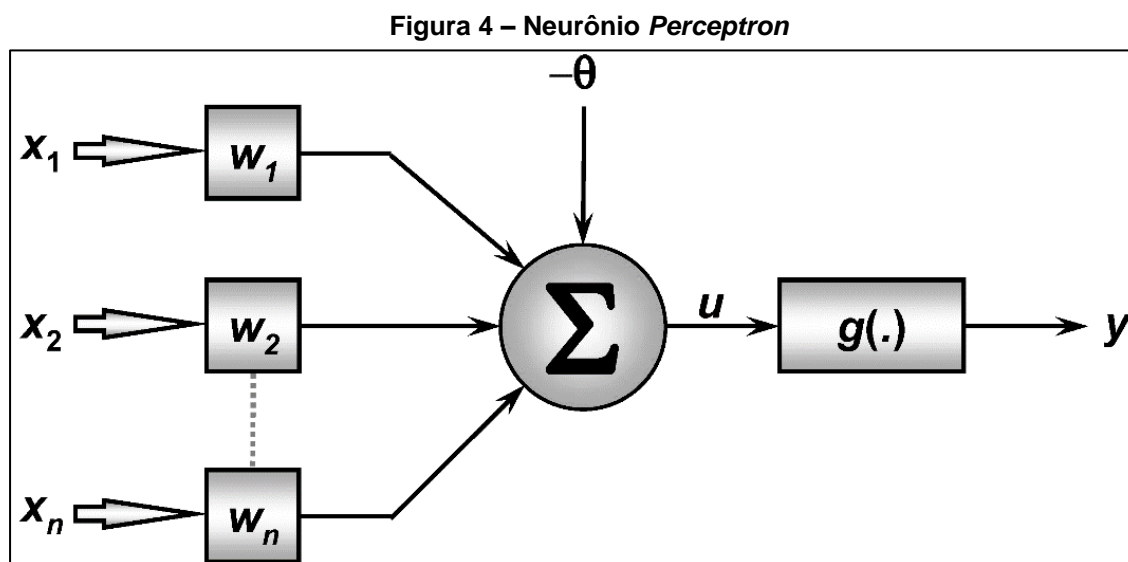
O uso de Redes Neurais Artificiais decorre das características que tornam a rede ideal para uso em sistemas inteligentes. Dentre as principais características encontram-se:

- a) Adaptação por Experiência: as adaptações dos parâmetros internos da rede, em geral dos pesos sinápticos, são alteradas a partir da apresentação de exemplos relacionados ao comportamento do processo. Isso possibilita que a rede adquira conhecimento por experiência.
- b) Capacidade de aprendizado: através de algoritmos de treinamento, a rede consegue extrair as informações necessárias que compõe a aplicação.

- c) Habilidade de generalização: A partir de um treinamento, a rede é capaz de generalizar o conhecimento adquirido, podendo estimar soluções até então desconhecidas.
- d) Tolerância a Falhas: o elevado número de interconexões entre os neurônios artificiais tolerante a falhas, deixando o sistema operacional mesmo com parte da rede corrompida.
- e) Facilidade de prototipagem: a implementação de um protótipo pode ser feita com facilidade, uma vez que, após o treinamento da rede, os resultados são obtidos matematicamente.

Devido as capacidades adaptativas das Redes Neurais Artificiais, elas podem ser utilizadas em diversos problemas de engenharia como aproximar funções, controle de processos, reconhecimento de padrões, sistemas de previsão entre outros.

Uma das configurações mais utilizadas é *Perceptron*, idealizado em 1958 por Rosenblatt. A rede *Perceptron* pertence à arquitetura *Feedforward*, onde o fluxo de informações na estrutura da rede é direcionado do sentido da camada de entrada para a camada de saída, sem realimentação de valores obtidos pelo único neurônio da rede. A Figura 4 apresenta a estrutura de um neurônio *Perceptron*.



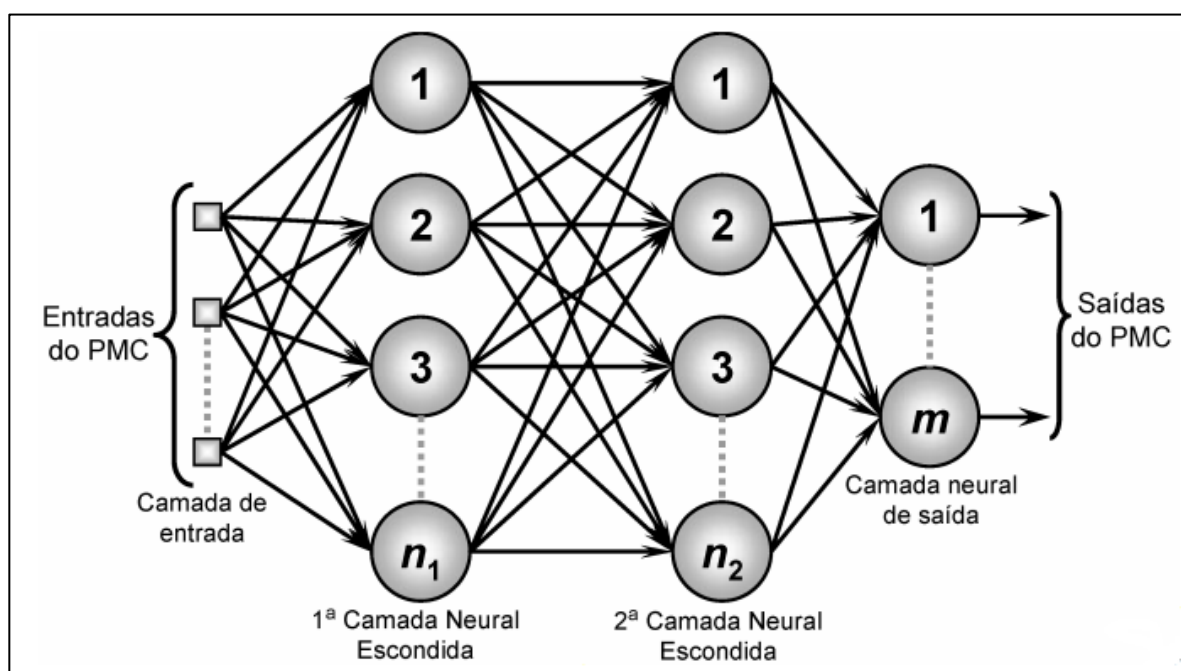
Fonte: SILVA, SPATTI E FLAUZINO (2010).

Apesar da rede *Perceptron*, em sua forma pura, ser muito utilizada para reconhecimento de padrões, para outras aplicações é necessária uma rede capaz de realizar mais do que apenas classificações de padrões. A rede *Perceptron* de múltipla

camada (PMC) soluciona tal inconveniente, podendo ser utilizada em aplicações de aproximação de funções, controle de processos, previsão de séries e otimização de sistemas (SILVA, SPATTI E FLAUZINO, 2010).

A rede PMC consiste de um conjunto de unidades sensoriais que constituem a camada de entrada, uma ou mais camadas ocultas de nós computacionais e uma camada de saída (HAYKIN, 2001). A Figura 5 exibe uma estrutura genérica de uma rede PMC.

Figura 5 – Rede *Perceptron* multicamadas



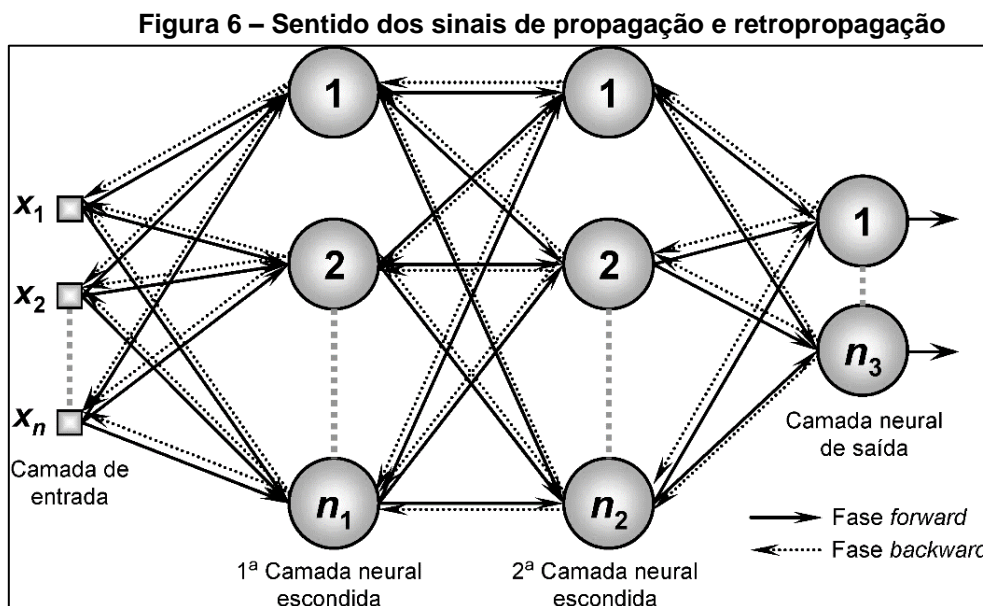
Fonte: SILVA, SPATTI E FLAUZINO (2010).

Pode-se observar que os sinais de entrada da rede percorrem as camadas escondidas até a camada de saída. A propagação dos sinais de entrada independe da quantidade de camadas intermediárias e a camada de saída é composta por múltiplos neurônios, cada um responsável por mapear uma saída da rede.

O ajuste dos pesos e do limiar de cada um dos neurônios da rede PMC é feito através do processo de treinamento supervisionado, com o algoritmo de retropropagação de erros. Este algoritmo é baseado na regra de aprendizagem por correção de erros (HAYKIN, 2001).

O processo de aprendizagem consiste em duas etapas: propagação e retropropagação. Na etapa de propagação, um conjunto de entrada é aplicado a rede

com pesos sinápticos fixos. Na etapa de retropropagação, os pesos sinápticos são ajustados. A Figura 6 exemplifica o processo de retropropagação.



Fonte: SILVA, SPATTI E FLAUZINO (2010).

2.5 Planejamento de Trajetória

Percurso é uma construção espacial, uma localidade no espaço que conduz de uma posição inicial para uma posição final. A trajetória é um percurso com limitações temporais (CORKE, 2011).

O planejamento da trajetória, de outro modo, determina características cinemáticas, como aceleração e velocidade do manipulador (MENDONÇA *et al.*, 2018). O problema do planejamento de trajetória concatena diferentes técnicas computacionais. Nesse contexto uma solução possível envolve a aplicação de algoritmos computacionais e dados coletados a partir de sensores, para obtenção do ângulo de um braço robótico ou a posição de um objeto na área de atuação do robô (CORKE, 2011).

O movimento de um robô pode ser suavizado controlando as velocidades e acelerações do movimento. Uma possível maneira de contornar tal problema é iniciar o movimento com uma aceleração constante, desenvolver uma velocidade de cruzeiro e por fim desacelerar constantemente. Esse processo evita os movimentos bruscos que levam a instabilidade.

O planejamento da trajetória pode ser realizado no espaço articular através de polinômios definidos a partir das condições de contorno iniciais como os ângulos iniciais e finais, já obtidos através da cinemática inversa (SOUZA et al., 2018). Uma das formas de realizar o planejamento de trajetória é através de uma Equação polinomial de quinta ordem (5), onde é possível obter a relação entre posição, velocidade e aceleração para o movimento desejado.

$$\theta(t) = c_0 + c_1t + c_2t^2 + c_3t^3 + c_4t^4 + c_5t^5 \quad (5)$$

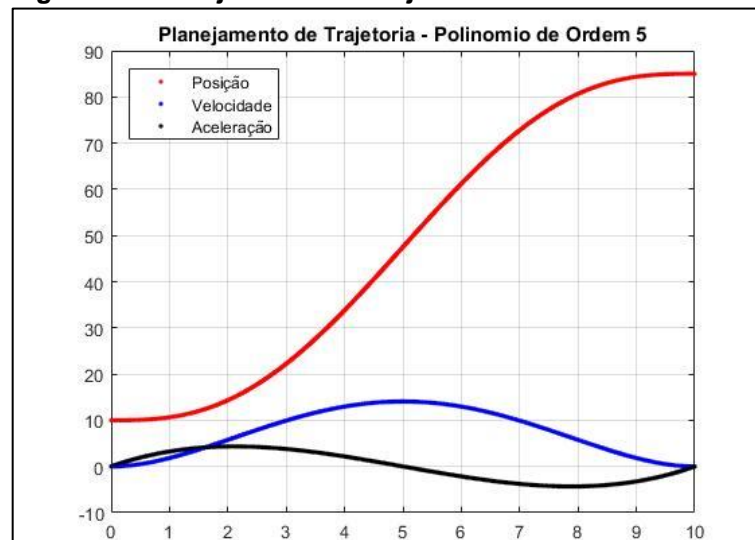
Utilizando o polinômio da Equação 5, é obtido o ângulo de controle do atuador para velocidades iniciais e finais, e acelerações definidas previamente. Derivando a equação da posição (5), encontramos a relação da velocidade no tempo (6), e derivando uma segunda vez, obtemos a relação entre aceleração e tempo (7).

$$\dot{\theta}(t) = c_1 + 2c_2t + 3c_3t^2 + 4c_4t^3 + 5c_5t^4 \quad (6)$$

$$\ddot{\theta}(t) = 2c_2 + 6c_3t + 12c_4t^2 + 20c_5t^3 \quad (7)$$

Essas equações, somadas as condições de contorno, permitem calcular os coeficientes de um polinômio de quinta ordem para cada movimento desejado. A Figura 7 mostra o desenvolvimento no tempo da posição, velocidade e aceleração utilizando o polinômio de quinta ordem.

Figura 7 – Planejamento de Trajetória – Polinômio de 5ª Ordem



Fonte: Autoria própria.

2.6 Acionamento dos atuadores

A etapa final do desenvolvimento do manipulador robótico autônomo é a execução dos movimentos realizados pelos atuadores. O acionamento dos atuadores segue o planejamento da trajetória desenvolvido. De maneira geral, o acionamento dos atuadores é realizado por um controlador. O controlador recebe os dados do computador, aciona os atuadores e coordena os movimentos de acordo com o *feedback* dos sensores (NIKU, 2011).

Os atuadores são responsáveis pela movimentação das juntas do manipulador robótico. Como citado anteriormente, os atuadores podem variar de acordo com os aspectos construtivos, sendo os mais comuns motores de corrente contínua, motores de passo, servo-motores, atuadores pneumáticos e atuadores hidráulicos (NIKU, 2011).

2.6.1 Modulação por Largura de Pulso (PWM)

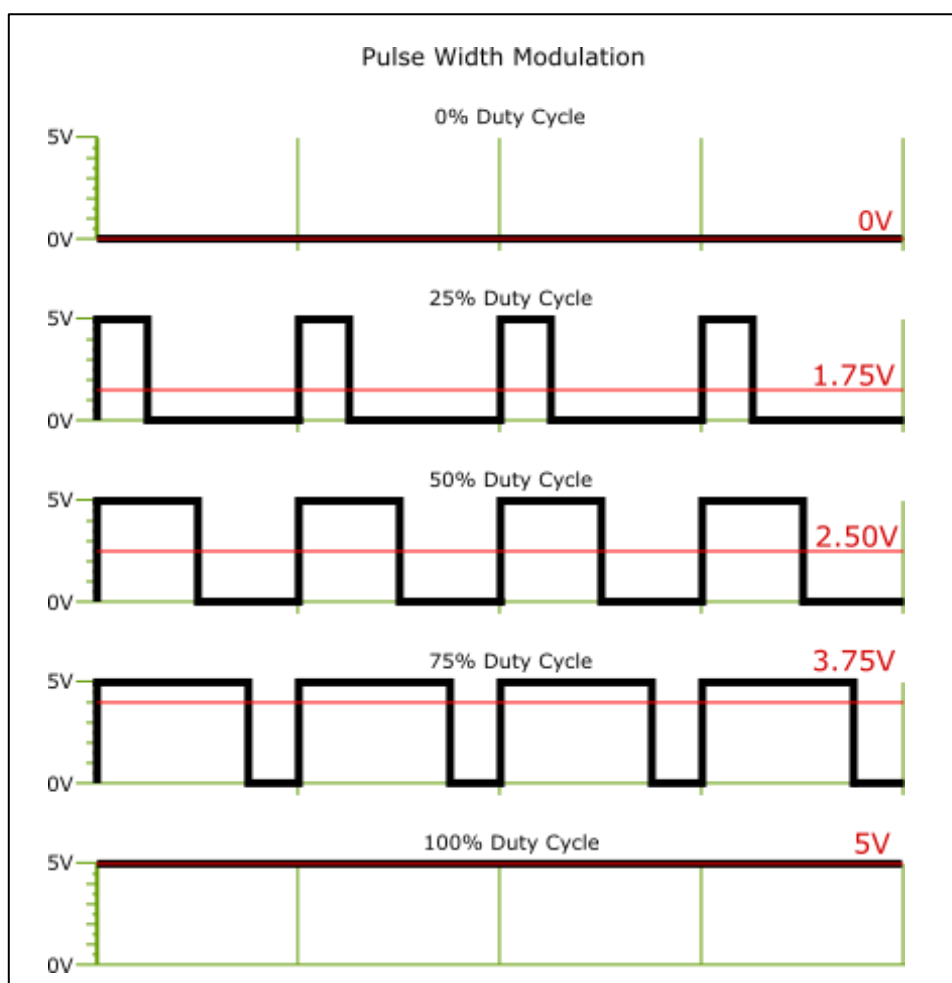
Em diversos casos, o acionamento dos atuadores pode ser realizado através do método PWM, utilizado para realizar o controle de tensão de saída através de pulsos únicos a cada período (AHMED, 2000). Para uma dada frequência, a largura de pulso ou *Duty cycle* pode variar, a fim de controlar a tensão de saída.

A tensão de saída resultante decorre da relação entre Período (T), Largura de Pulso (t_w) e *Duty Cycle*, representado nas Equações 8 e 9.

$$DutyCycle = 100 \times \frac{t_w}{T} \quad (8)$$

$$V_{PWM} = \frac{DutyCycle}{100} \times V_{CC} \quad (9)$$

Com isso, é possível controlar a tensão resultante (V_{PWM}) que varia em função do valor de *Duty Cycle* e a tensão de entrada (V_{CC}). A Figura 8 demonstra a variação no tempo da saída resultante e as larguras de pulso. Os pulsos variam de 0 a 100%, gerando saídas entre 0V a 5V, a linha vermelha representa a tensão resultante para cada intervalo de pulso.

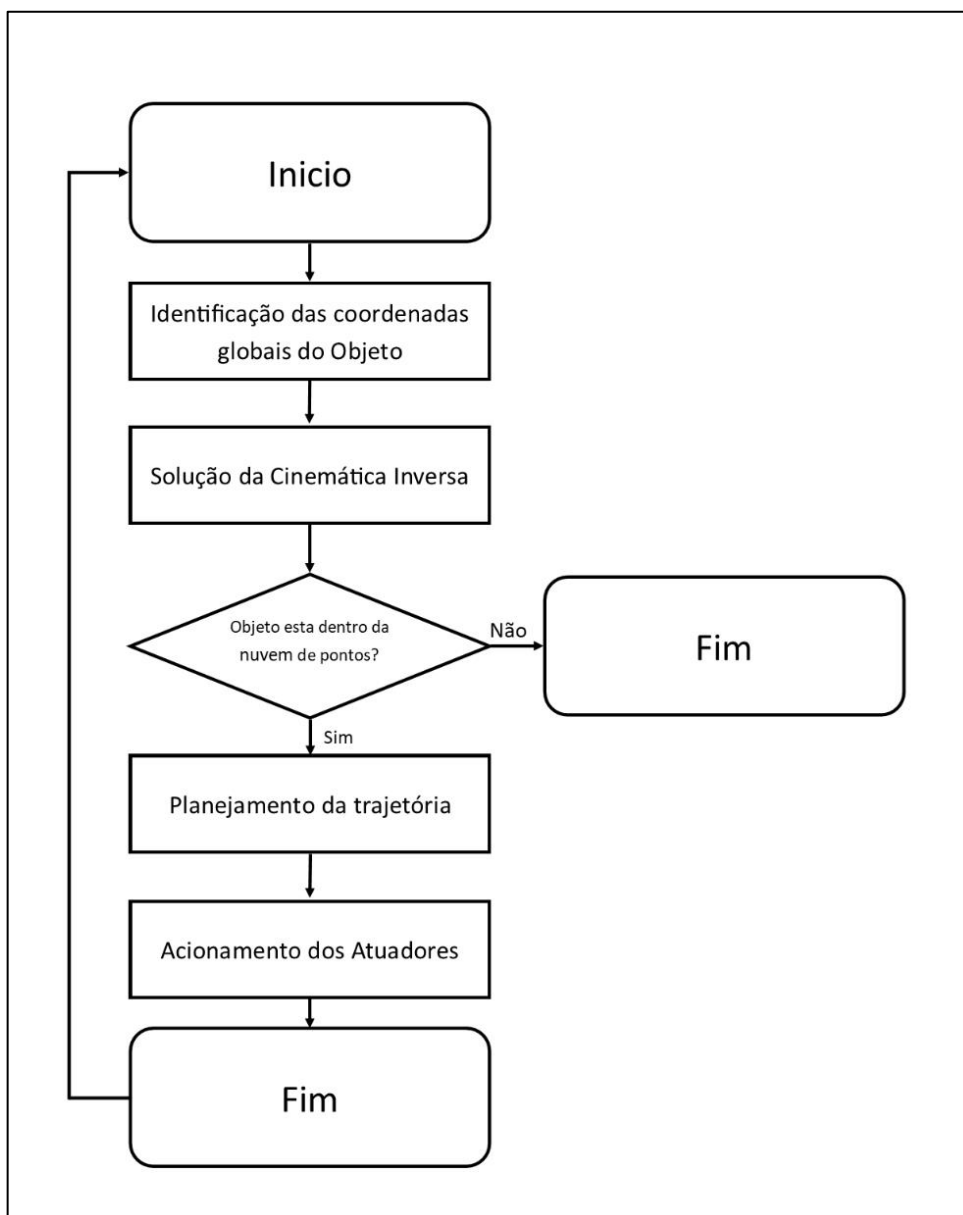
Figura 8 - Ciclos PWM

Fonte: Adaptado de HEATH (2017).

3 MATERIAIS E METODOS

O projeto foi dividido em cinco partes principais: manipulador robótico, sistema de visão computacional, solução da cinemática Inversa, planejamento de trajetória e acionamento dos atuadores. O fluxograma da Figura 9 demonstra o funcionamento do sistema integrado.

Figura 9 - Fluxograma: visão macro do sistema



Fonte: Própria Autoria.

3.1 Manipulador Robótico

A escolha do manipulador robótico utilizado nesse trabalho é baseada na capacidade de realizar um movimento em três dimensões. Para isso seria necessário utilizar um manipulador robótico com no mínimo 2 GDL. Para atingir uma área de atuação maior, foi decidido utilizar de 3 GDL. A Figura 10 exibe o manipulador escolhido, que utiliza juntas articuladas e atuadores do tipo servo-motores. A estrutura física foi feita com impressão 3D em plástico PLA (plástico de políácido láctico).

Figura 10– Manipulador Robótico utilizado



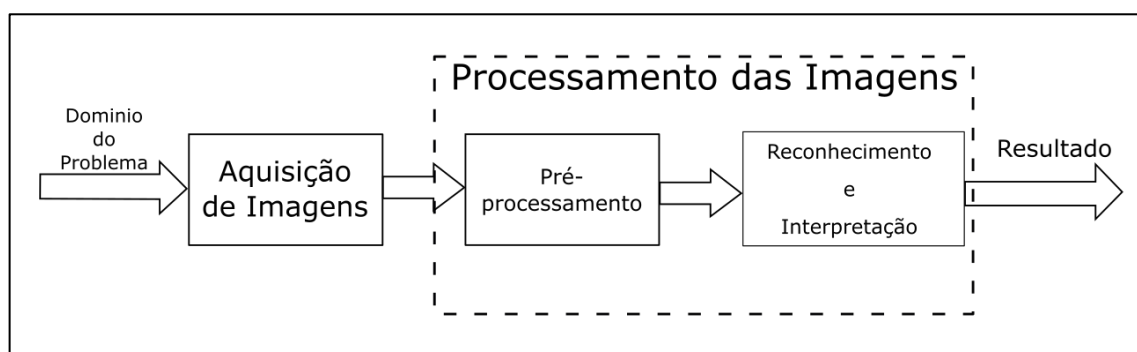
Fonte: Própria Autoria.

3.2 Sistema de Visão Computacional

O sistema de visão computacional será realizado com o auxílio da ferramenta computacional MatLab®, através da *toolbox* “*Computer Vision System*” e “*Image Acquisition*”. A *toolbox* disponibiliza ferramentas para a extração dos parâmetros intrínsecos e extrínsecos das câmeras utilizadas no sistema, reconhecimento do objeto de interesse e identificação das coordenadas globais do objeto.

O sistema de visão computacional desenvolvido nesse projeto segue uma estrutura de etapas baseada no modelo de Gonzalez e Woods (2000). Algumas das etapas do modelo original não foram abordadas por não serem de grande influência para o resultado final. O modelo utilizado é apresentado na Figura 11.

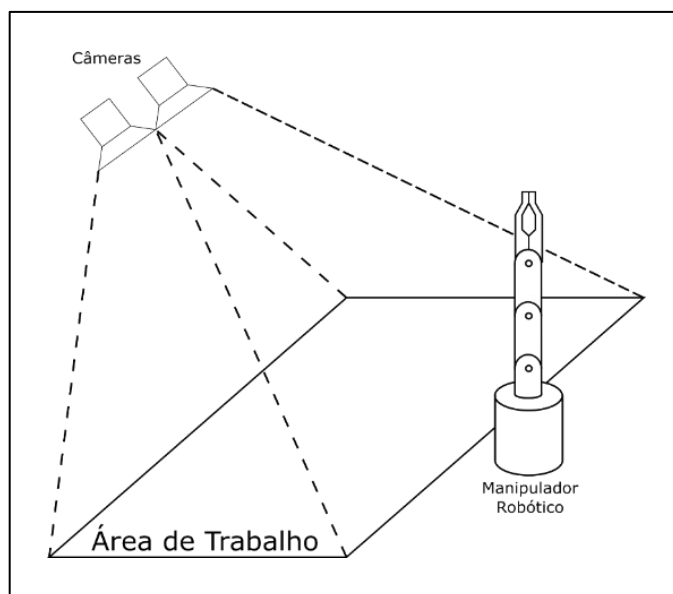
Figura 11 – Etapas de Processamento abordadas.



Fonte: Autoria Própria.

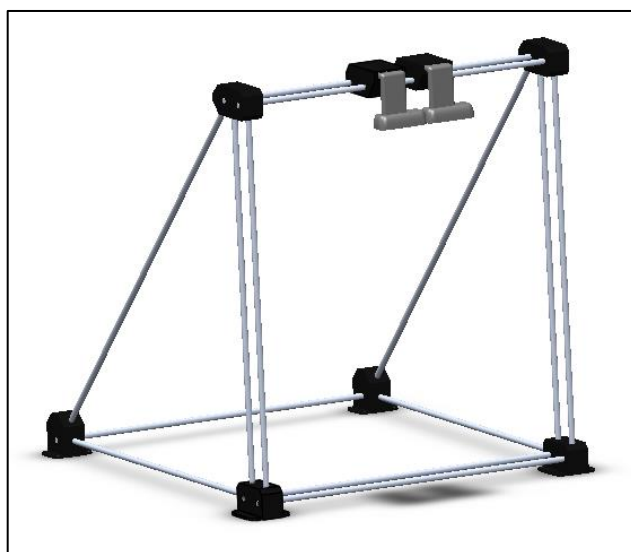
3.2.1 Posicionamento das Câmeras

Um dos aspectos importantes para o funcionamento correto do sistema de visão computacional é o posicionamento das câmeras que iram observar o ambiente e identificar o objeto a ser deslocado. Utilizando um sistema de visão estéreo (duas câmeras) é possível coletar dados adicionais aos obtidos com um sistema de câmera única, como profundidade dos elementos no ambiente (MATHWORKS, 2019). O esquema da Figura 12 representa a montagem do sistema final.

Figura 12 – Esquema de Montagem do Sistema.

Fonte: Própria Autoria.

Para atender as necessidades específicas do problema foi desenvolvido um suporte para as câmeras, tendo como parâmetros Altura, área de trabalho do robô, tamanho da imagem coletada, espaçamento entre as câmeras, entre outros. Utilizando a ferramenta *SolidWorks®*, foram modeladas partes individuais que compõem o suporte e impressas em plástico PLA. Para a construção do suporte também foram utilizadas barras de alumínio de diâmetro 5/16 polegadas e comprimento variado. A Figura 13 demonstra o modelo final do suporte projetado.

Figura 13 – Modelo do Suporte de Câmeras.

Fonte: Própria Autoria.

3.2.2 Aquisição de Imagens

A etapa de Aquisição de Imagens foi realizada com um conjunto de duas *Webcams C270*, da marca Logitech. Com o auxílio da *toolbox Computer Vision*, presente no software Matlab, foi realizado o processo de calibração das câmeras, extraindo os parâmetros intrínsecos e extrínsecos dos equipamentos. Esses parâmetros são utilizados futuramente, na etapa de processamento. A Figura 14 exhibe o modelo de câmeras utilizados no projeto.

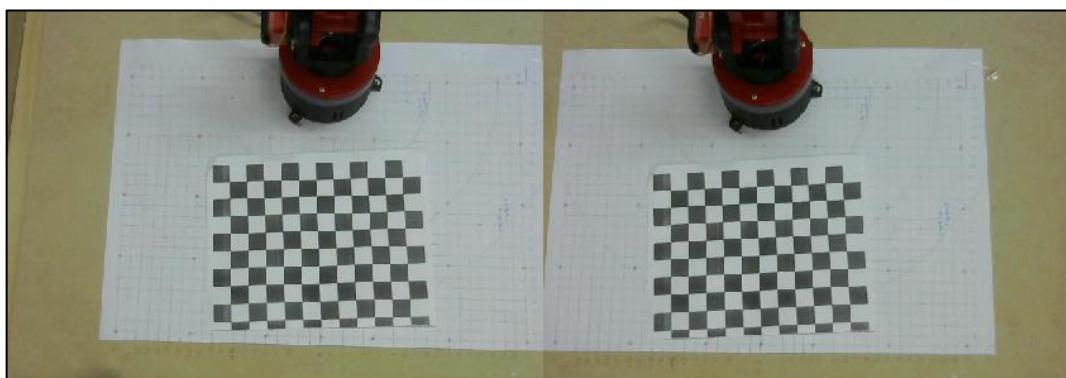
Figura 14 – Webcam Logitech C270



Fonte: <www.Logitech.com> Acesso em: 20/10/2019.

Utilizando um quadro de calibração, foram utilizadas entre 15 pares de imagens para identifica os parâmetros intrínsecos das câmeras. Um exemplo de par de imagem utilizado na calibração é exibido na Figura 15.

Figura 15 – Par de Imagem para Calibração

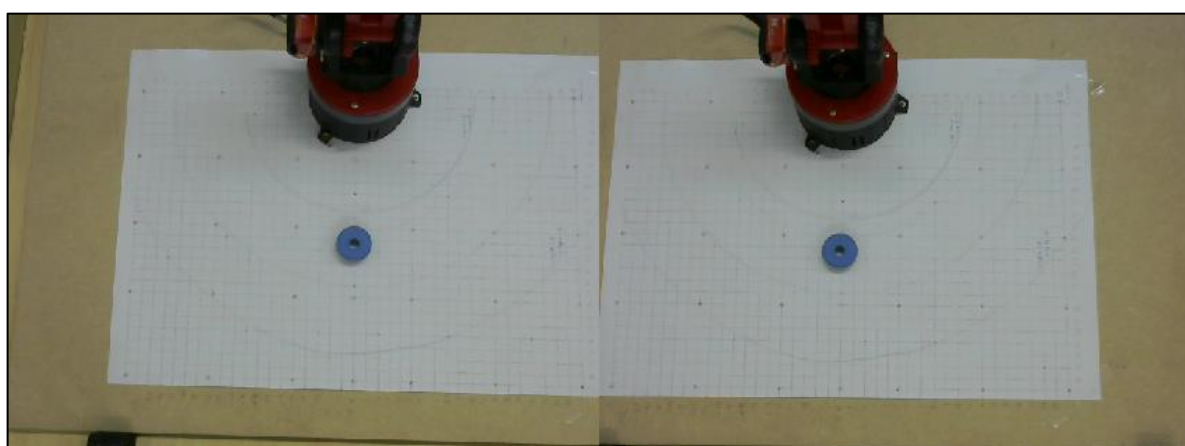


Fonte: Própria Autoria

No processo de calibração são identificados os vértices do tabuleiro e através da distância real dos ladrilhos são calculadas as distorções, matrizes de translação e erros de projeção. Uma vez realizado o processo de calibração das câmeras, as informações são armazenadas para futuras aplicações.

Para a aquisição das imagens que serão atualizadas no sistema foi utilizada a *toolbox Image Acquisition*. Na Figura 16 é apresentado um par de imagens de entrada do sistema.

Figura 16 – Imagens de Entrada



Fonte: Própria Autoria

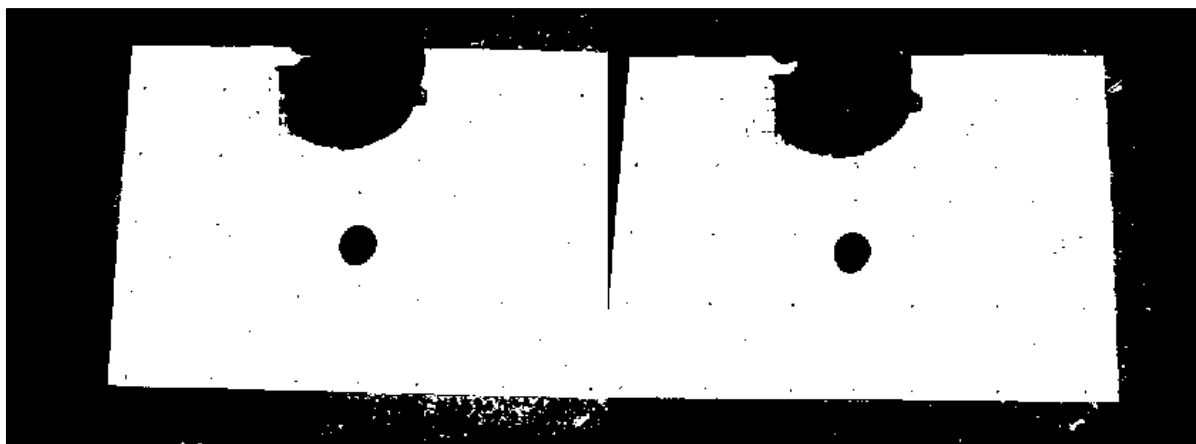
3.2.3 Processamento das imagens

O processamento de imagens digitais envolve procedimentos que são geralmente expressos em forma algorítmica (GONZALEZ E WOODS, 2000). O processamento de imagens é caracterizado por soluções específicas para cada problema.

O primeiro passo do processo foi a preparação das imagens que são utilizadas. Para melhorar o resultado final do processo de identificação foram realizadas duas manipulações na imagem de entrada.

A primeira manipulação realizada é a transformação da imagem de colorida (RGB) para Preto e Branco, utilizando a função *im2bw*. Foi escolhido fazer essa transformação devido as características do algoritmo de identificação de objeto utilizado na etapa seguinte. A Figura 17 mostra o par de imagens de entrada após a transformação.

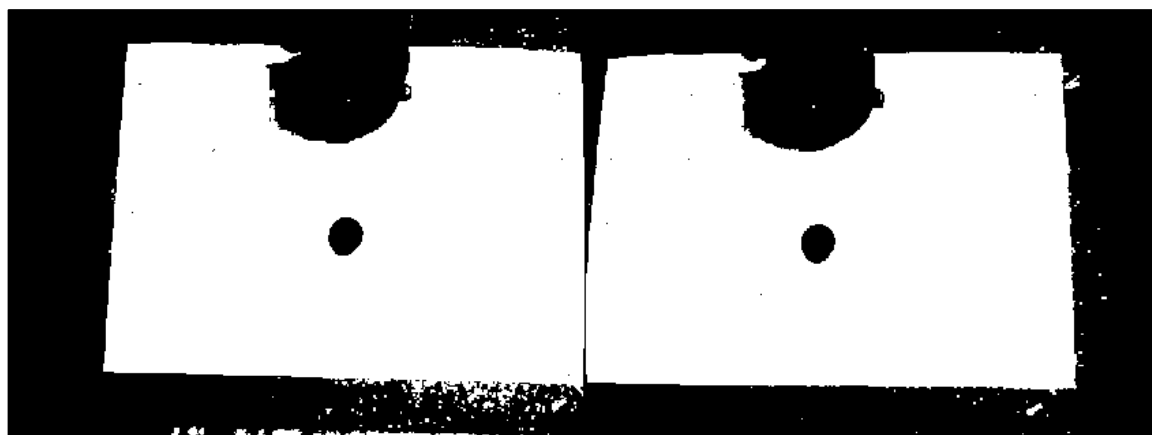
Figura 17 – Imagens de Entrada em Preto e Branco



Fonte: Própria Autoria

A segunda etapa é a remoção das distorções causadas pela lente das câmeras, utilizando os parâmetros intrínsecos e extrínsecos obtidos através da calibração das câmeras e a função *undistortImage*. Esse processo tende a diminuir os erros de projeção da imagem. O resultado da manipulação pode ser visto na Figura 18.

Figura 18 – Imagem distorcida



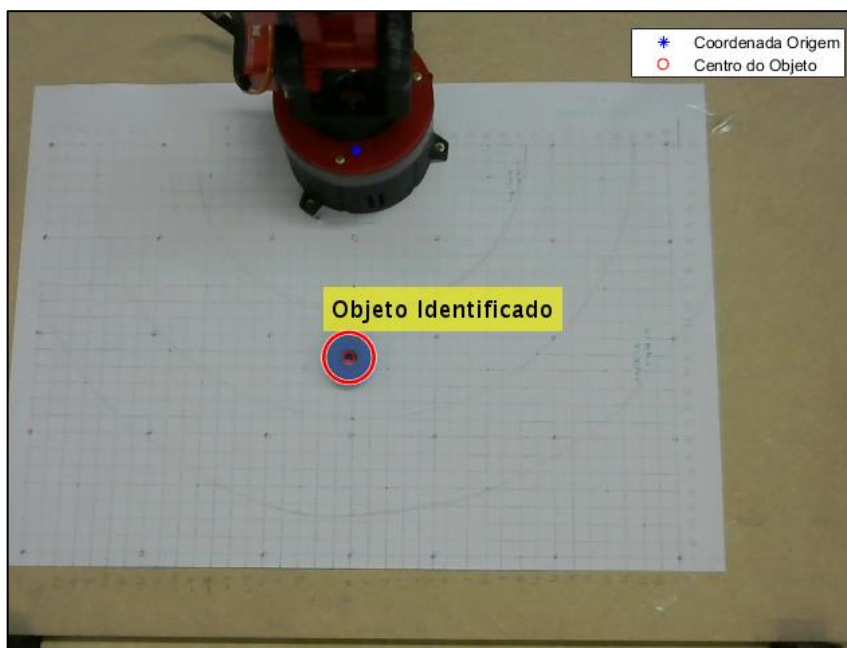
Fonte: Própria Autoria

Após o pré-processamento da imagem, é realizado a identificação do objeto desejado. Essa etapa pode ser realizada de diversas formas, variando de acordo com o objeto e resultados desejados. Para esse trabalho, foi decidido realizar a identificação de moedas, utilizando a função *imfindcircles*. A função escolhida utiliza a Transformada de Hough, um método de processamento de imagens e

reconhecimento de padrões, utilizado para detectar formas geométricas como linhas e círculos (LESTRIANDOKO e SADIKIN, 2016).

Como resultado da função de *imfindcircles*, são retornados o raio e as coordenadas do pixel central do círculo. A Figura 19 exibe a posição do objeto identificado.

Figura 19 - Imagem com o objeto identificado



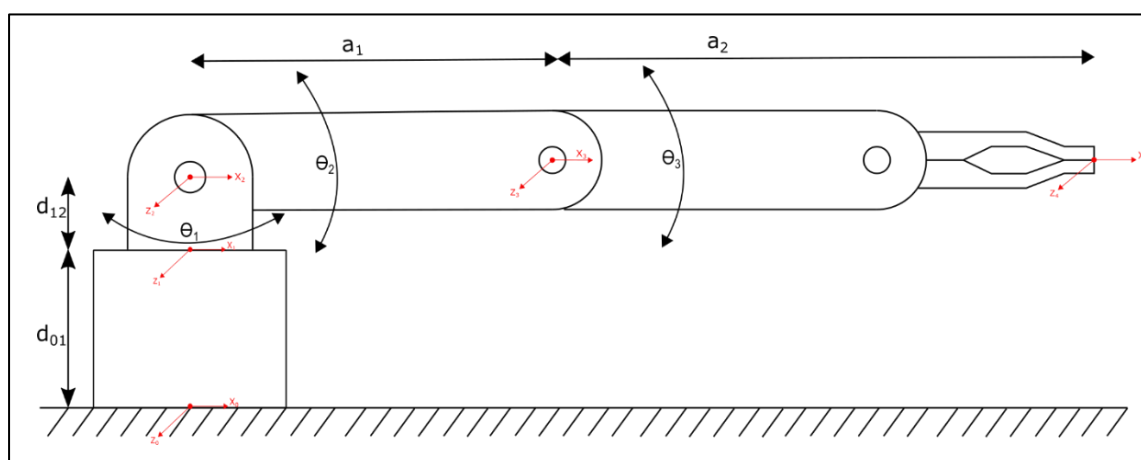
Fonte: Própria Autoria.

A partir das coordenadas do centro do objeto, em pixels (U , V) e da coordenada de referência definida, é realizado uma serie de manipulações matemáticas para calcular a posição real do objeto. As coordenadas são enviadas para o próximo subsistema, onde são utilizadas como *input* para a rede neural que calcula os ângulos a serem exercidos pelo robô.

3.3 Análise Cinemática

Definido o manipulador robótico a ser utilizado, foi realizado a construção do modelo de Denavit-Hartenberg para solução da cinemática direta. Figura 20 exibe a posição adotada para modelagem e a Tabela 1, os parâmetros utilizados para o modelo. No modelo, Θ_1 se refere ao movimento de rotação da base, Θ_2 ao movimento do braço e Θ_3 ao movimento do antebraço.

Figura 20 - Modelo D-H



Fonte: Autoria Própria

Tabela 1: Parâmetros do modelo D-H

Segmento	A (°)	a (mm)	Θ (°)	d (mm)
0-1	0	0	0	L1
1-2	90	0	0	L2
2-3	0	L3	0	0
3-4	-90	L4	90	0

Fonte: Autoria Própria

3.4 Cinemática inversa

Para a solução da cinemática inversa, optou-se pela criação de uma Rede Neural Artificial *Perceptron* multicamadas, utilizando uma topologia com três camadas escondidas, como apresentado por Dalmedico et al. (2018). Os dados de treinamento da rede serão obtidos através da cinemática direta utilizando o modelo DH. O

processo de validação cruzada foi aplicado para selecionar quantos neurônios seriam utilizados em cada camada e a taxa de aprendizado. O fator determinante para a seleção da rede foi a performance da rede, com performance mínima de $1,0E-05$. Para o treinamento das Redes Neurais Artificiais foi utilizado um processador Intel® Core™ i5-4200M CPU @2.5GHz com 8GB de memória RAM.

A Tabela 2 mostra os resultados da primeira etapa de validação cruzada.

Tabela 2: Primeira etapa de validação cruzada

Topologia	Camada 1	Camada 2	Camada 3	Épocas	Taxa Aprendizado	Performance
1	12	12	12	191	0,01	9,61E-06
2	12	12	12	201	0,05	9,97E-06
3	10	10	10	213	0,01	9,93E-06
4	10	10	10	311	0,05	9,97E-06
5	15	15	15	118	0,01	9,95E-06
6	15	15	15	143	0,05	9,99E-06
7	20	20	20	72	0,01	9,82E-06
8	20	20	20	93	0,05	9,99E-06
9	20	15	10	161	0,01	9,68E-06
10	20	15	10	112	0,05	9,90E-06
11	10	15	20	213	0,01	9,94E-06
12	10	15	20	228	0,05	9,81E-06
13	5	5	5	3000	0,01	1,90E-04
14	5	5	5	3000	0,05	2,62E-04
15	10	13	15	159	0,01	9,47E-06
16	10	13	15	218	0,05	9,83E-06
17	15	13	10	191	0,01	9,96E-06
18	15	13	10	169	0,05	9,91E-06
19	13	14	15	117	0,01	9,90E-06
20	13	14	15	108	0,05	9,91E-06
21	15	14	13	163	0,01	9,98E-06
22	15	14	13	104	0,05	9,59E-06

Fonte: Autoria Própria

Das topologias testadas, a com melhor resultado foi a de número 15, com 10 neurônios na primeira camada, 13 neurônios na segunda camada e 15 neurônios da terceira camada, taxa de aprendizado de 0,01 e performance de $9,47E-06$.

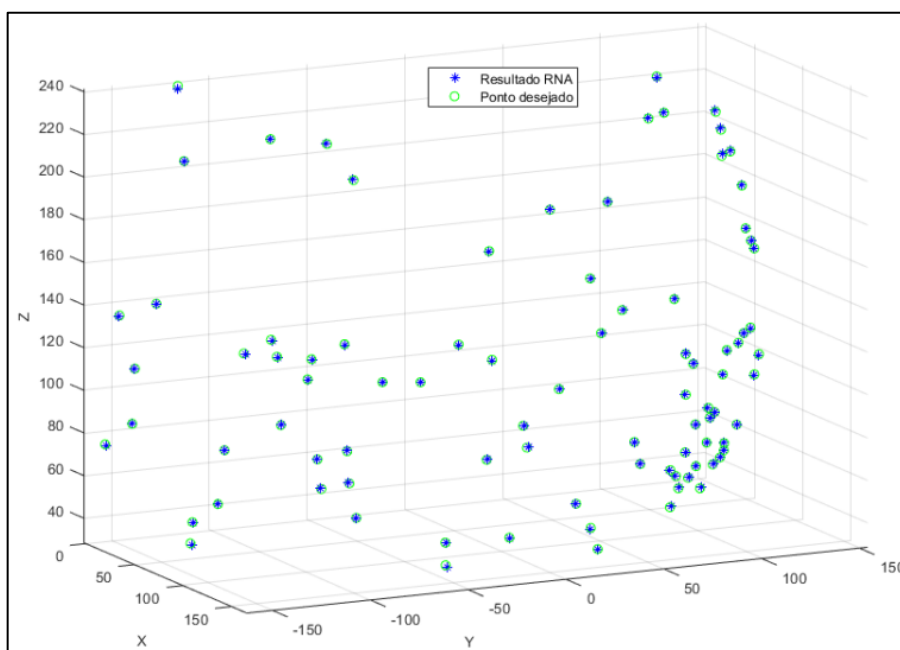
Selecionada a topologia, foram treinadas diversas redes com valores iniciais aleatórios para definir qual seria utilizada no sistema final. A Tabela 3 exibe os resultados obtidos.

Tabela 3: Validação cruzada da topologia selecionada

Topologia	Épocas	Performance	Tempo de treino
1	180	9,88E-06	00:00:50
2	173	9,87E-06	00:00:29
3	128	9,54E-06	00:00:18
4	201	9,84E-06	00:00:27
5	180	9,98E-06	00:00:27
6	133	9,87E-06	00:00:26
7	143	9,95E-06	00:00:29
8	158	9,90E-06	00:00:23
9	135	9,91E-06	00:00:21
10	221	9,82E-06	00:00:31

Fonte: Autoria Própria

A rede de número 3 apresentou a melhor performance e foi selecionada para compor o algoritmo final. A Figura 22 apresenta os pontos de testes desejados e obtidos de forma gráfica. A Tabela 4 exibe os resultados de performance e erro da rede escolhida

Figura 21 - Resultado da Rede Neural Escolhida

Fonte: Autoria Própria

Tabela 4: Resultado de validação da RNA.

DH (mm)			RNA (mm)			Erro (mm)		
X	Y	Z	X	Y	Z	X	Y	Z
90,39	124,41	73,98	90,11	124,45	73,87	0,28	-0,04	0,11
99,32	89,42	228,14	99,42	89,44	227,62	-0,11	-0,01	0,52
33,23	-156,35	90,42	32,68	-156,21	89,98	0,55	0,14	0,43
0,00	-143,15	68,33	0,33	-143,61	68,81	-0,33	-0,45	-0,48
141,53	-63,01	99,66	141,37	-63,08	99,47	0,16	-0,07	0,19
0,00	-125,85	29,69	-1,06	-125,31	28,84	-1,06	0,55	0,85
134,14	14,10	53,76	134,39	14,24	53,86	-0,25	-0,14	-0,10
145,24	15,27	58,27	145,43	15,40	58,29	-0,19	-0,14	-0,02
120,99	-25,72	241,63	121,03	-26,23	241,25	-0,03	-0,51	0,38
102,90	114,28	73,98	102,83	114,26	73,82	0,07	0,02	0,16
129,78	74,93	83,69	130,12	75,11	83,69	-0,34	-0,18	-0,01
138,89	-45,13	58,27	138,88	-45,27	58,20	0,01	-0,14	0,06
73,02	126,47	58,27	72,67	126,59	58,40	0,35	-0,12	-0,13
160,56	34,13	107,40	160,96	33,88	107,51	-0,40	0,25	-0,11
146,79	47,69	182,53	146,47	47,79	182,67	0,31	-0,10	-0,15
115,13	-103,67	99,66	115,09	-103,54	99,78	0,04	0,12	-0,12
79,96	138,50	132,75	79,93	138,61	132,59	0,03	-0,11	0,17
93,93	-129,29	149,51	93,67	-129,57	149,81	0,26	-0,28	-0,30
47,69	146,79	182,53	48,15	146,83	182,81	-0,45	-0,04	-0,28
48,81	-150,21	166,16	48,73	-150,29	165,94	0,08	-0,08	0,22

Fonte: Autoria Própria

Conforme Haykin (2001), os pontos selecionados para teste não são os mesmos utilizados para treinamento da rede. A rede treinada apresentou erro médio de 0,02 mm.

Com a rede treinada e validada, é necessário apenas fornecer como *Input* da rede, as coordenadas finais do objeto obtidas pelo sistema de visão computacional, e a saída serão os ângulos que devem ser realizados pelos atuadores.

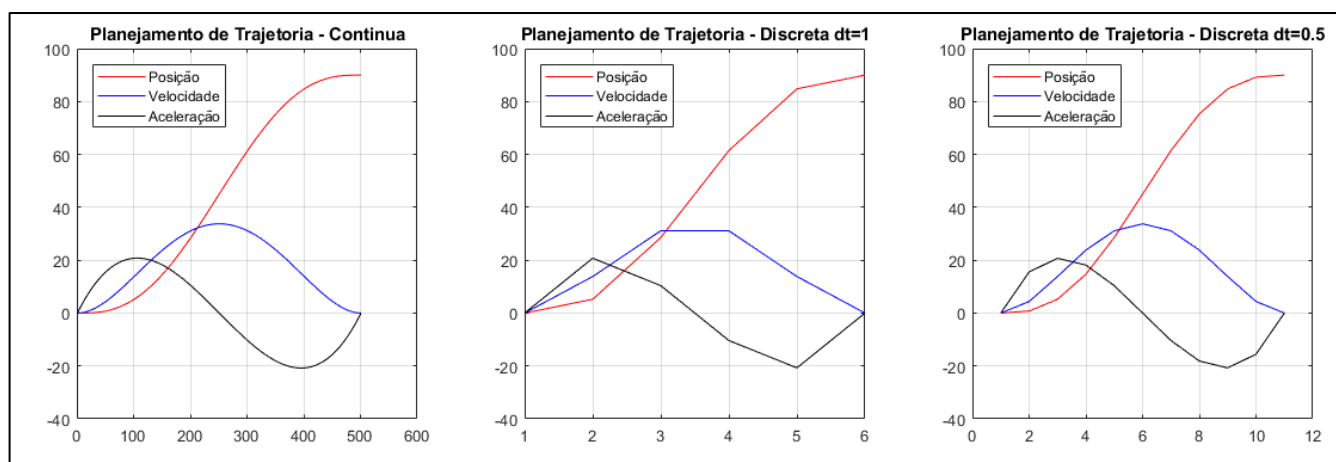
3.5 Planejamento de trajetória

O planejamento de trajetória é calculado a partir dos ângulos obtidos através da Rede Neural Artificial e a duração do movimento. Para o planejamento da trajetória nesse trabalho foi aplicado o polinômio de quinta ordem da Equação 4, onde obtido uma trajetória contínua. Para uma real aplicação nos servo-motores é necessário

realizar a discretização do movimento, uma vez que seu acionamento é realizado via pulsos PWM e servos-motores possuem apenas controle de posição.

Esse processo de discretização adiciona uma variável temporal que divide a trajetória em pequenos movimentos. O número de movimentos utilizados para realizar a trajetória completa influencia na percepção de continuidade e velocidade do atuador. Uma discretização com grande intervalo de tempo resulta em um movimento mais segmentado, com pausas longas ou altas velocidades. Para um movimento suave, é necessário um intervalo de tempo baixo, resultando em um movimento complexo que demanda maior poder de processamento. A Figura 23 apresenta um comparativo entre uma trajetória contínua, discreta com intervalo de 1 segundo e discreta com intervalo de 0,5 segundos.

Figura 22 – Comparativo entre Trajetórias Contínua e Discretas



Fonte: Própria Autoria.

Uma vez geradas as trajetórias desejadas, os ângulos de cada ponto no gráfico discretizado são enviados a próxima etapa, para que seja realizado o acionamento dos motores.

3.6 Acionamento dos Atuadores

A partir do vetor resultante do planejamento de trajetória, é realizado o acionamento dos atuadores. Os ângulos são enviados do Matlab® ao microcontrolador, onde o movimento é realizado no intervalo de tempo determinado pela discretização da trajetória.

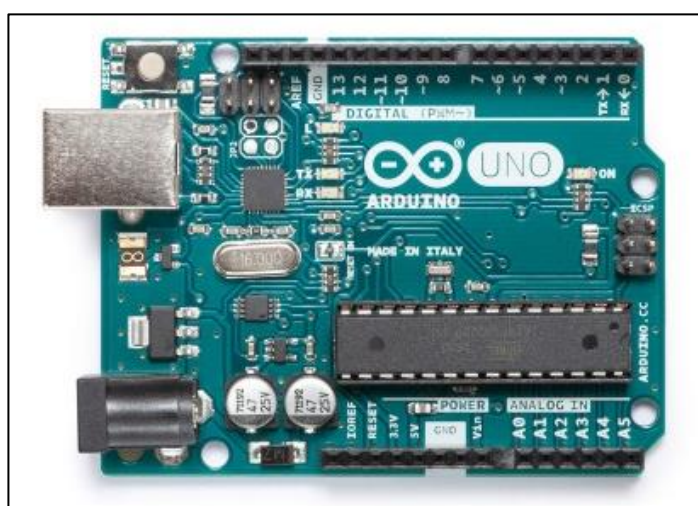
Foi decido pela utilização de um microcontrolador Arduino UNO com comunicação serial com o software Matlab®. Os atuadores utilizados no manipulador serão servo motores.

3.6.1 Arduino UNO

Arduino é um microcontrolador que pode ser programado para processar entradas e saídas entre o dispositivo e os componentes externos conectados a ele (MCROBERTS, 2010). A plataforma surgiu em 2005, na Itália, com a proposta de auxiliar estudantes de design a trabalhar com tecnologia de forma fácil e barata. Desde então, novos modelos com diferentes especificações foram disponibilizados no mercado, cada qual com suas vantagens e desvantagens. A Figura 24 apresenta o microcontrolador a ser utilizado.

O modelo UNO R3, utiliza um microcontrolador ATmega328P, com filosofia RISC (*reduced instruction set computer*), velocidade de *clock* de 16MHz e memória flash de 8KB, ideal para projetos que não exigem grande poder computacional. A placa disponibiliza 14 pinos digitais, que podem ser utilizados como entrada ou saída, sendo 6 delas, portas com suporte PWM, diversos protocolos de comunicação, como serial, SPI e I2C/TWI e conector de programação serial *in-circuit* (ICSP).

Figura 23 – Arduino UNO



Fonte: <Arduino.cc> Acesso em: 19/07/2018.

3.6.2 Servo-motores

Similares aos motores de corrente contínua, os servo-motores possuem a capacidade de rotacionar seu eixo para determinada posição especificada e possuem diversas aplicações, comuns em aplicações de robótica e aeromodelismo (MATARIC, 2007). Construídos a partir de motores de corrente contínua adaptados para realizar movimento em ambas as direções e permanecerem travados em uma posição específica, o acionamento dos servos motores pode realizado através de Modulação por largura de pulso (PWM). A Figura 25 apresenta o servo motor escolhido.

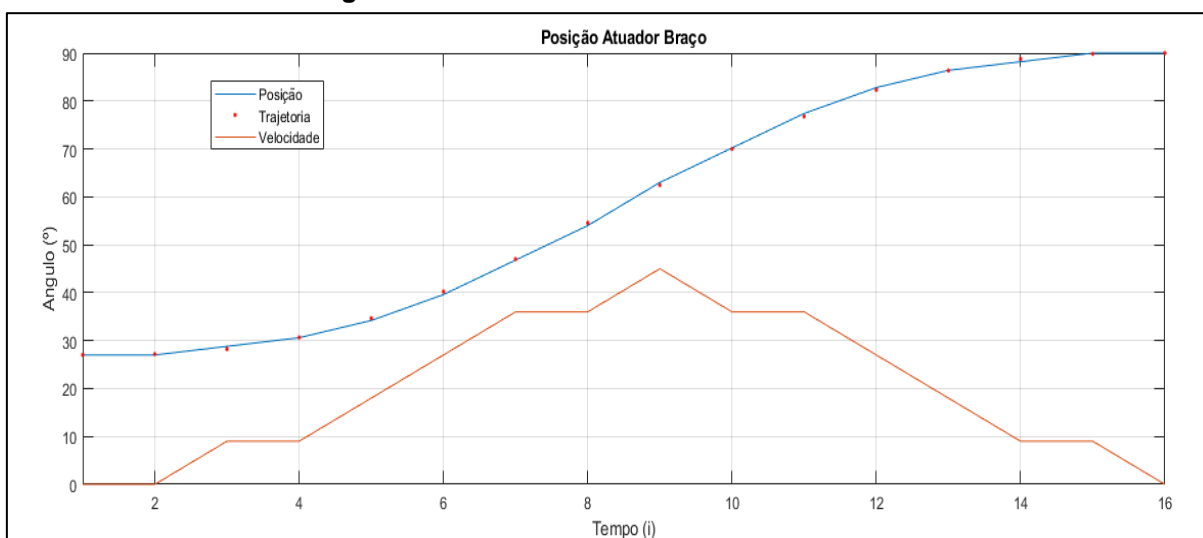
Figura 24 - Servo Motor JX PDI-6221MG



Fonte: < www.filipflop.com/ > Acesso em: 16/10/2018.

Este trabalho leva em consideração apenas os aspectos cinemáticos de um manipulador robótico. Os aspectos dinâmicos como análise de forças podem ser encontrados na literatura, como por exemplo no livro “Robótica”, de CRAIG (2006).

Na Figura 26 é exibido o gráfico do movimento realizado por um dos atuadores, onde é possível notar a proximidade entre o ângulo desejado e o executado pelo atuador.

Figura 25 – Gráfico de movimento do atuador.**Fonte: Própria Autoria.**

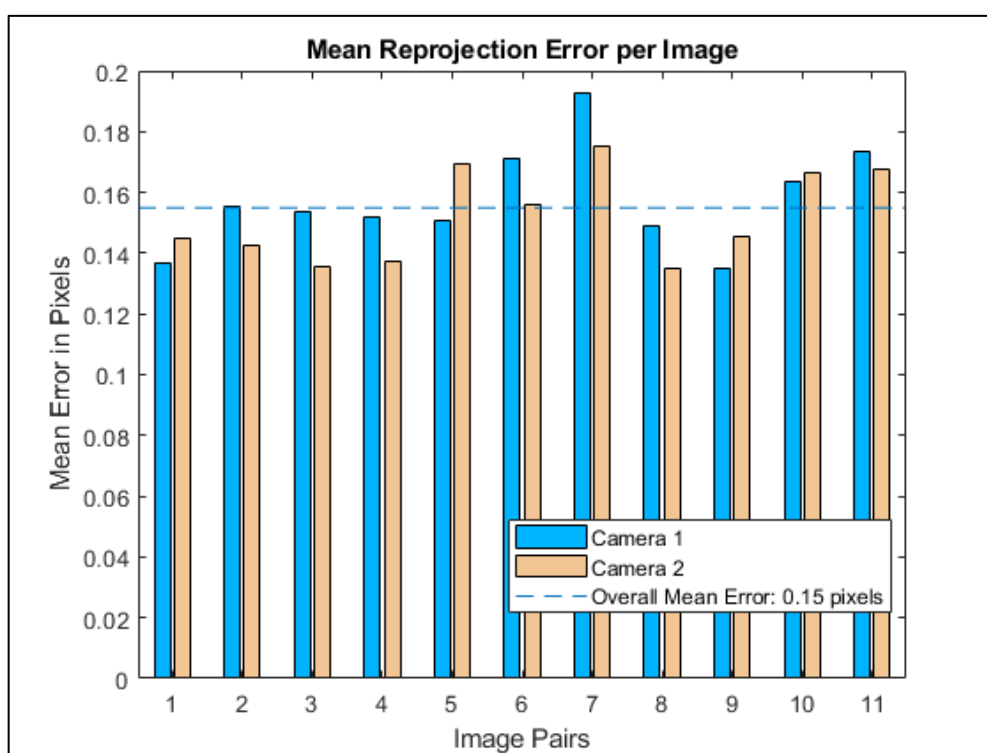
4 RESULTADOS

Nesta seção são apresentados os resultados individuais dos subsistemas desenvolvidos e resultados dos sistemas atuando em conjunto.

4.1 Visão Computacional

A análise dos resultados do sistema de visão Computacional se divide em três etapas: Calibração, Reconhecimento de imagem e Coordenadas. A Figura 27 exibe o histograma de erro de calibração das câmeras em pixels. O gráfico utiliza os pontos de calibração identificados nos pares de imagens coletadas e relaciona com os valores projetados pelo sistema de calibração.

Figura 26 - Histograma de Erro de Calibração.

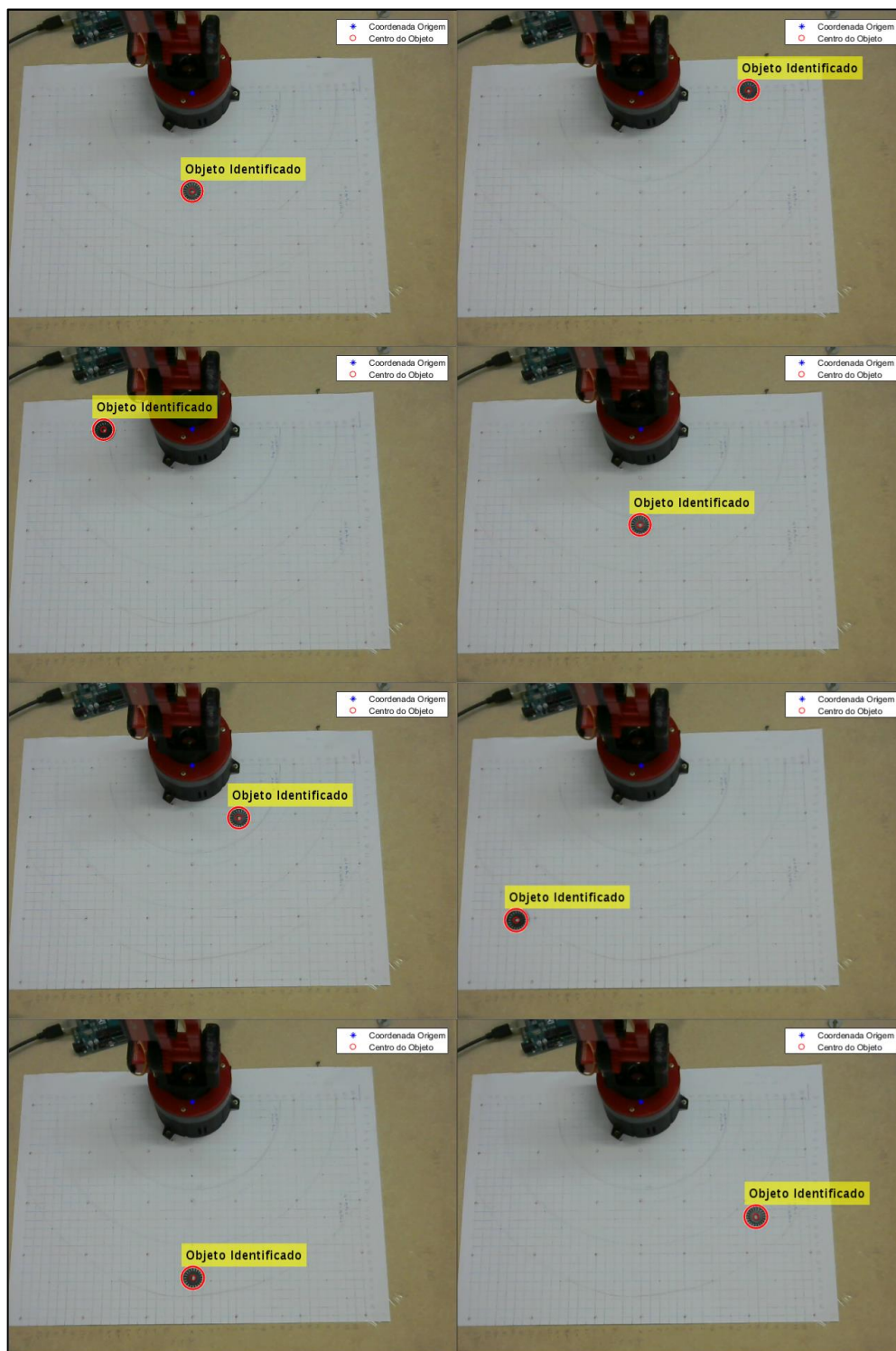


Fonte: Autoria Própria.

A partir do gráfico é observado que o erro médio entre os pontos projetados nas imagens em relação aos pontos reais é de 0,15 pixels. O erro alcançado é considerado pequeno e com isso define-se que o processo de calibração está dentro dos padrões desejados.

Para a identificação do objeto de trabalho foi definido que o disco possui tamanho constante. A Figura 28 exibe os resultados da etapa de reconhecimento de imagem, onde são exibidos alguns testes onde o objeto de trabalho foi identificado.

Figura 27 - Imagens Reconhecimento de Objeto.



Fonte: Autoria Própria.

Após a Identificação do objeto e extração das coordenadas foram comparados os valores calculados com as posições reais. A Tabela 5 que relaciona as coordenadas calculadas com as coordenadas reais e exibe o erro entre elas.

Tabela 5: Coordenadas Calculadas x Coordenadas Reais.

Coordenada Real			Coordenada Calculada			Erro		
X(mm)	Y(mm)	Z(mm)	X(mm)	Y(mm)	Z(mm)	X(mm)	Y(mm)	Z(mm)
120	0	0	121	-2	0	1	2	0
0	125	0	-4	131	-48	4	6	48
66	57	0	65	60	-20	1	3	20
183	-141	0	188	-152	-5	5	11	5
0	-107	0	-1	-113	-37	1	6	37
119	0	131	81	-14	98	38	14	33
129	0	260	45	-41	188	84	41	72
140	130	0	156	136	15	16	6	15

Fonte: Autoria Própria.

A análise da Tabela revela que nas coordenadas X e Y, o erro é menor para pontos próximos da origem (0, 0). O mesmo acontece para a coordenada Z, objetos mais próximos da câmera apresentam erro maior. A presença de erro é decorrente do posicionamento das câmeras, que origina um erro de paralaxe. Uma das soluções para esse problema é a modificação da posição das câmeras ou uma abordagem utilizando métodos computacionais inteligentes, como a implementação de uma Rede Neural Artificial.

Considerando que esse trabalho propõem o uso uma esteira de transporte com altura fixa e objetos de trabalhos com mesma altura, o sistema de transformação de coordenadas foi considerado validado para ser utilizado no sistema final.

Através dos resultados apresentados, o sistema de visão computacional desenvolvido apresenta resultados positivos e pronto para a integração do sistema final.

4.2 Modelo de Denavit-Hartenberg (D-H)

Após a modelagem do manipulador robótico, foi realizada a validação do modelo, comparando as coordenadas obtidas através do modelo com o modelo físico, para alguns ângulos determinados. A Tabela 6 demonstra os resultados.

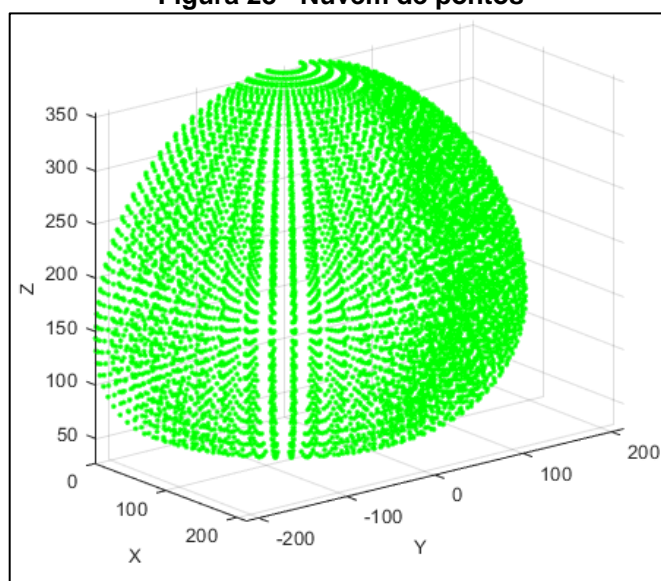
Tabela 6: Validação do modelo D-H

Ângulos Aplicados (°)			Modelo D-H (mm)			Manipulador			Erro		
Θ_1	Θ_2	Θ_3	X	Y	Z	X	Y	Z	X	Y	Z
0	0	0	220	0	140	225	0	140	5	0	0
0	0	90	100	0	20	95	5	25	5	5	5
0	90	0	0	0	360	0	0	355	0	0	5
0	90	90	120	0	240	115	0	240	5	0	0
90	0	0	0	220	140	0	220	100	0	0	40
90	0	90	0	100	20	0	100	20	0	0	0
90	90	0	0	0	360	0	0	360	0	0	0
90	90	90	0	120	240	0	140	235	0	20	5
-90	0	0	0	-220	140	10	-210	135	10	10	5
-90	0	90	0	-100	20	10	-100	20	10	0	0
-90	90	0	0	0	360	10	0	360	10	0	0
-90	90	90	0	-120	240	10	-100	240	10	20	0

Fonte: Autoria Própria

Com o modelo D-H validado, foi construída uma nuvem de pontos para visualizar a área de atuação do robô. A Figura 21 mostra a nuvem de pontos obtida.

Figura 28 - Nuvem de pontos



Fonte: Autoria Própria

4.2 Cinemática Inversa

A Rede Neural Artificial treinada foi testada com coordenadas aleatórias na área de trabalho do robô. Com os ângulos obtidos, foi calculada a cinemática direta do manipulador através do modelo DH construído. A Tabela 7 exibe os resultados dos testes e o erro entre a coordenada fornecida e a obtida com os ângulos da rede. Observa-se que a Tabela 7 difere da Tabela 4, pois utiliza como dados entrada as coordenadas obtidas pelo Sistema de Visão Computacional.

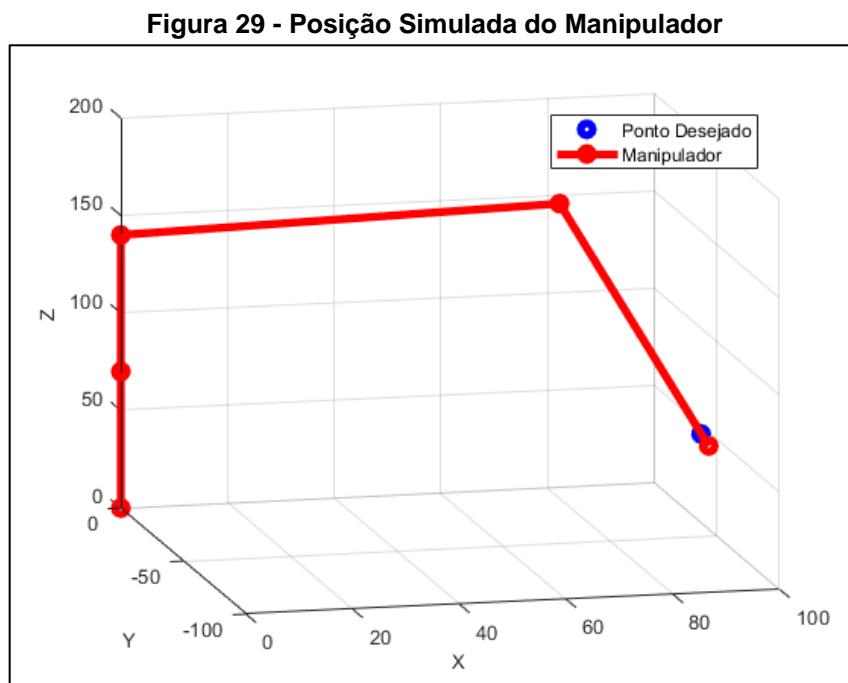
Tabela 7: Resultados do teste da RNA.

Coordenada Desejada (mm)			Ângulo Calculado (°)			Coordenada DH (mm)			Erro (mm)		
X	Y	Z	$\Theta 1$	$\Theta 2$	$\Theta 3$	X	Y	Z	ΔX	ΔY	ΔZ
0	180	120	90	29	-65	0,00	184,50	117,95	0,00	4,50	2,05
120	120	200	45	55	-66	123,85	125,85	199,02	3,85	5,85	0,98
20	150	250	82	68	-59	21,71	154,47	251,49	1,71	4,47	1,49
90	-80	70	-42	24	-99	90,97	-81,91	64,76	0,97	1,91	5,24
160	110	140	34	27	-51	164,75	111,13	136,59	4,75	1,13	3,41
90	0	30	0	3	-98	89,40	0,00	25,69	0,60	0,00	4,31
170	-120	170	-35	21	-24	174,64	-122,28	169,56	4,64	2,28	0,44
130	150	124	49	19	-44	133,38	153,44	121,84	3,38	3,44	2,16
180	80	65	24	-9	-23	183,20	81,56	60,77	3,20	1,56	4,23
210	20	140	5	12	-22	215,17	18,82	139,95	5,17	1,18	0,05
200	-40	120	-11	13	-35	204,86	-39,82	117,54	4,86	0,18	2,46
130	-50	230	-21	76	-79	134,46	-51,61	230,75	4,46	1,61	0,75

Fonte: Autoria Própria.

Os dados da Tabela 6 comprovam a eficiência da rede gerada, com erro médio entre as coordenadas desejadas e obtidas de 2,59 mm. Considerando os aspectos físicos e construtivos do robô utilizado, a rede utilizada apresentou bons resultados e com baixo erro. Entretanto para aplicações industriais onde a precisão é necessária, a rede não poderia ser aplicada.

O sistema de solução da cinemática inversa também exibe como saída uma simulação da posição do robô na posição final. A Figura 29 apresenta uma das simulações geradas.

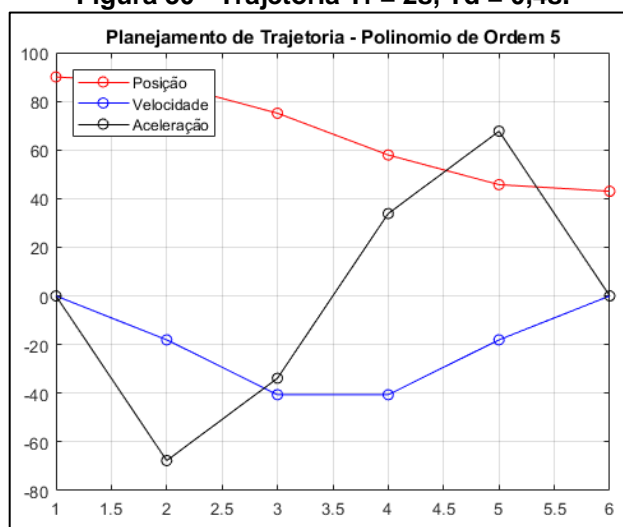


Fonte: Autoria Própria.

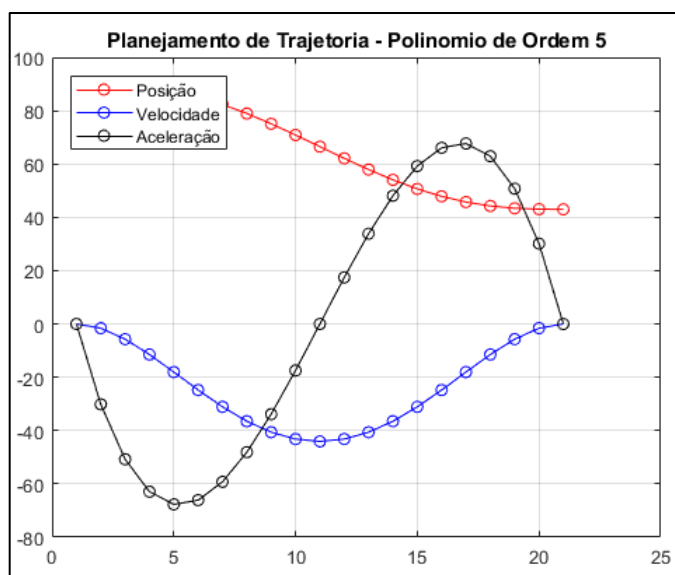
4.3 Planejamento de Trajetória

Através do polinômio de quinta ordem utilizado para o cálculo da trajetória foram geradas combinações de tempo de movimento (T_f) e tempo de discretização (T_d). A escolha da combinação entre T_f e T_d depende dos atuadores utilizados e duração da execução do sistema. As Figuras 30, 31 e 32 apresentam algumas das opções comparadas. Nos gráficos, o eixo das abscissas é representado pela iteração do movimento enquanto o eixo das ordenadas representa o ângulo do atuador.

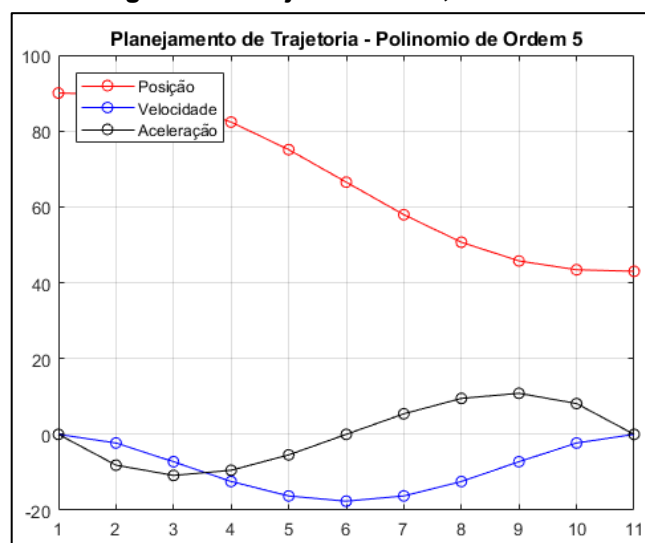
Na Figura 30 é utilizada uma combinação de $T_f = 2s$ e $T_d = 0,4s$, resultando em um movimento com aceleração alta e brusca.

Figura 30 - Trajetória $T_f = 2s$, $T_d = 0,4s$.**Fonte: Autoria Própria.**

Alterando o tempo de discretização (T_d) da Figura 31 de 0,5s para 0,1s obtemos um movimento com acelerações menos bruscas, apesar de ainda alcançar valores altos. A combinação de $T_f = 2s$ e $T_d = 0,1s$ é considerada para aplicação nos atuadores.

Figura 31 - Trajetória $T_f = 2$, $T_d = 0.1$.**Fonte: Autoria Própria.**

Uma das formas de reduzir os altos valores da aceleração dos atuadores é aumentar a duração do movimento (T_f). A Figura 32 demonstra a combinação $T_f = 5s$ e $T_d = 0,5s$, outra alternativa viável para ser aplicada no manipulador.

Figura 32 - Trajetória $T_f = 5$, $T_d = 0.5$.

Fonte: Autoria Própria.

6.4 Acionamento dos Atuadores

Para teste do funcionamento do sistema de acionamento dos atuadores foi utilizada combinação $T_f = 2s$ e $T_d = 0,4s$ para gerar a trajetória de um movimento completo. As Tabelas 7, 9 e 10 comparam a posição do atuador em relação a calculada para cada intervalo movimento.

Tabela 8: Testes Atuadores Trajeto Início-Objeto.

	Seguimento	Ang. Inicial	Trajeto Início-Objeto				
Simulado	Iteração	0	1	2	3	4	5
	Base (°)	90	87,22	74,76	57,24	44,78	42
	Braço (°)	90	86,35	70	47	30,65	27
	Cotovelo (°)	0	3	16,51	35,49	48,99	52
Atuador	Iteração	0	1	2	3	4	5
	Base (°)	90	86,4	75,6	57,6	45	41,40
	Braço (°)	90	86,35	70,2	46,8	30,65	27
	Cotovelo (°)	0	3,6	16,2	34,2	48,6	52,20
Erro	Iteração	0	1	2	3	4	5
	Base (°)	0	0,82	-0,84	-0,36	-0,22	0,60
	Braço (°)	0	0	-0,20	0,20	0	0
	Cotovelo (°)	0	-0,58	0,31	1,29	0,39	-0,20

Fonte: Autoria Própria.

Tabela 9: Testes Atuadores Trajeto Objeto-Destino.

	Seguimento	Ang. Inicial	Trajeto Objeto-Destino				
Simulado	Iteração	5	6	7	8	9	10
	Base (°)	42	49,99	85,81	136,19	172,01	180
	Braço (°)	27	30,65	47,00	70,00	86,35	90
	Cotovelo (°)	52	43,78	6,92	-44,92	-81,78	-90
Atuador	Iteração	5	6	7	8	9	10
	Base (°)	41,40	50,40	86,40	136,80	172,80	180
	Braço (°)	27	30,65	46,80	70,20	86,35	90
	Cotovelo (°)	52,20	43,20	7,20	-44,92	-82,60	-90
Erro	Iteração	5	6	7	8	9	10
	Base (°)	0,60	-0,41	-0,59	-0,61	-0,79	0
	Braço (°)	0	0,00	0,20	-0,20	0,00	0
	Cotovelo (°)	-0,20	0,58	-0,28	0,00	0,82	0

Fonte: Autoria Própria.**Tabela 10: Testes Atuadores Trajeto Destino-Início.**

	Seguimento	Ang. Inicial	Trajeto Destino-Inicio				
Simulado	Iteração	10	11	12	13	14	15
	Base (°)	180	174,79	151,43	118,57	95,21	90
	Braço (°)	90	90,00	90,00	90,00	90,00	90
	Cotovelo (°)	-90	-84,79	-61,43	-28,57	-5,21	0
Atuador	Iteração	10	11	12	13	14	15
	Base (°)	180	174,60	151,20	118,80	95,40	90
	Braço (°)	90	90,00	90,00	90,00	90,00	90
	Cotovelo (°)	-90	-84,6	-61,2	-28,8	-5,4	0
Erro	Iteração	10	11	12	13	14	15
	Base (°)	0	0,19	0,23	-0,23	-0,19	0
	Braço (°)	0	0,00	0,00	0,00	0,00	0
	Cotovelo (°)	0	0	0	0	0	0

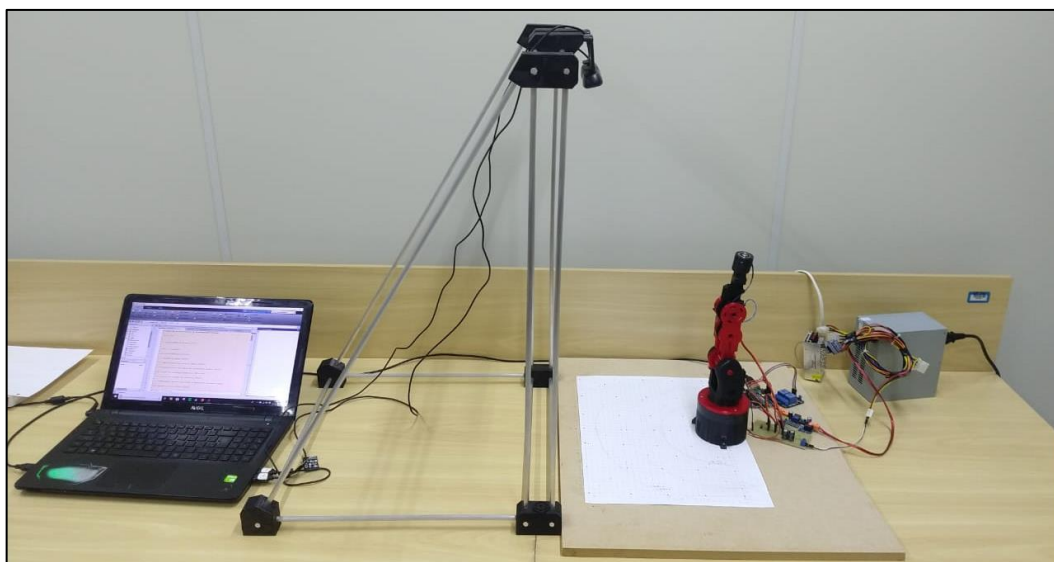
Fonte: Autoria Própria.

A partir da análise da diferença entre a posição real do atuador e a posição de referência, conclui-se que a precisão do movimento está dentro do padrão desejado, com erro médio de 0,19°.

4.4 Sistema Integrados Final

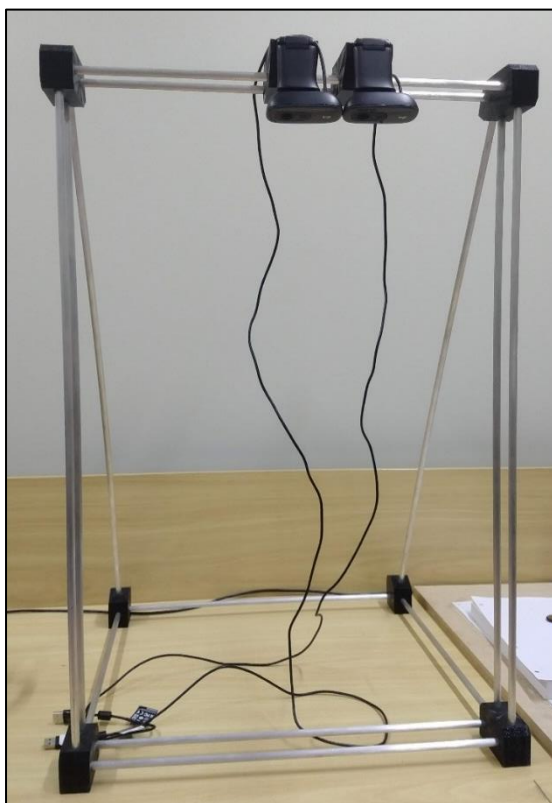
Após a validação de todos os subsistemas desenvolvidos foi realizada a etapa de integração. As Figuras 33, 34 e 35 exibem o protótipo montado.

Figura 33 – Protótipo desenvolvido.



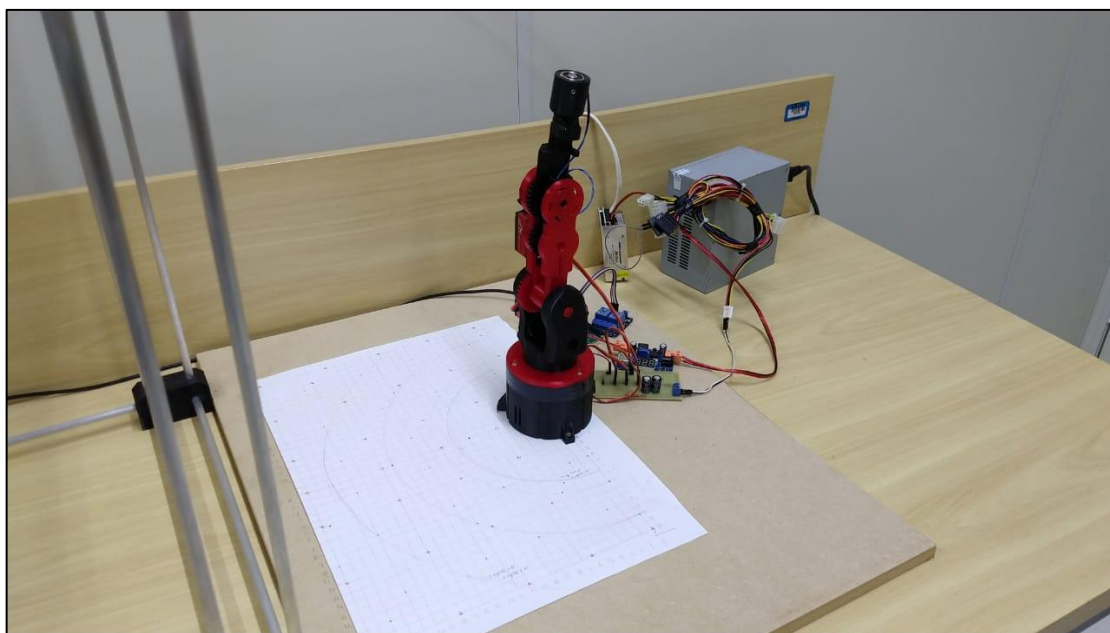
Fonte: Autoria Própria.

Figura 34 – Suporte de Câmeras.



Fonte: Autoria Própria.

Figura 35 – Área de trabalho do Manipulador robótico.



Fonte: Autoria Própria.

Com objetivo de testar o sistema completo em funcionamento foi posicionado um objeto de trabalho na superfície que simula uma esteira de transporte. Após a inicialização do sistema, o robô identifica a peça a ser manipulada e move-a da linha de transporte até a área de descarte. Uma vez descartado, o robô volta para a posição inicial e aguarda o comando para identificar a próxima peça e repetir o processo. A Figura 36 apresenta momentos chave do sistema funcionando.

Figura 36 – Sistema em funcionamento.



Fonte: Autoria Própria.

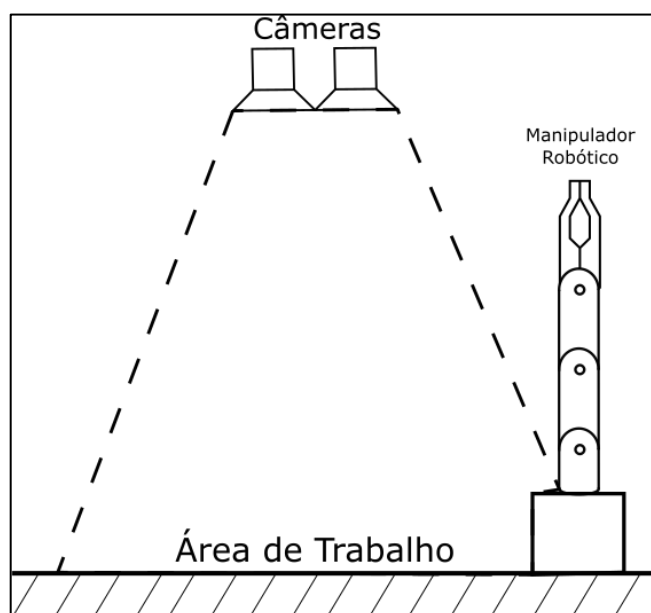
5 Discussão de Resultados

Com base nos resultados individuais dos subsistemas apresentados e da demonstração do sistema atuando em conjunto é possível afirmar a viabilidade da aplicação de um manipulador robótico autônomo capaz de identificar e remover peças de uma esteira de transporte entre outras possíveis aplicações industriais, por exemplo. É importante destacar que o protótipo desenvolvido apresenta erros de precisão que podem não ser aceitos em algumas aplicações industriais. Para melhores resultados, são sugeridas algumas alterações nos diversos subsistemas presentes.

5.1 Sistema de Visão Computacional

Devido ao posicionamento do conjunto de câmeras, o subsistema de identificação das coordenadas do objeto apresentou erro de paralaxe para deslocamento no eixo Z. Entre as possíveis formas de solucionar esse problema está o desenvolvimento de um novo suporte para o conjunto de câmeras, de forma que as lentes fiquem perpendiculares e centralizadas com a área de trabalho, como pode ser visto na Figura 37.

Figura 37 – Sugestão de posicionamento das câmeras.



Fonte: Autoria Própria.

Outra possível solução é a aplicação de um sistema inteligente para cálculo das coordenadas do objeto. A vantagem da aplicação de um sistema inteligente é a possibilidade de posicionar a câmera em qualquer posição, sem precisar considerar o erro de paralaxe gerado pelo posicionamento.

5.2 Sistema de Solução da Cinemática Inversa

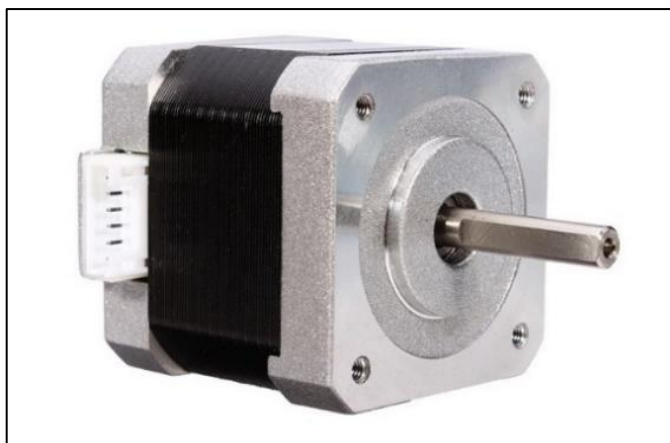
Apesar da solução desenvolvida nesse trabalho apresentar poucos erros na solução é possível ser realizado uma melhoria nos resultados através da implementação de novas soluções como diferentes topologias para a rede neural artificial ou até mesmo desenvolvimento de novos sistemas inteligentes.

5.3 Manipulador Robótico

Devido a suas características construtivas, os servo-motores são indicados para aplicações onde apenas o controle de posição é relevante. No caso de um manipulador robótico onde o controle das velocidades e acelerações são importantes, os servo-motores encontram dificuldades para atender as demandas.

Para possíveis melhorias no manipulador robótico é recomendado a utilização de atuadores do tipo motores de passo, ideais para controle de posição, velocidade e aceleração. A Figura 38 exibe um motor de passo comercial.

Figura 38 – Motor de Passo NEMA 17.



Fonte: www.filipflop.com.br.

Outra modificação é a utilização de um manipulador com 6 graus de liberdade, dessa forma é possível ter o controle não só da posição final do atuador, mas também da pose do mesmo. A Figura 39 exibe um manipulador robótico industrial de 6 GDL.

Figura 39 – Manipulador Robótico com 6 GDL.



Fonte: www.kuka.com.

6 CONCLUSÃO

Nesse trabalho foi realizada a montagem de um manipulador robótico autômato capaz de reconhecer um ponto no espaço e movê-lo para um espaço pré-determinado. Foram desenvolvidos três subsistemas principais para compor o projeto: Sistema de Visão Computacional, Sistema de Solução da Cinemática Inversa e Sistema de Acionamento dos Atuadores.

O sistema de visão computacional realiza a identificação do objeto de trabalho e calcula as coordenadas do objeto no espaço de trabalho. Nesse trabalho foi desenvolvido um algoritmo de identificação utilizando a transformada circular de Hough. Através desse sistema, são obtidas as coordenadas reais X, Y e Z do objeto.

A solução da cinemática inversa foi desenvolvida através da aplicação de Redes Neurais Artificiais do tipo *Perceptron* Multicamadas. A rede foi treinada utilizando dados da cinemática direta do robô, realizada através do Modelo de Denavit-Hartenberg.

No subsistema de acionamento dos atuadores é realizado o planejamento de trajetória a partir de um polinômio de quinta ordem. A trajetória definida é discretizada e os ângulos são enviados para os atuadores através da comunicação entre o software Matlab® e o microcontrolador Arduino Uno.

Os resultados obtidos apontam a validação dos subsistemas desenvolvidos, apresentando erros considerados baixos a nível de protótipo. Essa análise comprova a viabilidade de implementação do projeto em tamanho real.

Como trabalhos futuros, pode-se citar melhorias nos sistemas de visão computacional, novas alternativas para o sistema de solução da cinemática inversa e melhorias no manipulador robótico, como por exemplo, substituição dos servos motores por motores de passo ou substituição do manipulador robótico utilizado. Destaca-se também a possibilidade de implementação do sistema em maior escala, utilizando um manipulador robótico industrial.

REFERÊNCIAS

- AHMED, A. **Eletrônica de Potência**. 4. ed. São Paulo: Pearson Prentice Hall, 2000.
- 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.
- ARDUINO UNO SMD**. Disponível em: <<https://www.arduino.cc/en/Main/ArduinoBoardUnoSMD>>. Acesso em: 01 set. 2018.
- CORKE, P. **Robótica, Visão e Controle: Algoritmos Fundamentais no MATLAB**, Springer-Verlag, 2018
- CRAIG, J. J. **introduction to robotics mechanics and control**. 3ª edição, Prentice Hall, 1986.
- DALMEDICO, J. F.; MENDONÇA, M.; SOUZA, L. B.; BARROS, R. V. P. D.; CHRUN, I. R. **Artificial Neural Networks Applied in the Solution of the Inverse Kinematics Problem of a 3D Manipulator Arm**. 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brasil, 15 out. 2018.
- DENAVIT, J. **Description and displacement analysis of mechanics based on the 2x2 dual matrices**. 1 ed. University of Northwestern, Evanston, 1955.
- EVANS, M; NOBLE, J; HOCHENBAUM, J. **Arduino in Action**. Shelter Island: Manning, 2013. 370 p.
- FAN, X.; WANG, X.; XIAO, Y.; **Combined 2D-3D Vision System for Automatic Robot Picking**. In: International Conference on Advanced Mechatronic Systems, Kumamoto, Japan, 2014.
- GONZALEZ, R. C.; WOODS, R. E. **Processamento de Imagens Digitais**. Ed. Edgard Blücher, São Paulo, SP, 2000.
- GROOVER, M. P. et al. **Robótica: tecnologia e programação**. São Paulo: McGraw-Hill, 1989. 401 p.
- HAYKIN, S. **Neural Networks and Learning Machines**. 3rd ed. ed. [S.l.]: Pearson Education, 1999. 906 p.
- HEATH, J. **Pulse Width Modulation (PWM): What is it?**. 2017. Disponível em: <<https://www.analogictips.com/pulse-width-modulation-pwm/>>. Acesso em: 01 set. 2018.

HEXMOOR, H. **Essential Principles for Autonomous Robotics**. San Rafael, California: Morgan & Claypool Publisher, 2013. 157 p.

LESTRIANDOKO, N. H.; SADIKIN, R.; **Circle Detection Based on Hough Transform and Mexican Hat Filter**. In: International Conference on Computer, Control, Informatics and its Applications, 2016.

LOGITECH. **Webcam C270**. Disponível em: <<https://www.logitech.com/pt-br/product/hd-webcam-c270>> Acesso em: 20 out. 2019.

MATHWORKS. **Stereo Vision**. Disponível em: <<https://www.mathworks.com/discovery/stereo-vision.html>>. Acesso em: 20 out. 2019.

MCROBERTS, M. **Arduino básico**. São Paulo: Novatec Editora, 2011. 459 p.

MENDONÇA, M; SOUZA, L.B.; SOARES, P.P.; BARROS, R.V.D.; SANTOS, M.C.; **Visão Computacional e Redes Neurais Artificiais aplicadas à solução da cinemática inversa de um manipulador robótico**. In: VII Congresso Brasileiro de Engenharia de Produção, 2017, Ponta Grossa. Artigos Anais CONBREPRO 2017, 2017. p. 0-6

MOLZ, R. F. **Uma Metodologia para o Desenvolvimento de Aplicações de Visão Computacional utilizando um projeto conjunto de Hardware e Software**. 2001. 80 f. Tese (Doutorado em Ciência da Computação) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2001.

NIKU, S. B. **Introdução à robótica: análise, controle, aplicações**. 2ª edição, LTC, 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.

OH, J.; LEE, C.; **Development of a Stereo System for Industrial Robots**. In: International Conference on Control, Automation and Systems, 2007.

PARKER, J R. **Algorithms for processing Image and computer vision**. Indianapolis: Wiley Publishing Inc, 2011.

PEDRA, A. V. B. M.; MENDONÇA, M.; FINOCCHIO, M. A. F.; ARRUDA, L.V.R.; CASTANHO, J. E. C.; **Camera Calibration Using Detection and Neural Networks**. In: 11th IFAC Workshop on Intelligent Manufacturing Systems, 2013, 2013, São Paulo. Intelligent Manufacturing Systems, Volume# 11, 2013. v. 11. p. 1-6.

ROSÁRIO, J. M.; **INTRODUÇÃO A MECATRONICA**. São Paulo: Prentice Hall, 2005.

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.

SOUZA, L.B.; MENDONÇA, M.; DALMEDICO, J. F.; MONTEZUMA, M. A. F.; KONDO, H. S.; POCZE; **Inverse Kinematics and Trajectory Planning Analysis of a Robotic Manipulator**. In: International journal of advanced engineering research and science, v. 5, p. 207-214, 2018

ZHAO, S.; Li, B.; YUAN, J.; CUI, G.; **RESEARCH ON REMOTE METER AUTOMATIC READING BASED ON COMPUTER VISION**. In: IEEE/PES Transmission & Distribution Conference & Exposition: Asia and Pacific, p. 1-4, IEEE, 2005.