

**UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
UNIDADE ACADÊMICA DE GRADUAÇÃO
ENGENHARIA ELÉTRICA**

RÓGER THOMAS FONTOURA FRANÇA

**ESTUDO E CONSTRUÇÃO DE UM PROTÓTIPO DE ROBÔ DELTA COM
DETECÇÃO DE OBJETOS POR HAAR CASCADE**

**São Leopoldo
2020**

RÓGER THOMAS FONTOURA FRANÇA

**ESTUDO E CONSTRUÇÃO DE UM PROTÓTIPO DE ROBÔ DELTA COM
DETECÇÃO DE OBJETOS POR HAAR CASCADE**

Trabalho de Conclusão de Curso
apresentado como requisito parcial para
obtenção do título de Bacharel em
Engenharia Elétrica, pelo Curso de
Engenharia Elétrica da Universidade do
Vale do Rio dos Sinos - UNISINOS

Orientador: Prof. Ms. João Olegário de Oliveira de Souza

São Leopoldo

2020

Dedico este trabalho a minha mãe por toda força e garra inabalável que teve durante todos esses anos de graduação, para que eu pudesse realizar esse grande sonho.

AGRADECIMENTOS

Agradeço primeiramente a minha mãe que sempre se esforçou ao máximo para que eu pudesse chegar até aqui, sem ela eu jamais conseguiria traçar essa longa e árdua caminhada que é a graduação, também agradeço a minha namorada Karistiane Müller pela ampla compreensão que teve durante esses 8 anos, e seus pais que me acolheram e me trataram como um filho, e peço desculpas aos meus amigos por todos os eventos perdidos.

Agradeço também aos grandes mestres que tive na graduação pelo conhecimento passado e pelos seus altos níveis de cobrança, acredito que fui obrigado a me dedicar até a exaustão a determinados trabalhos e disciplinas, dando o máximo possível, e até mesmo me surpreendendo com a minha capacidade, também agradeço a um grande amigo Régis Cardoso por todas as horas e finais de semana que passamos juntos estudando, e por último, mas não menos importante, ao meu professor e orientador João Olegário de Oliveira de Souza por toda a dedicação e paciência em me auxiliar no desenvolvimento desse trabalho, e ao seu trabalho como coordenador do curso que muitas vezes passa despercebido aos olhos dos alunos.

RESUMO

Os processos industriais exigem cada vez mais precisão e robustez, a robótica industrial vem suprir essa necessidade conseguindo reduzir erros, aumentando a produtividade e facilitando o controle das variáveis de manufatura. Os robôs industriais substituem os humanos em tarefas simples como o empilhamento de caixas até tarefas complexas como solda e usinagem de precisão, a utilização de robôs industriais também contribui com a segurança, uma vez que eles podem ficar em ambientes hostis ou perigosos. A robótica industrial está integrada com a indústria 4.0 onde a flexibilidade na linha de produção é um item de suma importância, os *softwares* dos robôs podem ser atualizados de forma praticamente instantânea alterando as rotinas de trabalho e parâmetros.

Este trabalho aborda o estudo e construção de um protótipo de uma topologia clássica da robótica, um robô paralelo de Clavel de três graus de liberdade com detecção de objetos utilizando o algoritmo de *Haar Cascade*. Em seu conteúdo o trabalho apresenta as etapas de: construção mecânica, cinemática reversa, planejamento da trajetória, aquisição e detecção de objetos. Ao final será possível ver o protótipo construído juntamente com os resultados do teste de sistema de detecção de coordenadas onde foi possível extrair a coordenada da peça com êxito com um pequeno erro, também apresentará o resultado dos testes de *pick and place* onde o protótipo teve uma taxa de sucesso de 64%.

Palavras-chave: Robótica. Delta. Paralelo. Haar Cascade. Detecção.

LISTA DE FIGURAS

Figura 1 - Robôs soldadores na indústria automobilística	13
Figura 2 - Sistema de inspeção visual contribuindo no setor de qualidade	14
Figura 3 - Robô paralelo e robô série	16
Figura 4 - Interligação das tecnologias.....	17
Figura 5 - Diferentes tipos de robôs série	19
Figura 6 - Robô delta de Clavel.....	20
Figura 7 - Juntas Mecânicas	21
Figura 8 - Modelo de servo motor desmontado.....	22
Figura 9 - Diagrama simplificado de controle de um motor CC	22
Figura 10 - Sinal PWM de controle.....	23
Figura 11 - Sistemas de referência	24
Figura 12 - Espaço das juntas e espaço cartesiano.....	25
Figura 13 - Trajetória.....	27
Figura 14 - Imagem $f(x, y)$	28
Figura 15 - Efeito da amostragem	29
Figura 16 - Efeito da quantização.....	30
Figura 17 - Vizinhos de um pixel	32
Figura 18 - Exemplo de processo de realce de imagem	33
Figura 19 - <i>Features</i> aplicados a detecção de faces.....	34
Figura 20 - Árvore de decisões do algoritmo.....	35
Figura 21 - <i>Features</i>	35
Figura 22 - AdaBoost	36
Figura 23 - CPU e Microcontrolador	38
Figura 24 - Placa de desenvolvimento Stm32	39
Figura 25 - Microcomputador Raspberry pi 3 model B	40
Figura 26 - Volume de trabalho do robô delta desenvolvido por Santacatarina	42
Figura 27 - Montagem robô delta pelos graduandos da UTFPR	43
Figura 28 - Proposta Montagem pelos graduandos do IFSC	44
Figura 29 - Tela software desenvolvido.....	44
Figura 30 - Exemplo de Estrutura.....	47
Figura 31 - Servo motor 3325mg.....	48
Figura 32 - Tubos de fibra de carbono	49

Figura 33 - União articulada angular	49
Figura 34 - Base inferior	50
Figura 35 - Garra robótica com três dedos articulados	51
Figura 36 - Câmera Raspberry	51
Figura 37 - Placa de desenvolvimento BluePill	52
Figura 38 - Raspberry Pi 3 Model B+	52
Figura 39 - Conversor DC-DC	53
Figura 40 - Fluxograma do Processamento de detecção de peças	55
Figura 41 - Fluxograma robótica	57
Figura 42 - Placa utilizada	60
Figura 43 - Câmera e iluminação	60
Figura 44 - Base impressa em 3D.....	61
Figura 45 - Montagem do Protótipo.....	62
Figura 46 - Foto capturada da plataforma de trabalho do robô delta	63
Figura 47 - Peça padrão.....	64
Figura 48 - Diálogo de ajuste de origem	65
Figura 49 - Escolha coordenada de origem	65
Figura 50 - Sistema de calibração.....	66
Figura 51 - Sistema de calibração em execução.....	67
Figura 52 - Fluxograma de calibração.....	68
Figura 53 - Calculo de conversão de coordenada.....	69
Figura 54 - Simplificação do Robo Delta	72
Figura 55 - Fluxograma Conversão de coordenada	76
Figura 56 - Resultado do sistema de teste de detecção de peças	78
Figura 57 - Algoritmo para teste <i>pick and place</i> Fonte: Elaborado pelo Autor.....	80
Figura 58 - Teste de <i>pick and place</i> sendo executado	81

LISTA DE TABELAS

Tabela 1 - Número de bytes de armazenamento para vários valores de N e m	31
Tabela 2 - Resultados do teste do <i>pick and place</i>	82

LISTA DE SIGLAS

ABB	<i>Asea Brown Boveri</i>
ADC	<i>Analogue-Digital Converter</i> ou Conversor analógico digital
AGC	<i>Apollo Guidance Computer</i>
CA	Corrente Alternada
CAD	<i>Computer Aided Design</i> ou Desenho Assistido por computador
CC	Corrente Contínua
CMY	<i>Cyan, magenta and yellow</i> ou ciano, magenta e amarelo
CPU	<i>Central Process Unit</i> ou Unidade Central de Processamento
GDL	Graus de Liberdade
GPIO	<i>General Purpose Input Output</i> ou Entradas e saídas de propósito geral
HDMI	<i>High Definition Multimedia Interface</i> ou Interface multimídia de alta definição
MDF	<i>Medium Density Fiberboard</i> ou Fibra de média densidade
PCI	Placa de Circuito Impresso
PWM	<i>Pulse Width Modulation</i> ou Modulação por largura de pulso
RAM	<i>Random Access Memory</i> ou Memória de acesso aleatório
Rx	Sinal de recepção de comunicação serial
Tx	Sinal de transmissão de comunicação serial
USB	<i>Universal Serial Bus</i> ou Barramento Serial Universal

LISTA DE SÍMBOLOS

$^{\circ}$	Graus
i	Subscrito i , índice para referência as articulações
Θ_i	Ângulo do servomotor com índice i
$\Theta_i(t)$	Referente a trajetória do ângulo do servomotor
®	Marca Registrada
™	<i>Trademark</i>
X	Posição no eixo x no espaço cartesiano
Y	Posição no eixo y no espaço cartesiano
Z	Posição no eixo z no espaço cartesiano
X_{origem}	Coordenada x de origem sistema de visão
Y_{origem}	Coordenada y de origem sistema de visão
X_0	Coordenada x no instante inicial
Y_0	Coordenada y no instante inicial
Z_0	Coordenada z no instante inicial
X_f	Coordenada x no instante final
Y_f	Coordenada y no instante final
Z_f	Coordenada z no instante final
X_c	Coordenada x corrigida
Y_c	Coordenada y corrigida
Z_c	Coordenada z corrigida
T_0	Instante inicial
T_f	Instante final
x	Posição x da matriz da imagem
y	Posição y da matriz da imagem

SUMÁRIO

1 INTRODUÇÃO	12
1.1 Objetivos	15
1.1.1 Objetivo Geral	15
1.1.2 Objetivos Específicos	15
2 FUNDAMENTAÇÃO TEÓRICA	16
2.1 Robótica	16
2.1.1 Robô série	18
2.1.2 Robô paralelo	19
2.1.3 Juntas.....	20
2.1.4 Atuadores	21
2.1.4.1 Servo motores	21
2.1.5 Cinemática direta	23
2.1.6 Cinemática inversa	24
2.1.7 Dinâmica	26
2.1.8 Trajetória	26
2.2 Processamento digital de imagens.....	27
2.2.1 Imagem	28
2.2.2 Amostragem e Quantização	28
2.2.3 Armazenamento	30
2.2.4 Realce de imagens.....	31
2.2.5 Haar Cascade	33
2.3 Sistemas embarcados.....	36
2.3.1 Microcontroladores e Processadores	37
2.3.2 Microcomputadores	39
3 ESTADO DA ARTE	41
3.1 Trabalhos Acadêmicos	41
3.2 Robô Delta ABB FlexPicker®	45
3.3 Conclusões e Diferenciais	45
4. METODOLOGIA	47
4.1 Construção do protótipo	47
4.2 Sistema de visão computacional	53
4.3 Sistema de controle do robô delta.....	55

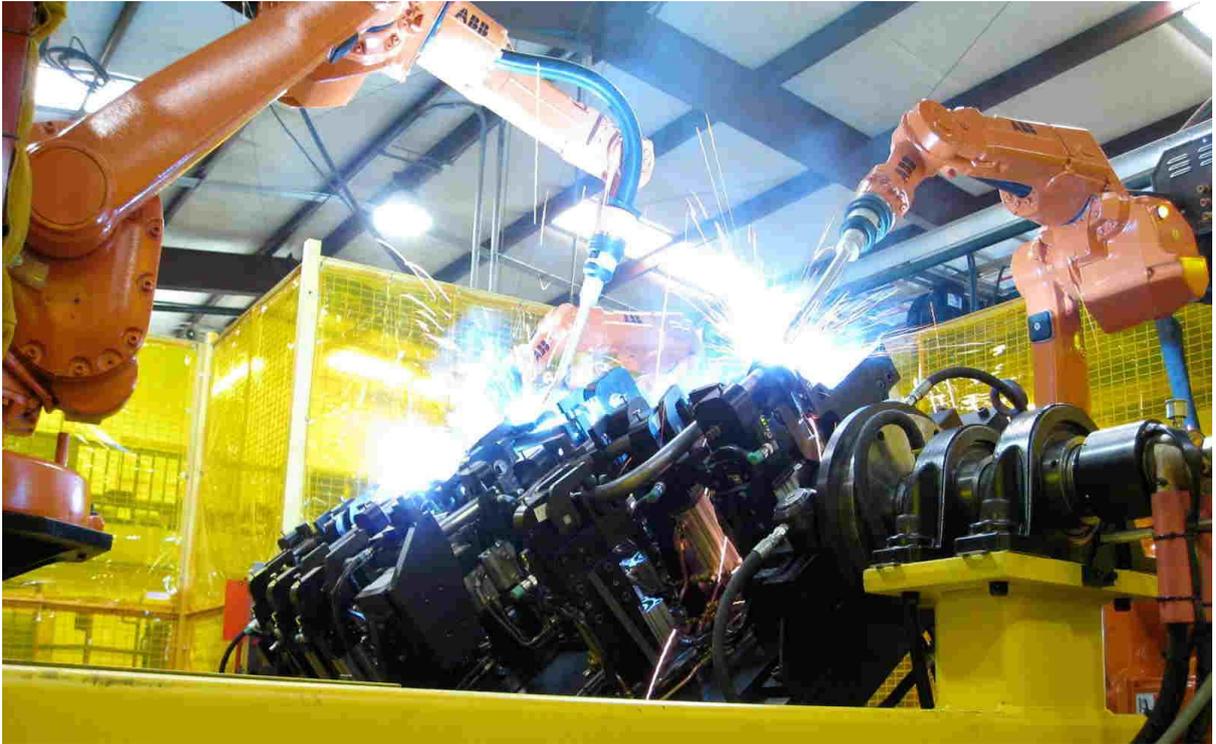
4.4 Métricas para avaliação de resultados	57
4.4.1 Erro absoluto de detecção.....	58
4.4.2 Erro absoluto de posição.....	58
4.4.3 Teste <i>Pick and Place</i>	58
5. DESENVOLVIMENTO	59
5.1. Hardware	59
5.2. Sistema de detecção de peças	62
5.2.1 Treinamento classificador de <i>Haar Cascade</i>	62
5.2.2 Ponto de origem	64
5.2.3 Calibração do sistema de detecção	65
5.2.4 Extração de coordenadas.....	69
5.3. Sistema Robótico	70
5.3.1 Cinemática Inversa.....	70
5.3.2 Planejamento da Trajetória	74
5.3.3 Conversão de coordenadas	75
6. ANÁLISE DOS RESULTADOS	77
6.1 Avaliação do sistema de detecção de peças	77
6.2 Avaliação do sistema robótico	78
6.3 Teste <i>Pick and Place</i>	79
7. CONCLUSÕES	83

1 INTRODUÇÃO

A robótica industrial não é um assunto novo, as primeiras publicações de robôs em universidades e laboratórios datam em torno da década de trinta, mas eles só vieram a aparecer com maior frequência no segmento industrial nos anos setenta, entretanto só grandes multinacionais tinham acesso a essa tecnologia que permitiam que eles produzissem material de ponta com mais qualidade em menos tempo desestabilizando a competitividade do mercado. (MOUSSA, 2011). Nos últimos anos médias e pequenas empresas estão conseguindo adquirir seus robôs indústrias, até mesmo por meio de aluguel, trazendo de volta este equilíbrio ao mercado. (ÉPOCA, 2017). Este fato abre a porta para startups que produzem robôs de baixo custo tenham um vasto mercado de trabalho, fazendo com que os engenheiros tenham que possuir conhecimentos sobre esta área.

A área da robótica é muito extensa atualmente existem sistemas robóticos para inúmeras finalidades, desde varrer o chão de lojas, até robôs subaquáticos capazes de realizar inspeção visual nas tubulações de petróleo. Dentro dessa vasta área há um segmento que se destaca, o segmento da robótica industrial, esses tipos de máquinas são aplicadas para as mais diversas finalidades como solda arco, solda ponto, pintura, montagem e manipulação, normalmente substituem o ser humano. Para cada aplicação existe uma topologia de manipulador apropriada para o serviço, uma das topologias é chamada de robô delta, ele apresenta maior flexibilidade, aumento da capacidade de carga uma vez que o peso é dividido em todos os atuadores, como todo o sistema apresenta desvantagens a topologia delta exigem um volume de espaço físico muito grande para cobrir uma pequena área de trabalho. (ANGELES, 1997). A Figura 1 exemplifica uma utilização comum no segmento automobilístico, robôs substituem o ser humano em uma tarefa perigosa como a solda. Quando a empresa faz essas alterações ela aumenta a qualidade do processo uma vez que o manipulador realiza um maior número de soldas com maior precisão e exatidão.

Figura 1 - Robôs soldadores na indústria automobilística



Fonte: Mecânica (2018).

Para realizar o processamento digital de imagens é necessária a execução de diversas etapas como aquisição, pré-processamento, filtragem, pós-processamento e reconhecimento de padrões. Diferentes áreas têm seu foco destinado em alguns desses itens, como a medicina que para a otimização de imagens de raios x, ultrassonografias e tomografias faz o uso de avançados filtros e técnicas de processamento, por sua vez, a inspeção visual tem seu foco nos reconhecimentos dos padrões. (GONZALEZ; WOODS, 2000). No segmento industrial está cada vez mais corriqueiro o uso de sistemas de inspeção visual para realizar o controle de qualidade de produtos, com a utilização desses sistemas em diversos setores da linha de produção é possível reduzir custos, prevendo produtos defeituosos e os eliminando das linhas de produção, sistemas de visão computacional também são utilizados na detecção de objetos. A Figura 2 apresenta um exemplo de sistema de inspeção visual utilizado no setor alimentício para realizar o controle de qualidade em garrafas de água, garantindo que todas as tampas estejam bem posicionadas e lacradas, a câmera captura as imagens e o computador é capaz de reconhecer determinados padrões, como os de garrafas sem tampa ou com lacres rompidos.

Figura 2 - Sistema de inspeção visual contribuindo no setor de qualidade



Fonte: Sensor (2018).

Essa monografia vem estruturada em seis capítulos para apresentar ao leitor um guia das etapas que foram realizadas para o estudo de construção do robô delta. O capítulo dois introduz diversos conceitos básicos para a compreensão do que é necessário para a confecção do dispositivo, esses conceitos abrangem a área da robótica, microcontroladores, e sistemas de visão computacional. O capítulo três apresenta um robô delta comercial, exemplificando o que existe atualmente disponível no mercado e alguns trabalhos acadêmicos sobre o mesmo tema, seguido do capítulo quatro que mostra a proposta desse trabalho através de esquemas e fluxogramas. O capítulo cinco apresenta as etapas para a execução do projeto, montagem, detecção de peças, sistema de controle do robô delta. O capítulo seis apresenta a análise dos resultados.

1.1 Objetivos

1.1.1 Objetivo Geral

Este trabalho tem como o objetivo a exploração do estudo da movimentação e construção de um robô paralelo tipo delta com três graus de liberdade (GDL), e o desenvolvimento de um sistema de visão computacional para identificação de um formato definido de peça. Foram realizados ensaios para avaliação do robô delta e do sistema de detecção de peças. Por fim o conjunto do sistema foi testado executando a operação de *pick and place* (pegar e largar). O protótipo possibilita o estudo e aplicação de uma grande variedade de projetos futuros nas áreas de robótica, processamento digital de imagens, inteligência artificial e controle todos podendo ser desenvolvidos em cima da plataforma.

1.1.2 Objetivos Específicos

- a) desenvolver um sistema de processamento digital de imagens capaz de reconhecer uma determinada peça padrão e obter as coordenadas para indicar ao sistema robótico;
- b) mapear o erro de detecção do sistema;
- c) construir um robô paralelo com diâmetro de atuação de 300 mm (milímetros);
- d) estudar e utilizar as equações de cinemática inversa e planejamento de trajetória desenvolvido por estudos anteriores de outros autores;
- e) mapear o erro de posicionamento do efetuador final;
- f) testar o conjunto detecção e manipulação na operação de *pick and place*;

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são introduzidos alguns conceitos de robótica, visão computacional, sistemas embarcados e motores, que são conhecimentos básicos para a execução deste trabalho.

2.1 Robótica

Com o passar dos anos diversos autores e organizações de robótica formularam suas definições do que seria um robô, porém existem alguns pontos onde todos convergem, um robô é um manipulador, programável, multifuncional, com três ou mais eixos, podendo ser móvel ou imóvel. O mesmo ocorre com a classificação dos robôs alguns autores os classificam pela função, pelo tipo de movimentação, pelo tipo de articulação, pela técnica de controle, pela mimetização do corpo humano, nas topologias dos robôs industriais essa classificação se restringe a praticamente dois tipos: paralelos como os da Figura 3a) ou séries como os da Figura 3b).

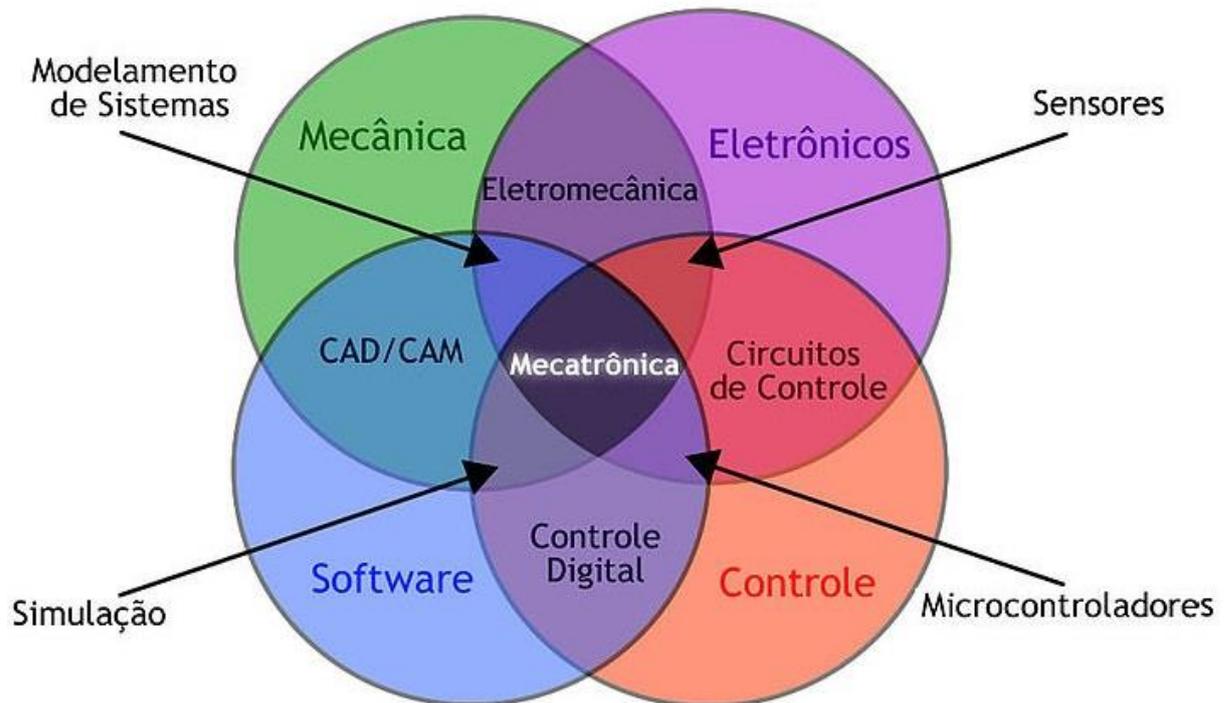
Figura 3 - Robô paralelo e robô série



Fonte: Adaptado de ABB (2018).

A robótica é uma área complexa da mecatrônica onde é necessário o uso de recursos de ponta de diversos setores tecnológicos, a Figura 4 mostra a interligação das tecnologias para o desenvolvimento da mecatrônica, logo após uma breve contribuição de cada área.

Figura 4 - Interligação das tecnologias



Fonte: Fabiones (2018).

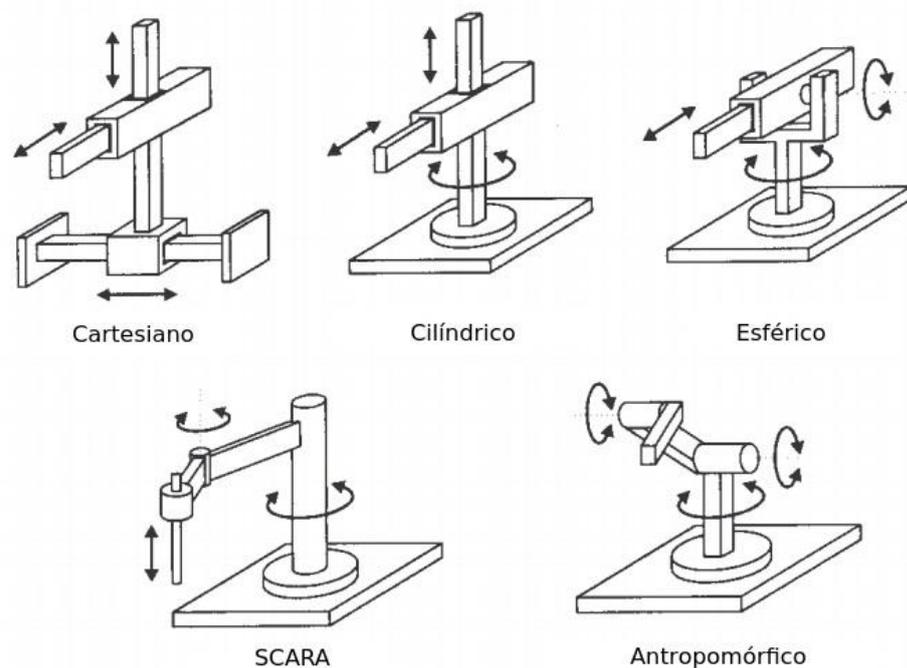
- a) mecânica para a análise da estrutura do sistema juntas, elos e equações de dinâmica e cinemática;
- b) eletrônicos para a construção de *drivers* de acionamentos dos motores;
- c) software para os desenvolvimentos dos sistemas embarcados;
- d) eletromecânica para o desenvolvimento de servo motores e motores de passo;
- e) sensores do tipo *encoders*, no caso particular da robótica os sistemas de visão computacional.

2.1.1 Robô série

Os robôs séries são os mais utilizados na indústria, são os mais simples dos sistemas robóticos, esse tipo de equipamento habitualmente recebe o nome de manipulador, que significa mover com as mãos, os robôs deste tipo no setor industrial tentam mimetizar o comportamento do braço humano. (ANGELES, 1997). O braço humano é composto de juntas muito complexas, então foram criados diferentes tipos de tecnologias para obter determinadas características do braço humano, resultando em diferentes tipos de *design* como mostra a Figura 5, cada um com sua devida aplicação. De acordo com Moussa (2011) as principais características dos manipuladores são:

- a) robô cartesiano é composto de três juntas deslizantes, possui alta rigidez e alta exatidão nos movimentos, apresenta um controle bem simples devido ao fato que cada atuador controla uma coordenada do efetuador, como desvantagem ele apresenta uma pequena área de trabalho;
- b) robô cilíndrico possui uma junta de rotação e duas deslizantes, o volume de trabalho é maior que o cartesiano, mas aumenta a complexibilidade do controle;
- c) robô esférico, possui duas articulações de revolução e uma deslizante, possui um volume de trabalho maior que os cilíndricos, porém maior grau de dificuldade ao realizar o controle;
- d) robô SCARA amplamente utilizado na indústria, ele assim como o esférico apresenta duas juntas de revolução e uma deslizante alterando apenas o ângulo da última junta deslizante, adaptado para problemas de montagem;
- e) robô antropomórfico muito utilizado pelas montadoras de veículo, modelo mais popular no segmento automobilístico apresenta três juntas de revolução, seu controle é considerado um dos mais difíceis.

Figura 5 - Diferentes tipos de robôs série



Fonte: Barrientos (2007).

Em resumo os robôs com juntas deslizantes apresentam maior rigidez mecânica e um controle mais simples, o controle se torna fácil devido ao fato que cada coordenada é estipulada pelo posicionamento de um motor, ao adicionar juntas de revolução o sistema fica mais complexo, porque ao alterar a posição de um motor altera mais de uma coordenada do atuador final.

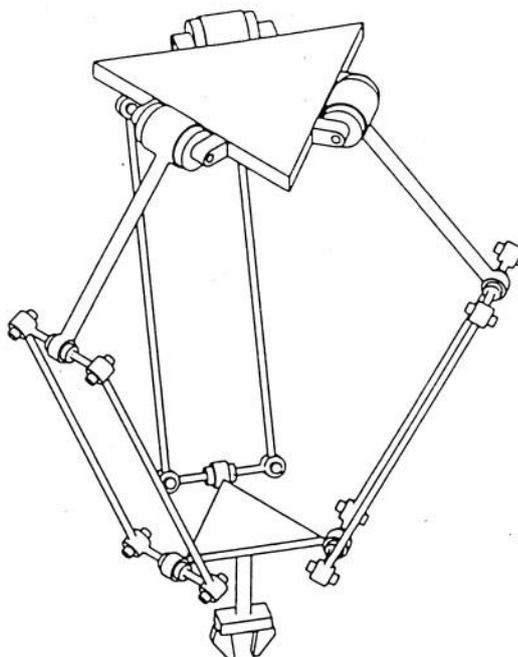
2.1.2 Robô paralelo

A topologia delta foi proposta por Raymond Clavel em 1980, as vantagens da topologia são a incrível velocidade e alta precisão no manipulador final, devido a isso, esse tipo de robô tem sido muito utilizado na função *pick and place* para empacotamento e organização de itens. (ProMotion, 2017). Porém tem uma grande desvantagem por causa da alta complexidade do equacionamento e controle do sistema.

O manipulador paralelo da Figura 6 é composto de uma base fixa e uma base móvel, interligadas por três ou mais manipuladores séries compostos de juntas esféricas e barras lineares. (ANGELES, 1997). A topologia de Clavel possui um dos

sistemas mais complexos de controle, devido a utilização das juntas esféricas na sua construção e por serem compostos por estruturas séries ligadas em paralelo, daí a origem do nome robô paralelo.

Figura 6 - Robô delta de Clavel



Fonte: Robotics (2014).

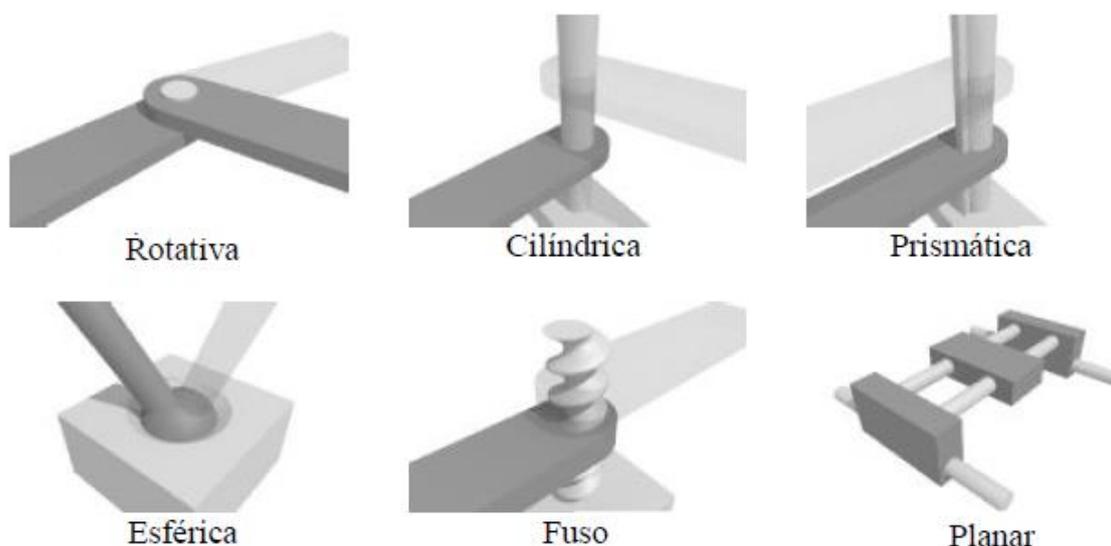
2.1.3 Juntas

As juntas mecânicas são utilizadas para unir dois ou mais objetos ou vínculos elas podem ser fixas, ou seja, não permitem o movimento como em tubulações hidráulicas e estruturas metálicas, ou podem ser móveis como as juntas homocinéticas de carros,

Alguns tipos de juntas mecânicas são mostradas na Figura 7, as juntas prismáticas e rotativas são comumente encontradas em manipuladores industriais. A junta prismática ou deslizante é composta por dois vínculos um interno e outro externo onde é possível que um vínculo deslize pelo outro. A junta rotativa ou de rotação também é composta por dois vínculos interligados por uma dobradiça que permite o movimento de revolução. A junta utilizada no robô paralelo são as juntas esféricas ou de bola e encaixe, são o tipo mais complexos de juntas e permitem

movimentos em 3 GDL, a utilização dessa peça é um dos motivos da grande complexibilidade da topologia delta. (MOUSSA, 2011).

Figura 7 - Juntas Mecânicas



Fonte: Adaptado de Gomes (2014).

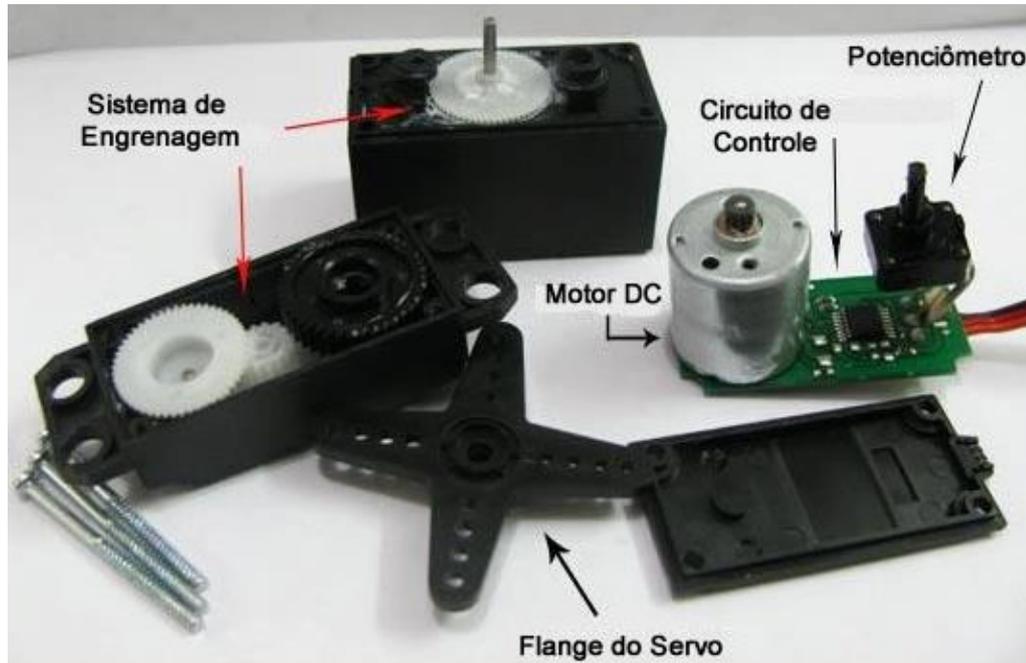
2.1.4 Atuadores

Os atuadores dos manipuladores industriais são considerados os músculos do sistema, normalmente são motores elétricos, mas também existem dispositivos que utilizam sistemas hidráulicos e pneumáticos. (NIKU, 2011).

2.1.4.1 Servo motores

Os servos motores são motores comuns CA ou CC, porém eles possuem um sensor de posição e uma unidade de controle dentro do seu invólucro. Normalmente são construídos com diâmetros menores e corpo alongado, através da realimentação do sensor é possível realizar o controle da velocidade e posição. (SILVEIRA, 2018). Nos servos motores industriais são utilizados *encoders* para realizar a medição da posição, mas existem servos menores como o da Figura 8, que são de menor custo que utilizam um potenciômetro para realizar essa medição.

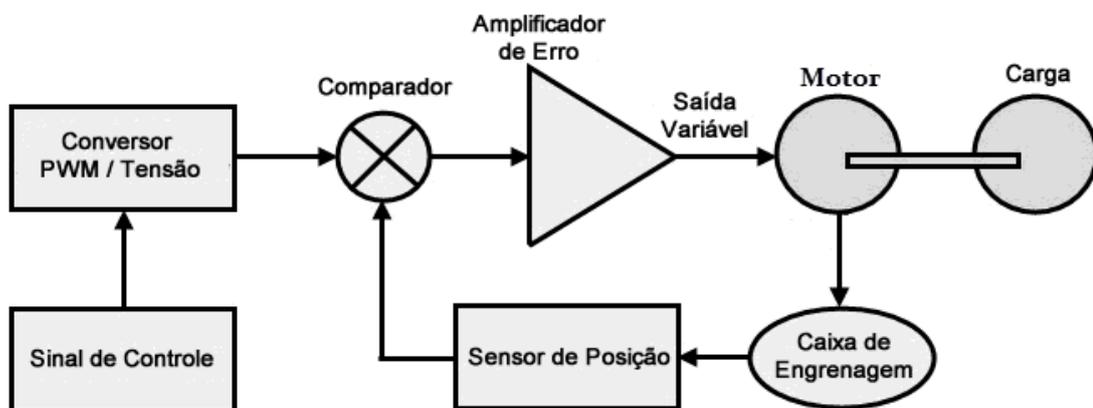
Figura 8 - Modelo de servo motor desmontado



Fonte: Adaptado de Silveira (2018).

Para realizar o controle do servo é necessário o envio de um sinal *Pulse Width Modulation* (PWM) ou modulação por largura de pulso, definido pelo fabricante, a Figura 9 apresenta um esquema do sistema de controle do servo motor, o PWM passa por um conversor que gera um sinal de referência, que é comparado com o sinal medido pelo potenciômetro, essa comparação resulta no erro do sistema, a placa controladora se encarrega de fazer essa correção.

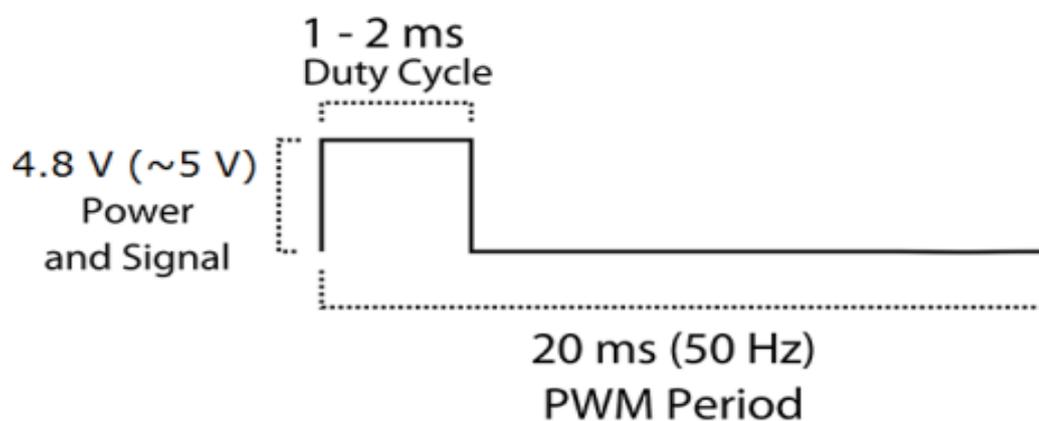
Figura 9 - Diagrama simplificado de controle de um motor CC



Fonte: Silveira (2018).

Para realizar o controle da posição dos servos motores é necessário a geração do sinal PWM da Figura 10, a frequência da portadora deve ser de 50 hertz (Hz) e a variação do nível logico alto de 1 a 2 milissegundos (ms), onde 1,5 ms é considerado o ângulo de 0 graus ($^{\circ}$), e 1 e 2 ms são respectivamente -90° e $+90^{\circ}$.

Figura 10 - Sinal PWM de controle



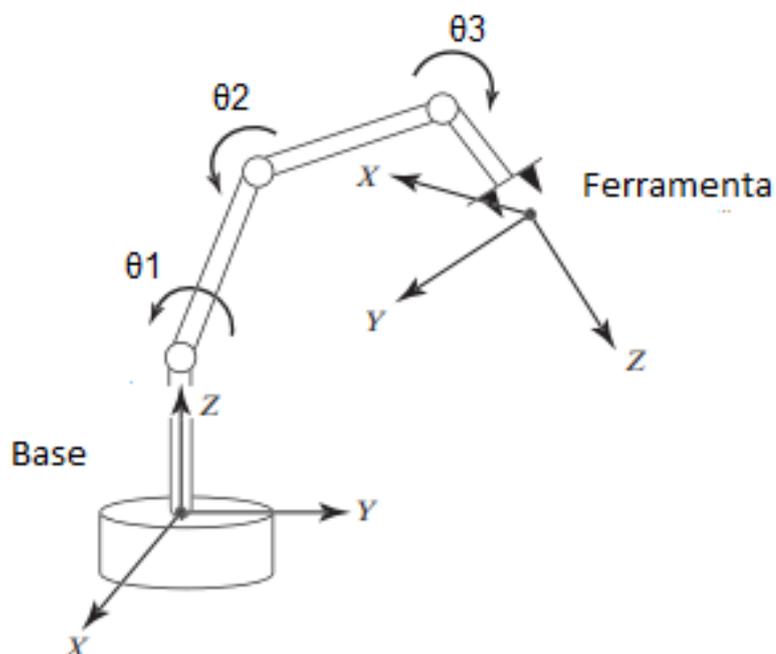
Fonte: SG90 (2018).

2.1.5 Cinemática direta

Cinemática é a ciência que estuda o movimento, considerando apenas posição, velocidade e aceleração desconsiderando as forças sobre o sistema, no caso dos manipuladores entra-se numa subárea referente a movimentação dos corpos rígidos. Como visto anteriormente um robô normalmente é composto de um conjunto de juntas e barras lineares que também recebem o nome de elos. O estudo da cinemática dos corpos rígidos consegue através de equações matemáticas descrever o movimento das partículas de um corpo ao longo do tempo. (CRAIG, 2012).

Os manipuladores possuem diversos sistemas de coordenadas, dentre eles dois sistemas recebem destaque, um deles é o sistema de coordenadas de referências onde o centro é a base fixa do dispositivo, e subsequente na outra ponta, temos o efetuador que também recebe seu sistema de coordenadas chamado de sistema de referência da ferramenta conforme Figura 11. (CRAIG, 2012).

Figura 11 - Sistemas de referência



Fonte: Adaptado de Craig (2012, p. 4).

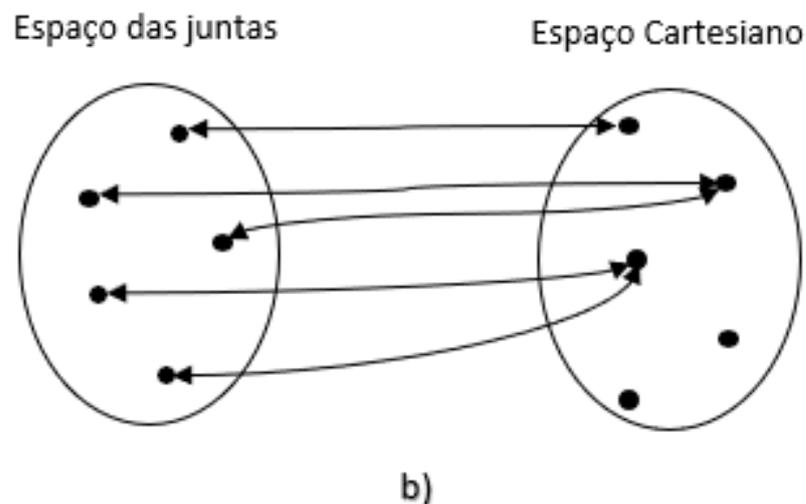
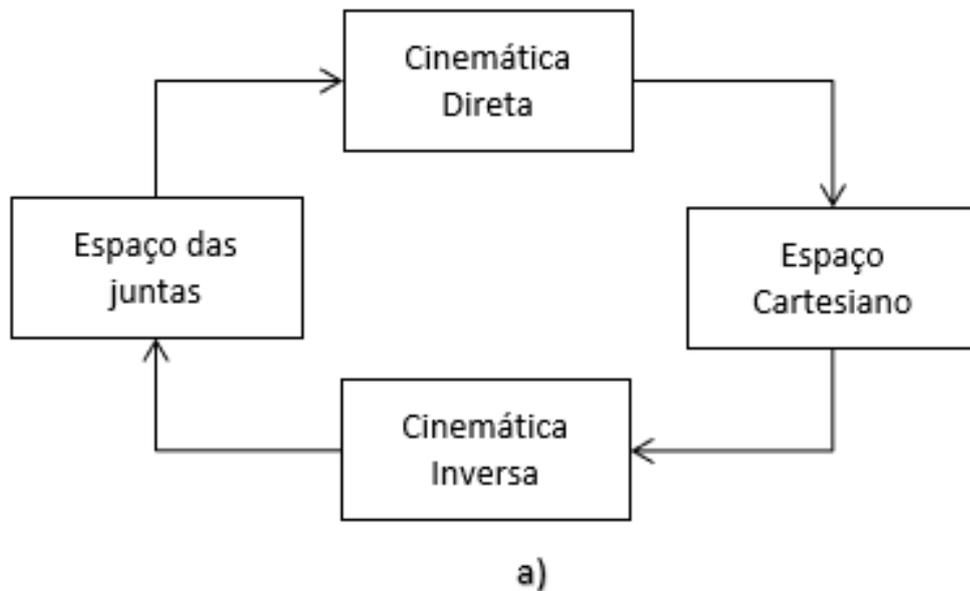
A cinemática direta é um problema da área da geometria onde temos a seguinte situação, conhecendo todos os ângulos (θ_1 , θ_2 , θ_3) e comprimento dos elos é possível determinar exatamente a posição e orientação do efetuador (X , Y , Z). (CRAIG, 2012).

2.1.6 Cinemática inversa

Na robótica a cinemática inversa é mais relevante comparado com a cinemática direta, uma vez que a primeira busca o processo inverso da outra, dado um determinado conjunto de coordenadas de ferramenta (X , Y , Z) determinar o conjunto de ângulos que as juntas (θ_1 , θ_2 , θ_3) devem se posicionar para obter o resultado desejado da ferramenta, esse problema pode se tornar extremamente complexo uma vez que o conjunto de equações do sistema é não linear e existem mais de um conjunto de ângulos de juntas que resultam na mesma posição de ferramenta. (CRAIG, 2012).

Para simplificar as representações das coordenadas de efetuador foram estipulados dois espaços, o espaço das juntas e o espaço cartesiano, onde é possível saltar de um para o outro utilizando a cinemática direta e inversa, como exemplificado na Figura 12a). A Figura 12b) reforça o conceito de que partindo de um determinado ponto no espaço das juntas utilizando a cinemática direta, ela nos leva a um ponto no espaço cartesiano lembrando que um conjunto de juntas diferentes pode levar ao mesmo ponto no espaço cartesiano, e utilizando a cinemática inversa saímos de um ponto do espaço cartesiano e ele pode nos levar a mais de um conjunto de juntas, porém queremos encontrar apenas um.

Figura 12 - Espaço das juntas e espaço cartesiano



2.1.7 Dinâmica

Dinâmica é a área da física que estuda o efeito da força aplicada a sistemas, no caso da robótica as equações de dinâmica costumam ser as únicas a sofrerem simplificações devido a sua alta complexibilidade. A dinâmica também é responsável pelo dimensionamento dos atuadores. Os atuadores devem ser dimensionados não somente para erguer o peso de um determinado objeto, mas é necessário conhecer a aceleração desejada do robô e os atuadores devem ser dimensionados para suportar a inércia do sistema, a inércia do sistema é a resistência a alteração do movimento. (NIKU, 2013). Se desejarmos fazer uma parada brusca numa carga em movimento o atuador deve ser capaz de se opor ao torque gerado por essa frenagem.

O uso das equações de dinâmicas em conjunto com as da cinemática torna possível a simulação dos sistemas robóticos, uma vez que aplicando torques na entrada do sistema se obtém o movimento em todo o tempo. (NIKU, 2013).

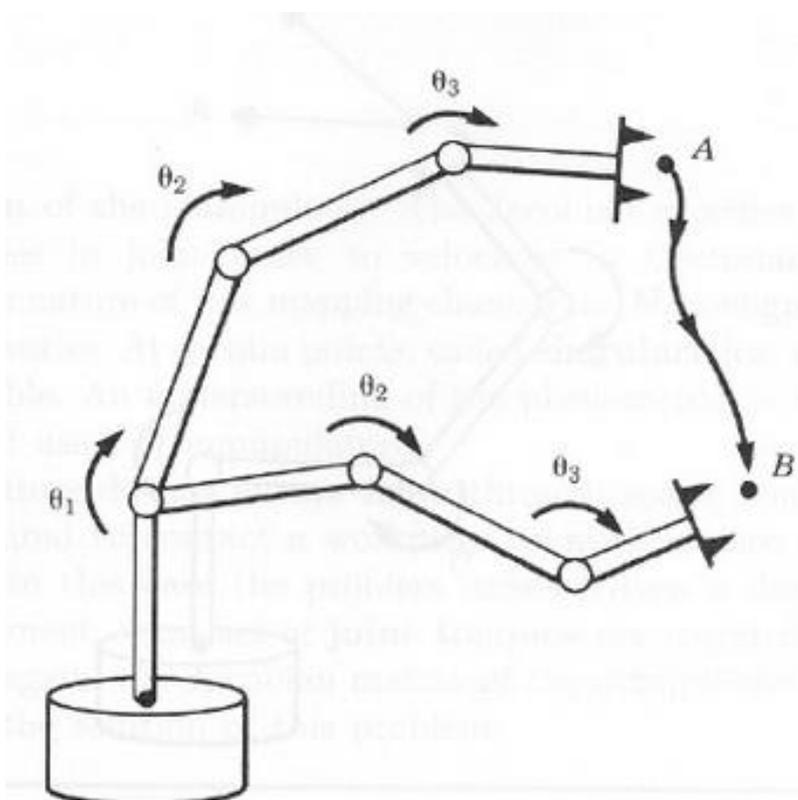
2.1.8 Trajetória

Uma vez resolvida a questão da cinemática inversa que diz que a partir de um conjunto de coordenadas é possível obter os ângulos de juntas necessários para posicionar o efetuador nas coordenadas desejadas, parte-se para a geração da trajetória, em geral deseja-se que todos os atuadores do sistema iniciem e alcancem seus objetivos simultaneamente, isso resulta num movimento coordenado. Para isso são definidos alguns pontos de passagem, ou até mesmo um caminho de referência que o efetuador deve seguir (CRAIG, 2012). Os algoritmos de geração de trajetória tentam resolver o problema: dado um conjunto de pontos no espaço (X_0, Y_0, Z_0) onde o efetuador se encontra no instante T_0 , qual a melhor trajetória para posicionarmos o efetuador nas coordenadas (X_f, Y_f, Z_f) , em um instante T_f .

A definição da trajetória fica ainda mais complexa quando existem restrições no espaço, por exemplo, obstáculos no caminho com que o manipulador não pode colidir. Esta área da robótica recebe um foco muito

especial porque a resolução do problema da trajetória está ligada diretamente a otimização do robô. A Figura 13 mostra o trajeto necessário para o efetuador se deslocar de um ponto A até um ponto B, por mais que o cérebro humano se depare com essa situação diversas vezes por dia, sua resolução computacional é muito complexa. (CRAIG, 2012).

Figura 13 - Trajetória



Fonte: Craig (2012, p. 10).

2.2 Processamento digital de imagens

O estudo de processamento digital de imagens se divide em duas grandes áreas, a otimização da qualidade das imagens muito utilizada pela área da saúde onde a imagem é condicionada para a percepção humana, na outra área a interpretação dos dados é feita pelo computador, na indústria temos os processos de seleção, montagem e inspeção de produtos. Um sistema de visão computacional é

composto por diversos elementos alguns deles são aquisição da imagem, armazenamento, processamento e detecção. (GONZALEZ; WOODS, 2000).

Nesta seção serão abordados assuntos básicos como o que é uma imagem digitalizada, amostragem e quantização, realce de imagens e o algoritmo de *Haar Cascades*.

2.2.1 Imagem

Uma imagem em seu formato digital é tratada como uma função bidimensional $f(x, y)$ que realiza o mapeamento da intensidade luminosa. A formação da imagem começa na emissão de uma fonte luz conhecida como $i(x, y)$, essa luz ao encontrar um anteparo, parte da luz é absorvida e parte é refletida $r(x, y)$, a função $f(x, y)$ é formada pelo produto das duas. Devido ao produto da função $i(x, y)$ conhecida como iluminação, e $r(x, y)$ conhecida como refletância, forma a função de intensidade $f(x, y)$, observa-se que ao alterar a intensidade da iluminação sem mudar o anteparo teremos um valor diferente de brilho para o mesmo objeto. (GONZALEZ; WOODS, 2000).

A Figura 14 mostra a função $f(x, y)$ que é uma matriz espacial e tem tamanho $N \times M$, onde cada elemento armazena a quantização discreta da intensidade.

Figura 14 - Imagem $f(x, y)$

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \dots & f(0, M-1) \\ f(1, 0) & f(1, 1) & \dots & f(1, M-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1, 0) & f(N-1, 1) & \dots & f(N-1, M-1) \end{bmatrix}$$

Fonte: Adaptado de Gonzalez e Woods (2000, p. 22)

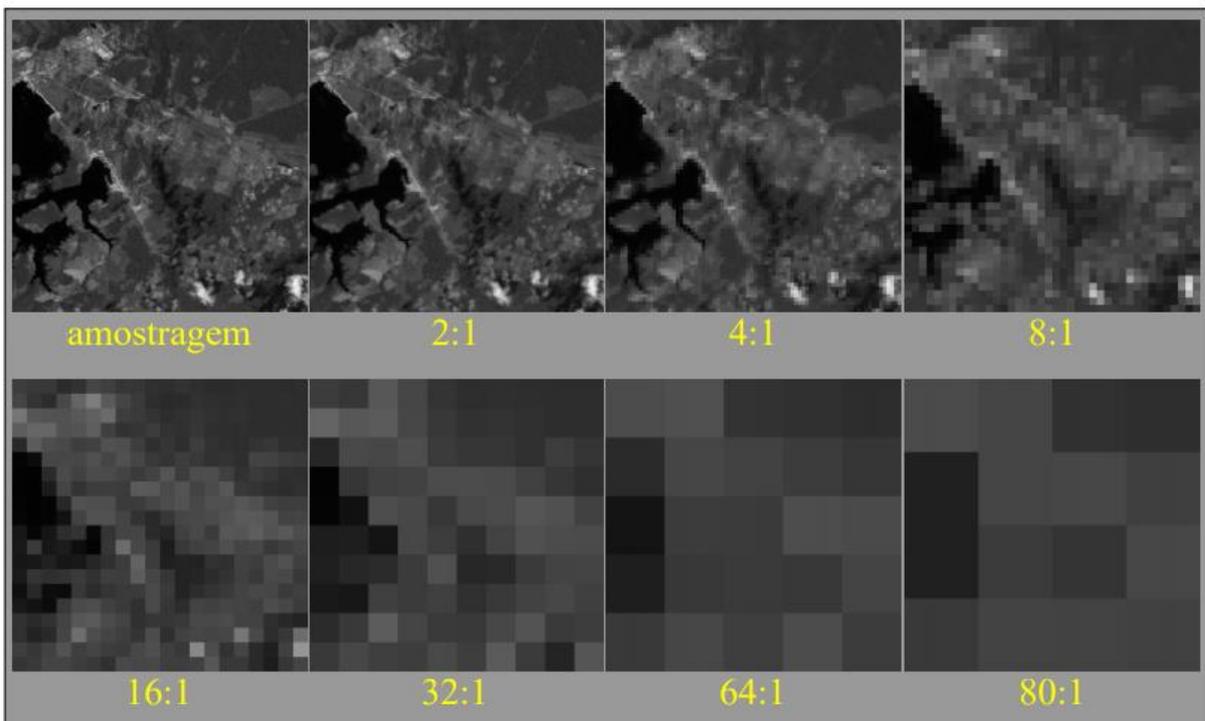
2.2.2 Amostragem e Quantização

A natureza das imagens é analógica, uma vez que são intensidades de luz e essa grandeza é analógica e contínua no tempo, para o processamento pelo computador é necessário que essa imagem seja digitalizada em dois níveis, em

coordenadas espaciais (x, y) essa digitalização recebe o nome de amostragem, e a digitalização da intensidade chama-se quantização. (GONZALEZ; WOODS, 2000).

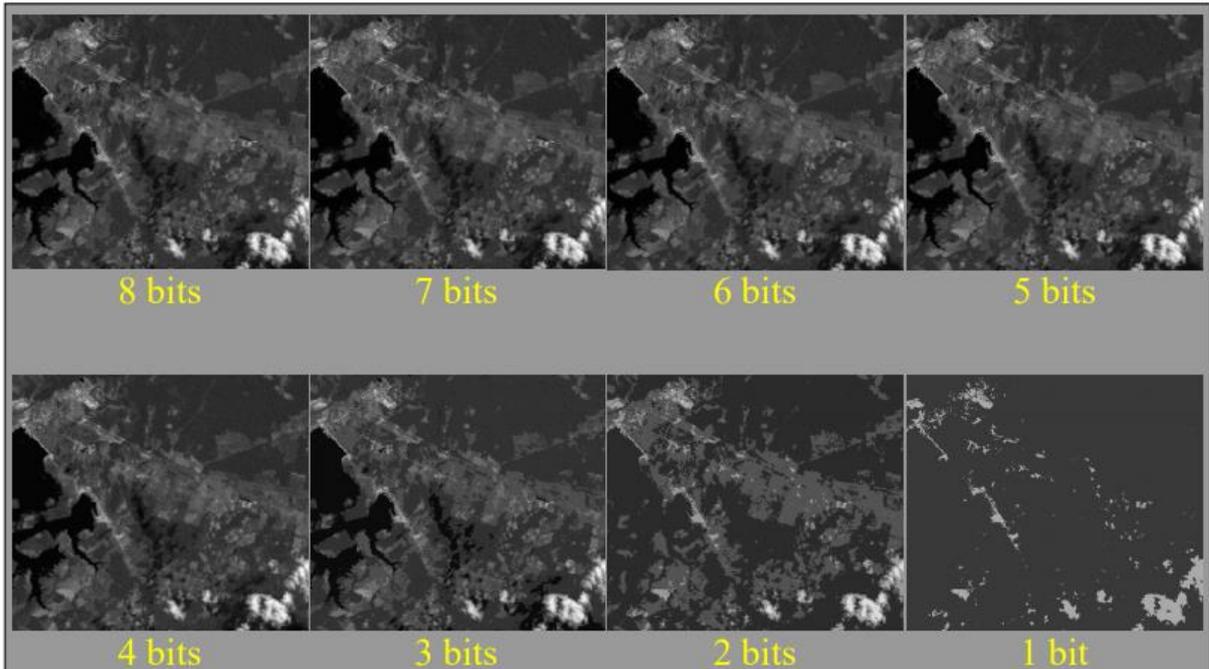
Para entendermos os efeitos que esses dois processos causam na formação da imagem temos a Figura 15 onde ocorre um processo de amostragem bidimensional, o primeiro quadro mostra o processo de amostragem de uma imagem e os posteriores apresentam uma redução na escala, mostrando que perdemos detalhes ao reduzir o número de *pixels* da imagem, por sua vez a Figura 16 apresenta o efeito da quantização, o primeiro quadro mostra uma quantização de 8 *bits* que resulta em 256 níveis de cinza diferente enquanto uma imagem de 1 *bit* possui apenas duas tonalidades de cores.

Figura 15 - Efeito da amostragem



Fonte: Adaptado de Santos (2009)

Figura 16 - Efeito da quantização



Fonte: Santos (2009)

2.2.3 Armazenamento

Antes de realizar qualquer tipo de processamento de imagem é necessário ter ela salva na memória do sistema embarcado, neste tópico teremos uma noção da quantidade de memória que uma imagem ocupa. É papel do engenheiro definir o tamanho da imagem que se deseja processar, sabendo que quanto maior a imagem maior será a capacidade de memória e de processamento que o sistema embarcado deverá possuir para realizar o tratamento dos dados.

Definimos como G o número de tons de cinza de uma imagem, calculamos como uma potência de dois, onde m é o número de *bits* necessários para formar a quantidade de níveis de cinza desejados. (GONZALEZ; WOODS, 2000).

$$G = 2^m$$

O número de *bits* necessário para armazenar uma imagem depende do produto do tamanho da amostragem da imagem N e M , e a quantidade de *bits* necessários para definir os tons de cinza desejados. (GONZALEZ; WOODS, 2000).

$$bits = N \times M \times m$$

A Tabela 1 apresenta dados exclusivos para imagens quadradas, ou seja, $N = M$, que compara a amostragem da imagem e o número de *bits* de quantização, com os bytes necessários para armazenar a imagem.

Tabela 1 - Número de bytes de armazenamento para vários valores de N e m

N = M	1	2	4	8
32	128	256	512	1.024
64	512	1.024	2.048	4.096
128	2.048	4.096	8.192	16.384
256	8.192	16.384	32.768	65.536
512	32.768	65.536	131.072	262.144
1.024	131.072	262.144	524.288	1.048.576

Fonte: Gonzalez e Woods (2000, p. 23)

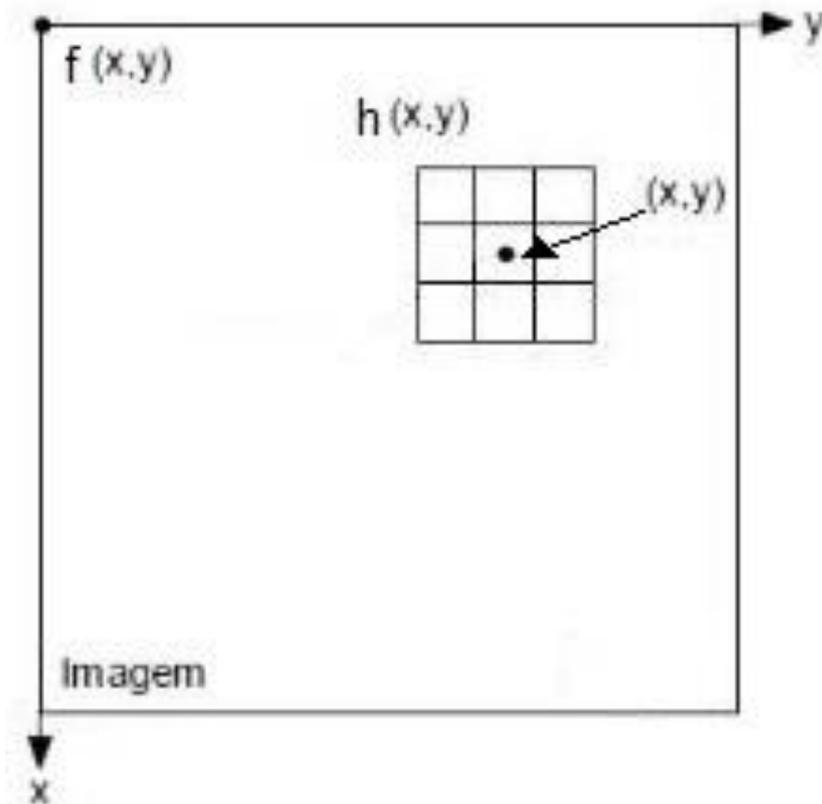
2.2.4 Realce de imagens

O objetivo de realçar uma imagem é facilitar o seu processamento mesmo que para isso seja necessário tornar a imagem indecifrável ao olho humano, esse processo de realce recebe o nome de pré-processamento devido ao fato de ainda não estarmos analisando a imagem de fato, e sim otimizando ela para facilitar as análises posteriores. Em processamento digital de imagens existem dois caminhos que podem ser seguidos para realizar o processamento, o método no domínio espacial que consiste na manipulação dos *pixels* da imagem da função $f(x, y)$, e outro chamado de método no domínio das frequências, onde é necessária a utilização de transformadas para a sua execução, normalmente são utilizadas as transformadas de Fourier bidimensionais, não se exclui o uso da combinação das duas técnicas. (GONZALEZ; WOODS, 2000).

Para realizar o processamento digital de imagens no domínio espacial as operações são realizadas diretamente nos *pixels* da imagem, para isso é escolhido um ponto (x, y) , após é selecionado uma região em sua vizinhança que recebe o

nome de $h(x, y)$ conforme Figura 17, $h(x, y)$ é uma subimagem que está contida na $f(x, y)$, neste subespaço h são realizadas operações matemáticas, e o resultado é salvo em uma função de saída $g(x, y)$, após o processo é reiniciado, ocorrendo o deslocamento do ponto inicial, e é realizado o mesmo procedimento, assim é feito até realizar a varredura de toda a imagem $f(x, y)$. (GONZALEZ; WOODS, 2000).

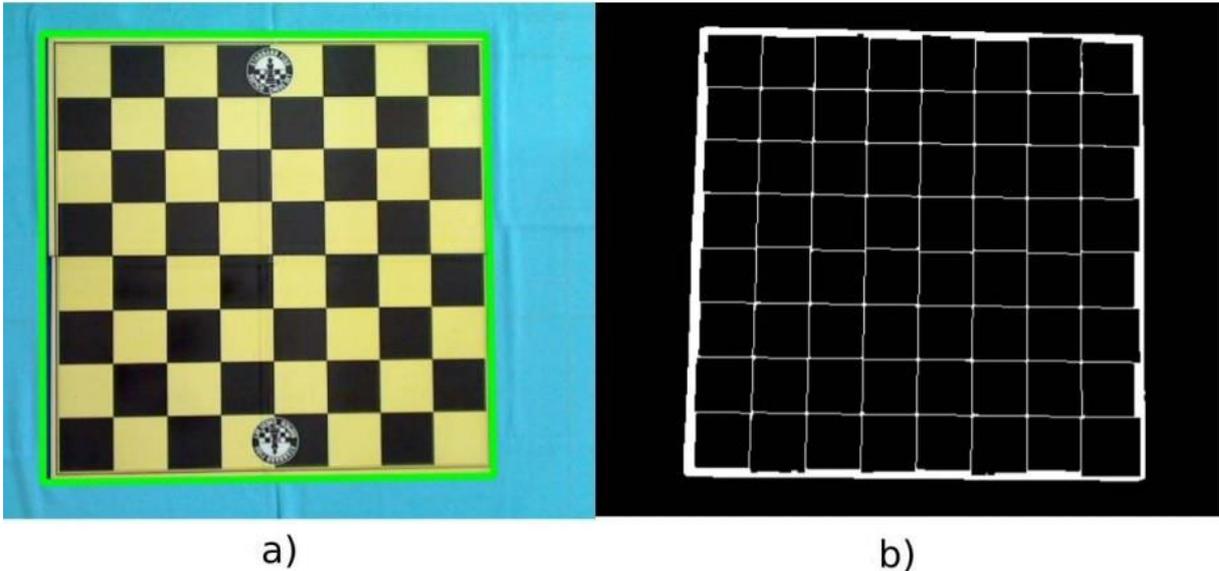
Figura 17 - Vizinhos de um pixel



Fonte: Adaptado de Gonzalez e Woods (2000, p. 116)

A Figura 18 mostra um exemplo de realce de imagens onde se desejava identificar as linhas que separam as cores amarelas e pretas do tabuleiro do lado a), embora para isso a imagem possa ter sido distorcida ao olho humano tornando difícil concluir que o lado b) é um tabuleiro de damas, para facilitar o reconhecimento de formas obteve-se êxito no realce desta imagem.

Figura 18 - Exemplo de processo de realce de imagem



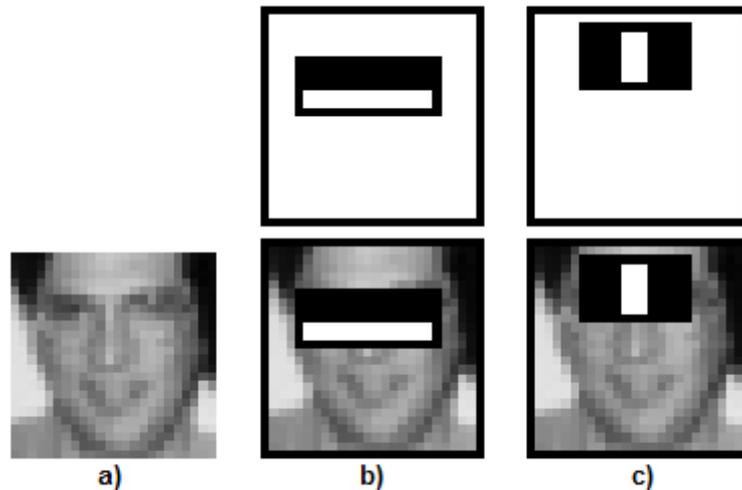
a)

b)

Fonte: Adaptado de Christie, Kusuna e Musa (2017, p. 5)

2.2.5 Haar Cascade

Haar Cascade é um algoritmo adaptado a partir dos estudos de detecção de objetos em imagens proposto por Paul Viola e Michael Jones (2001) que no caso foi um estudo para a detecção de faces. O conceito consiste na utilização de *features* como os da Figura 19 sobrepostos a imagem realizando comparações para obter características para identificação padrões. Na Figura 19 temos um exemplo de funcionamento do algoritmo na coluna Figura 19a) temos uma imagem original que se deseja realizar a detecção de face, na coluna Figura 19b) é possível perceber que a região das bochechas são mais claras que a região dos olhos onde foi aplicado o *feature* de borda. Na coluna Figura 19c) outro *feature* é definido pela região dos olhos ser mais escura que a região do nariz resultando em um *feature* de linha, se essas premissas forem verdade então essa imagem pode ser uma face. (VIOLA; JONES, 2001).

Figura 19 - *Features* aplicados a detecção de faces

Fonte: Viola, Jones (2001, p. 4).

No algoritmo são criados diversos estágios de *features* para identificar se a imagem contém o objeto desejado, para o algoritmo identificar se uma subimagem contida na imagem original possui o objeto desejado ele passa por diversos estágios, se todos forem verdadeiros o algoritmo aponta a subimagem de teste como o objeto desejado. Esta é uma característica que torna esse algoritmo tão propício para o uso em sistemas embarcados que possuem limitados recursos computacionais, o algoritmo não será executado por inteiro se nas etapas anteriores tiverem ocorrido reprovações. Na Figura 20 há um exemplo de árvore de decisões utilizadas no algoritmo de *Haar Cascade*, o sistema somente indicará o objeto como positivos se todos os estágios forem verdadeiros, caso contrário ele apontará que o objeto não existe na imagem. (VIOLA; JONES, 2001).

Figura 20 - Árvore de decisões do algoritmo

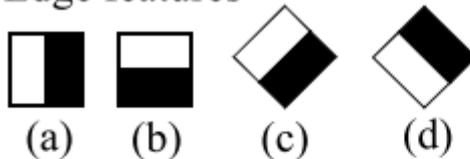


Fonte: Elaborado pelo autor, baseado em Viola, Jones (2001, p. 5).

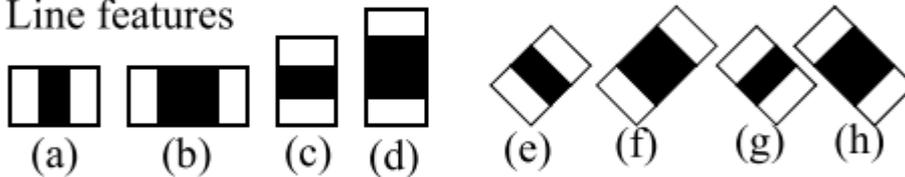
Em 2002 o algoritmo foi otimizado com a aplicação do conceito de *features* em diagonal como os da Figura 21.1.c) e Figura 21.1.d). Isso resultou no aumento das características que são capazes de se comparar, como consequência houve um acréscimo do tempo de treinamento e de detecção devido ao fato dos *features* serem um pouco mais complexos, porém aumentou a acurácia do algoritmo em aproximadamente 10%. (LEINHARD; MAYDT, 2002).

Figura 21 - *Features*

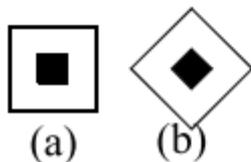
1. Edge features



2. Line features



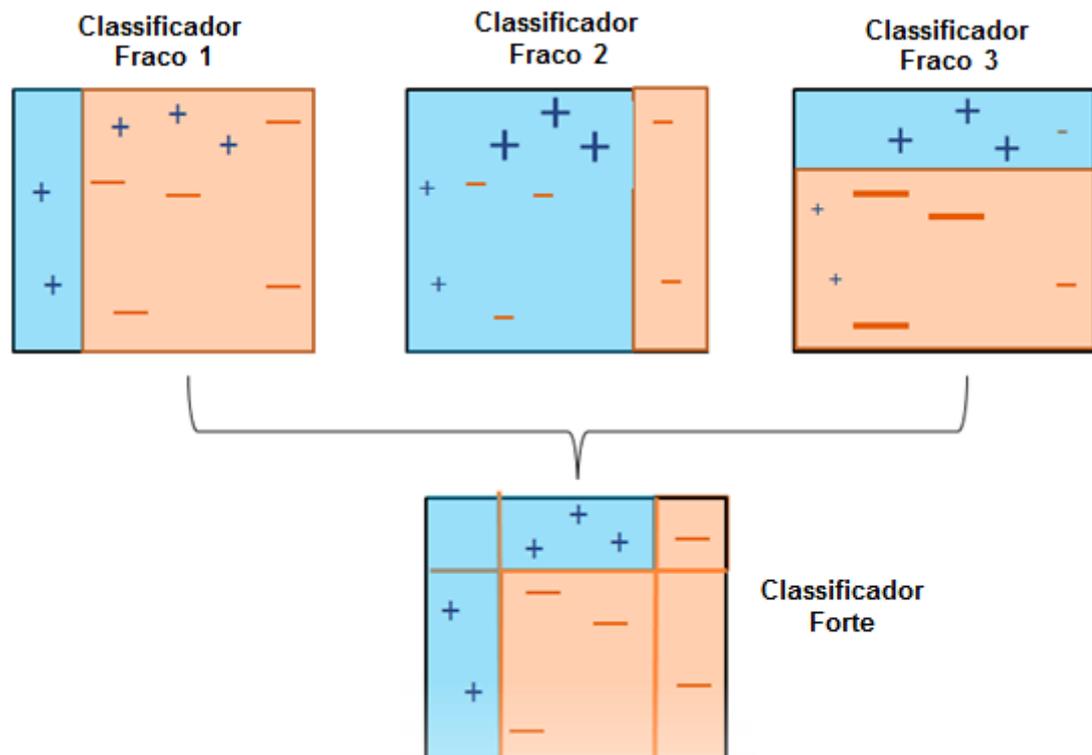
3. Center-surround features



Fonte: Adaptado de Lienhart, Maydt (2002, p. 901)

Outras características do *Haar Cascade* é a utilização do algoritmo de *AdaBoost* que permite a solução de problemas complexos. O algoritmo de *AdaBoost* de forma simplificada utiliza uma cascata de classificadores fracos para criar um classificador forte. Lembrando que isso é outra vantagem para o uso em sistemas embarcados porque os classificadores fracos possuem uma matemática simples enquanto os classificadores fortes requerem cálculos mais complexos, ou seja, maior tempo de processamento. A Figura 22 exemplifica o uso de *AdaBoost* que através de três classificadores fracos é possível criar um classificador forte. (DANGETI, 2017).

Figura 22 - AdaBoost



Fonte: Adaptado de Dangeti (2017).

2.3 Sistemas embarcados

Define-se como sistema embarcado um computador totalmente projetado para uma determinada finalidade, diferente dos computadores de uso geral. O

primeiro sistema considerado embarcado foi AGC (*Apollo Guidance Computer*) era responsável por todo o controle da espação nave apollo. (EMBARCADOS, 2013). Com o avanço da tecnologia atualmente os sistemas embarcados estão em todo o lugar, desde sistemas robóticos aos roteadores de nossas residências.

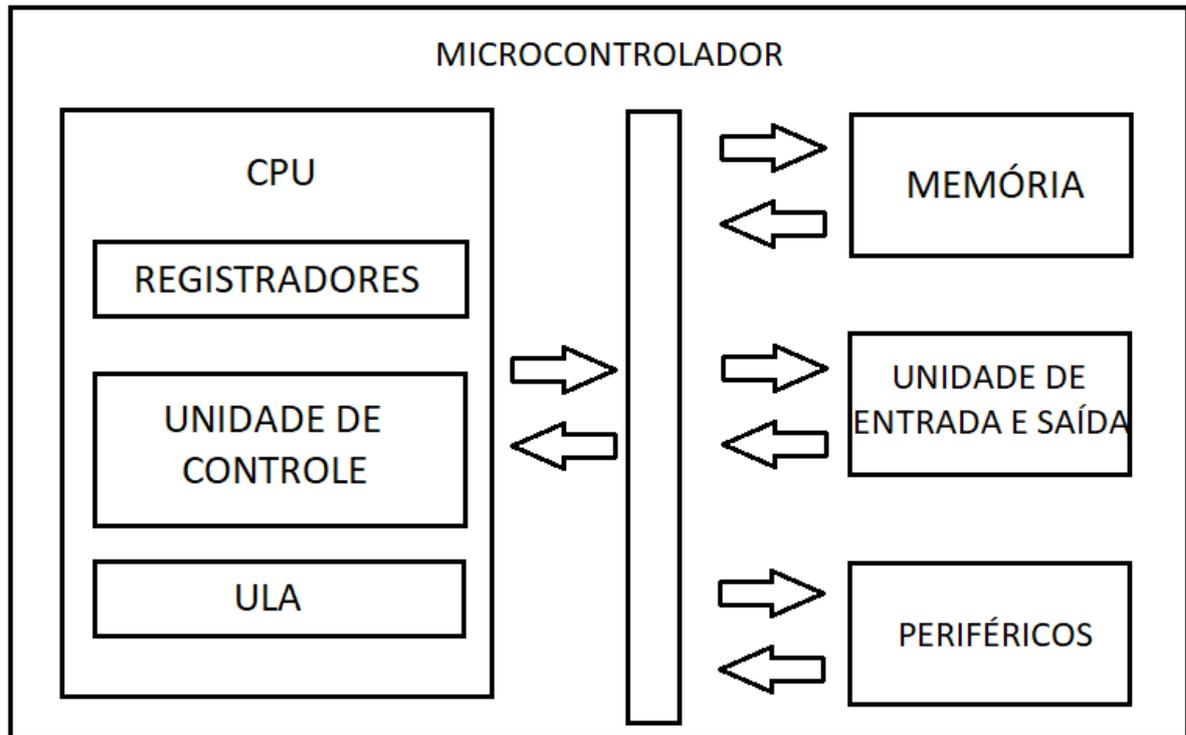
2.3.1 Microcontroladores e Processadores

Antes de definirmos a diferença de uma CPU (*Central Process Unit*) também chamado de processador e de um microcontrolador é necessário conhecer alguns componentes básicos, como unidade de controle, unidade lógica aritmética e registradores, conforme NERYS (2018):

- a) Unidade de controle é responsável pelo controle e sincronismo de todas as unidades internas da CPU;
- b) Unidade lógica aritmética realiza operações como adição, subtração e funções lógicas;
- c) Registradores são utilizados para o armazenamento de dados, alguns registradores têm funções específicas outros são de uso geral.

Um processador é responsável por executar um conjunto definido de instruções, ele contém uma Unidade Lógica Aritmética uma unidade de controle e um conjunto de registradores. Enquanto os microcontroladores possuem mais periféricos como memórias, portas de entrada e saídas, conversores, periféricos de comunicação serial e paralela, todos encapsulados em um único circuito integrado. (NERYS, 2018). A Figura 23 mostra uma CPU ou processador e suas unidades internas registradores, unidade de controle e unidade lógica aritmética, ao mesmo tempo esse processador faz parte de algo maior, o microcontrolador, que implementa memória, unidades de entrada e saída e diversos periféricos.

Figura 23 - CPU e Microcontrolador



Fonte: Adaptado de Nerys (2018)

Atualmente existem diversos microcontroladores comerciais, cada um possui um grupo de periféricos, cabe ao engenheiro escolher qual o *chip* se encaixa na sua aplicação. Os microcontroladores possuem diversos periféricos alguns deles são:

- a) *General Purpose Input Output* (GPIO) ou Entradas e saídas de uso geral – basicamente são portas de entradas e saída de dados;
- b) *Digital Analogue Converter* (ADC) ou Conversor analógico digital – responsável pela conversão de grandezas analógicas em digitais, o mundo que conhecemos é contínuo e analógico para o processamento pelo microcontrolador as grandezas devem estar discretizadas e quantizadas;
- c) *Counters/Timers* ou Contadores/Temporizadores – Dispositivos capaz de realizar contagem, utilizado para a contagem de tempo, sincronização e PWM.

A gravação do programa e confecção da placa de circuito impresso (PCI) era um trabalho árduo e demorado isso dificultava o aprendizado dos técnicos e engenheiros e atrasava a indústria na área de prototipagem e teste, uma solução para o problema foi desenvolver uma placa que já possui os componentes necessários para a utilização do circuito integrado.

Um exemplo de placa de desenvolvimento está na Figura 24, ela é conhecida como *BluePill* e contém um microcontrolador ARM M3 o *Stm32f103c8t6*. Essa placa contém o microcontrolador, dois cristais osciladores, um botão de *reset*, uma interface USB *Universal Serial Bus*, uma interface de programação, e todos os componentes básicos para o funcionamento básico do microcontrolador.

Figura 24 - Placa de desenvolvimento Stm32



Fonte: Filipeflop (2020)

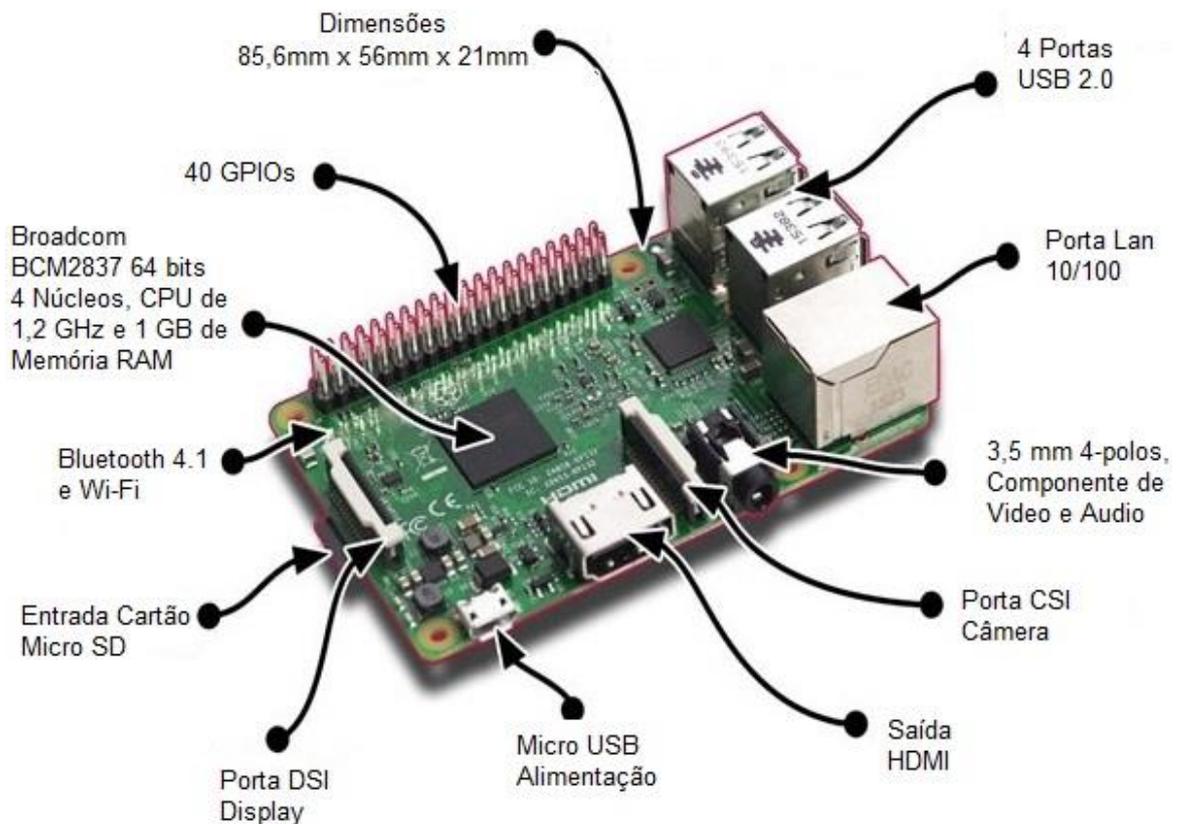
2.3.2 Microcomputadores

Os sistemas microcontrolados mesmo os de alta tecnologia tem suas limitações, nesse momento entram os microcomputadores onde é possível identificar de forma separada os componentes, atualmente existem diversas organizações e empresas produzem os microcomputadores como *Raspberry*, *Beaglebone*, *Cubieboard*, *NXP* e a gigante *Intel*.

A fundação *Raspberry* desenvolve placas de desenvolvimento de alta capacidade como a *Raspberry PI 3 Model B+*, contendo um processador *Quad*

Core 64 Bits com 1,2 GHz, 1 GB de memória RAM (*Random Access Memory*), 40 GPIO, saída HDMI (*High-Definition Multimedia Interface*), conexões para câmeras. A Figura 25 mostra a placa e um guia da localização dos seus componentes. Esta placa utiliza um sistema operacional baseado no *debian*, chamado de Raspbian, permitindo diversos recursos que um computador pessoal possui, como acesso a internet e *softwares* que rodam em cima dessas plataformas.

Figura 25 - Microcomputador Raspberry pi 3 model B



Fonte: Adaptado de Bynery (2018)

3 ESTADO DA ARTE

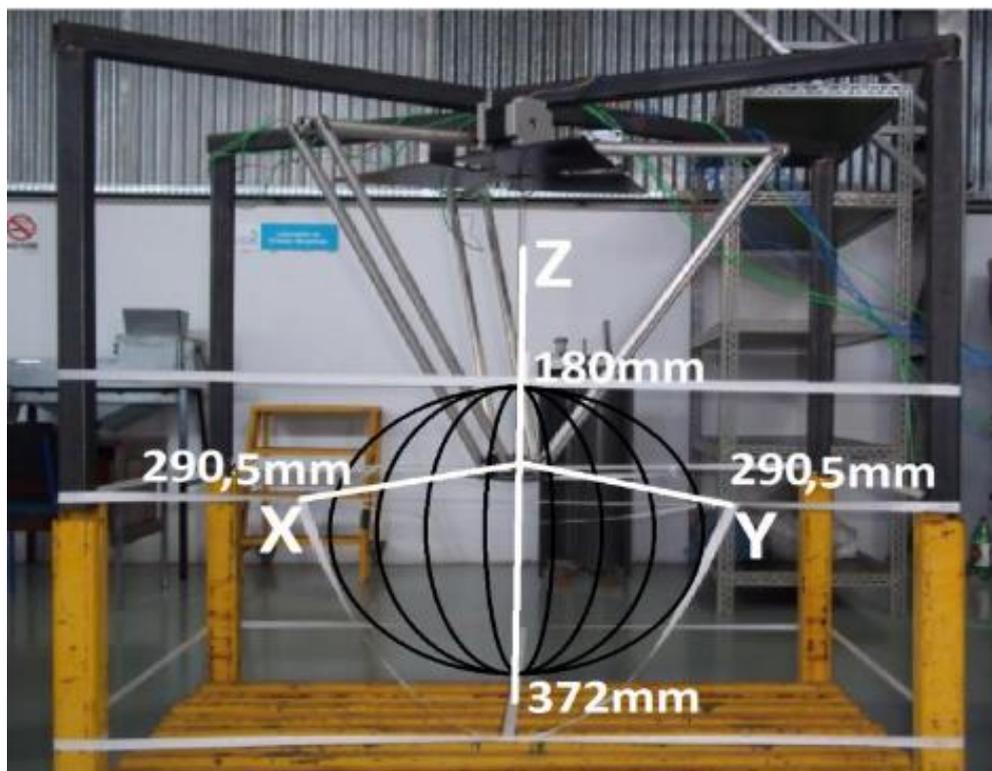
O estado da arte é uma definição de como se encontra um produto ou técnica atualmente, ao desenvolver um trabalho é necessário revisar o que já foi desenvolvido e o que é usado hoje em dia. Neste tópico veremos alguns trabalhos acadêmicos e um robô delta comercial, o objetivo desta etapa é levantar observações, características e técnicas usadas pelos autores ou empresas, também analisar as sugestões de melhorias propostas nos trabalhos para a construção do protótipo, evitando cometer os mesmos erros.

3.1 Trabalhos Acadêmicos

A construção de um robô delta não é algo atual, os trabalhos acadêmicos e publicações já estão no ramo de otimização das subáreas, ou seja, não existem trabalhos atuais sobre a construção do robô, mas de técnicas de otimização utilizando inteligência artificial no controle, na otimização da trajetória e obtenção dos ângulos das juntas da cinemática reversa, modelos de simulação de sistemas robóticos, avançadas técnicas de reconhecimento de padrões em imagens.

O trabalho de conclusão do Moises Augusto Santacatarina (2015) da Faculdade de Horizontina (FAHOR) apresentou um grande foco na análise em torno do volume de trabalho, dimensionamento das cargas, dimensionamento das hastes e do projeto mecânico das peças em *Computer Aided Design (CAD)*. Foram realizados alguns testes de força e deslocamento, porém os resultados não foram apresentados de uma forma clara. A Figura 26 mostra um exemplo da análise do volume de trabalho máximo alcançado pelo robô delta, mostrando que o alcance forma um elipsoide de revolução.

Figura 26 - Volume de trabalho do robô delta desenvolvido por Santacatarina



Fonte: Santacatarina (2015, p.36)

Os graduandos Felipe dos Santos Lazzari, Henrique Ribeiro de Oliveira e Teilor Sabino B. de Q. Paixão (2012) da Universidade Tecnológica Federal do Paraná (UTFPR), encontraram grandes problemas na escolha do microcontrolador do projeto, devido a sua capacidade reduzida de processamento e velocidade de comunicação, também tiveram problemas com as junta esféricas, a junta escolhida tinha um ângulo de pivotação reduzido, que resultou em uma menor área de atuação do efetuador. De positivo eles realizaram uma curva de aceleração suave na movimentação do robô que resultou em menores choques mecânicos, o trabalho foi desenvolvido inteiramente utilizando o *software* Labview™, que já possui um *package* para sistemas de visão computacional o *Vision Assistant*. A validação foi realizada colocando uma ficha de pôquer aleatoriamente no campo de visão da câmera, o robô a reconhecia e a colocava em cima de pontos pintados em uma folha de papel, como pode ser visto na Figura 27.

Figura 27 - Montagem robô delta pelos graduandos da UTFPR

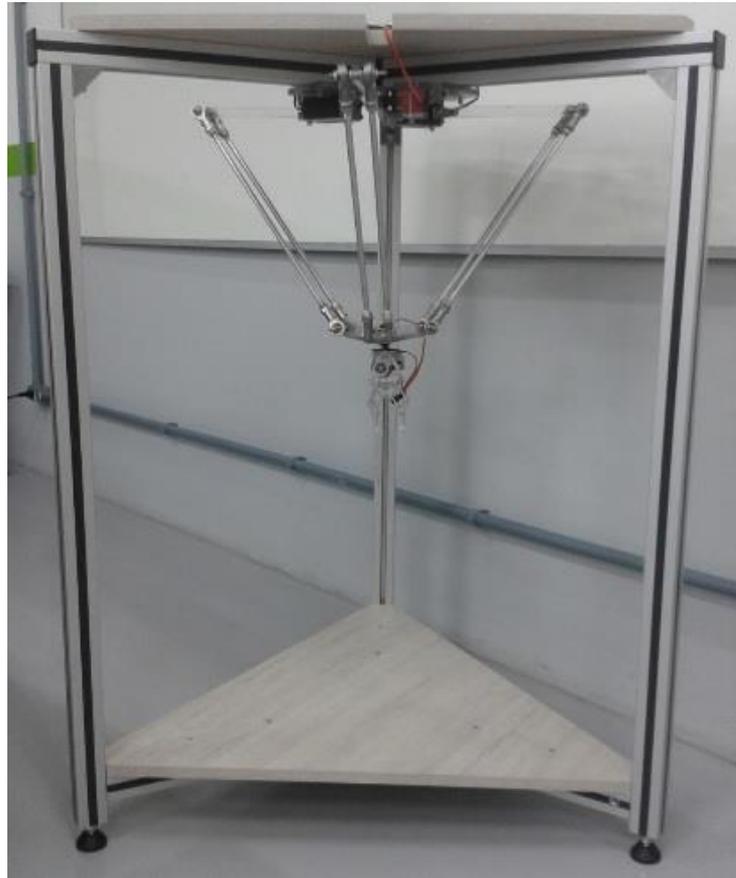


Fonte: Lazzari, Oliver e Paixão (2012, p.30)

A proposta de montagem do trabalho de conclusão dos graduandos Fabio Junior Jubelli, Marcelo de Lima e Robson Luiz Hortsmann (2013) do Instituto Federal de Ciência, Educação e Tecnologia (IFSC), foi o mais completo ele apresentou o desenvolvimento do cálculo das equações de cinemática direta e inversa, uma análise na seleção dos materiais para a confecção do robô, projetos em CAD, o trabalho foi todo estruturado com o auxílio do *software* MatLab. A Figura 28 mostra o protótipo construído pela equipe.

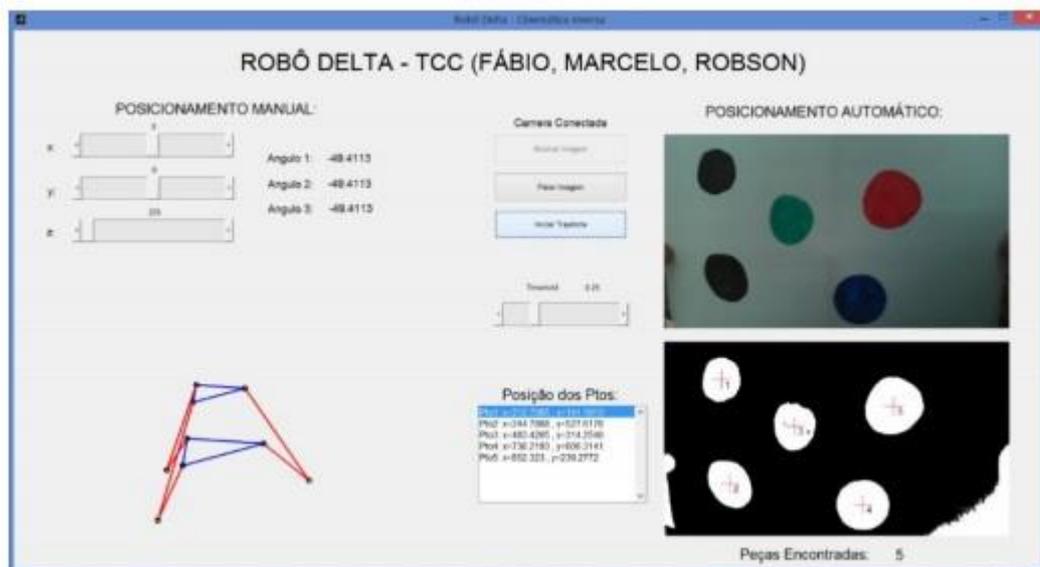
Com a utilização do MatLab o grupo desenvolveu uma interface gráfica, como pode ser visto na Figura 29, e um sistema de processamento digital de imagens que consegue reconhecer o centro de diferentes círculos coloridos em uma folha de papel, os centros foram utilizados como ponto de referência e o sistema de visão computacional consegue calcular as coordenadas (X, Y, Z) do centro do círculo, também é possível selecionar manualmente um conjunto de coordenadas (X, Y, Z) para posicionar o efetuador.

Figura 28 - Proposta Montagem pelos graduandos do IFSC



Fonte: Jubelli, Lima e Hortsmann (2013, p.93)

Figura 29 - Tela software desenvolvido



Fonte: Jubelli, Lima e Hortsmann (2013, p.63)

3.2 Robô Delta ABB FlexPicker®

A ABB (Asea Brown Boveri) é uma das maiores empresas de robótica industrial do mundo, ela possui diversos tipos de manipuladores robóticos dentre eles a linha IRB 360 FlexPicker® que pode ser visualizado na figura 3a), que é a representante dos robôs delta, esta linha foi otimizada para tarefas de empacotamento em alta velocidade, com uma satisfatória capacidade de carga. Esta linha de robôs é comercializada seguindo dois parâmetros cruciais, a capacidade de carga suportada pelo mecanismo, ou seja, o peso do objeto que ele pode sustentar que pode ser de 1 kilograma (Kg), 3 kg, 6 kg ou 8 kg, e a outra característica relevante é o alcance do efetuador que pode ser de 800 mm, 1130 mm ou 1600 mm. Outras duas características observadas são a alta repetibilidade de 0,1 mm, e o tempo de ciclo para um diferente conjunto de operações. (ABB, 2018).

Esta análise é interessante porque ao final do trabalho será possível levantar esses dados obtidos do protótipo, e comparar os resultados com um equipamento comercializado.

3.3 Conclusões e Diferenciais

Após ler as sugestões dos trabalhos acadêmicos e observação do robô delta da ABB, foi levantado uma lista de itens relevantes durante a execução do trabalho:

- a) utilizar materias leves para não sobrecarregar os atuadores, os materiais também devem possuir grande rigidez mecânica para evitar problemas com vibração;
- b) as juntas esféricas devem ser de alta qualidade, não permitindo folgas uma vez que elas interferem no posicionamento do efetuador;
- c) ajustar o posicionamento da câmera para sofrer o mínimo possível com o problema da perspectiva, e manter as hastes do delta fora do campo de aquisição de imagens;
- d) utilização de um fundo e peças com cores bem distintas para facilitar o processamento de imagens;
- e) sistema de comunicação robusto implementando técnicas de integridade de dados;
- f) fazer ensaios para levantar as características técnicas do protótipo;

- g) fazer ensaios para levantar características técnicas do sistema de detecção de peças.

Todos os trabalhos acadêmicos propostos fazem uso de *softwares* que exigem licença para sua utilização, seja o MatLab da Mathworks® ou o LabView da National Instruments™, a proposta desse trabalho utilizará software *open source* e duas placas de desenvolvimento a *Raspberry Pi* e a *bluepill*, a proposta também inclui a utilização da biblioteca livre *OpenCV*, utilizada mundialmente para aplicações acadêmicas e comerciais na área de visão computacional. Outra observação é que nos trabalhos citados não foi realizado nenhum tipo de análise apresentando dados de erro de posicionamento do manipulador, erro de detecção de peças que são os parâmetros principais de robôs que executam a função *pick and place*.

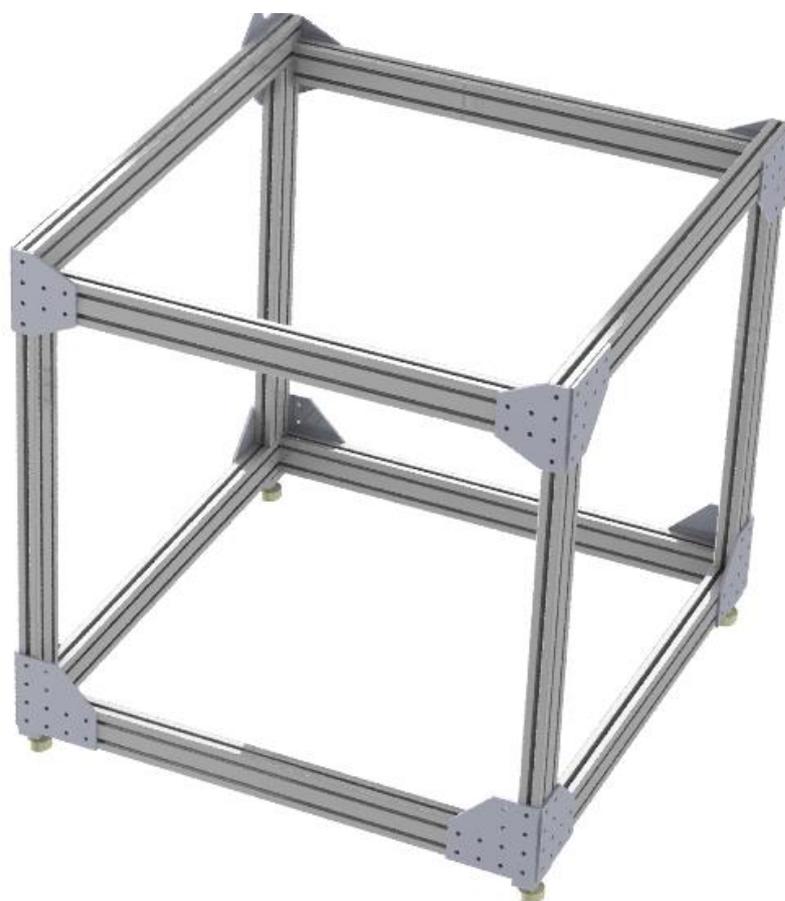
4. METODOLOGIA

Neste capítulo é apresentada a proposta de montagem do trabalho, levando em consideração toda a pesquisa realizada dos capítulos dois e três, também é apresentada algumas restrições de escopo. A metodologia se divide em quatro subcapítulos a construção do protótipo, o sistema de detecção de imagens, o sistema de controle do robô delta e a métricas de avaliação do protótipo.

4.1 Construção do protótipo

A carcaça de sustentação do protótipo foi montada com perfil de alumínio estrutural 20x20 mm como a da Figura 30, a estrutura tem aproximadamente 500x500x1000 mm, onde foram acrescentados uma base e um topo em *Medium Density Fiberboard* (MDF) ou Fibra de Média Densidade.

Figura 30 - Exemplo de Estrutura



Fonte: Tekkno (2018)

A construção do robô delta é uma das mais simples de todos os sistemas robóticos industriais, o manipulador é composto basicamente de três motores posicionados em 120° , juntas esféricas, hastes lineares, base fixa, base móvel e efetuator.

Os motores selecionados para realizar a movimentação do protótipo são servos motores 3325mg com capacidade de carga de 25 kgf/cm, tensão nominal 6 volts, e ângulo de rotação de 180° . A Figura 31 mostra o servo motor.

Figura 31 - Servo motor 3325mg



Fonte: Elaborado pelo autor

Como as hastes lineares foram usados tubos de fibra de carbono como os da Figura 32, a fibra de carbono é muito utilizada na área de aeromodelismo por serem extremamente leves e rígidas, ambos critérios cruciais para a construção do protótipo. Os tubos possuem diâmetro externo de 5 mm e interno de 3,5 mm.

Figura 32 - Tubos de fibra de carbono



Fonte: Elaborado pelo autor

Como visto nos trabalhos correlatados, a junta esférica é uma peça muito importante do robô delta, juntas com baixos ângulos de pivotação resultam em redução no alcance do efetador, restringindo o espaço de trabalho, a junta escolhida foi a WGRM-05 LC MS da Iigus® mostrada na Figura 33, devido ao baixo custo e possuir um ângulo de pivotação de 25°.

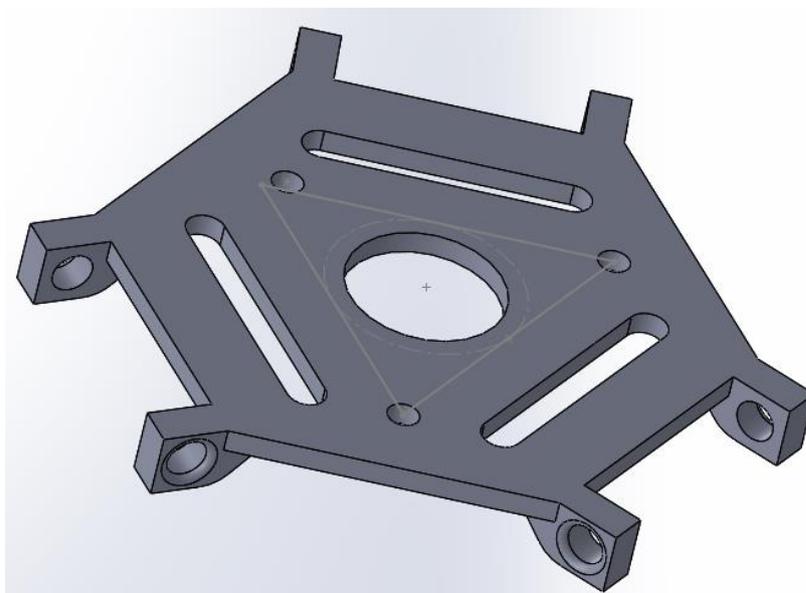
Figura 33 - União articulada angular



Fonte: Elaborado pelo autor

A base móvel foi projetada no *SolidWorks* e impressa em 3D, o projeto previu o encaixe das uniões articuladas, e a furação para a fixação da garra, a Figura 34 mostra o projeto da base móvel do protótipo que foi impresso em 3D.

Figura 34 - Base inferior



Fonte: Elaborado pelo Autor

O efetuator final escolhido foi uma garra articulada de três dedos, ela foi desenvolvida para capturar objetos cilíndricos, é uma peça impressa em 3D e sua movimentação é realizada por um servo motor, o micro servo 9g que possui um torque de 1,8 kgf/cm. A

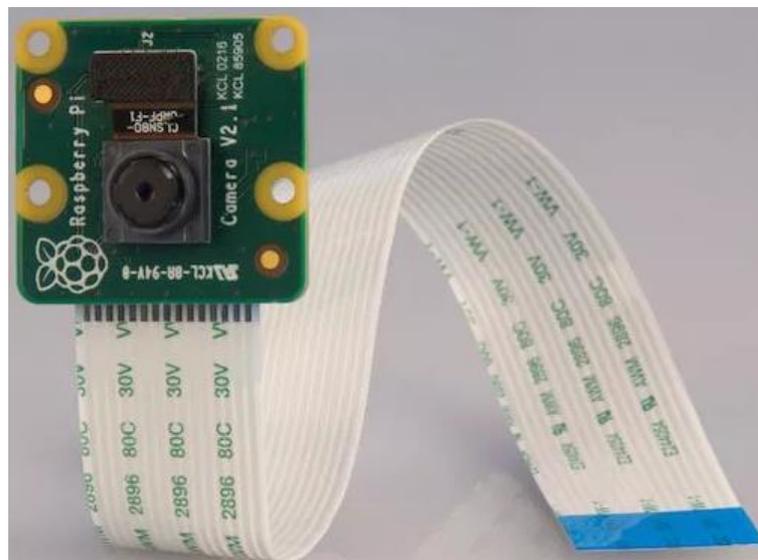
Figura 35 - Garra robótica com três dedos articulados



Fonte: Elaborado pelo Autor

O sistema também possui uma câmera que realiza a aquisição das imagens para processamento e reconhecimento de padrões, a câmera possui 5 *MegaPixel*, capaz de tirar fotos em 2592 x 1944 e ângulo de visão de 65°.

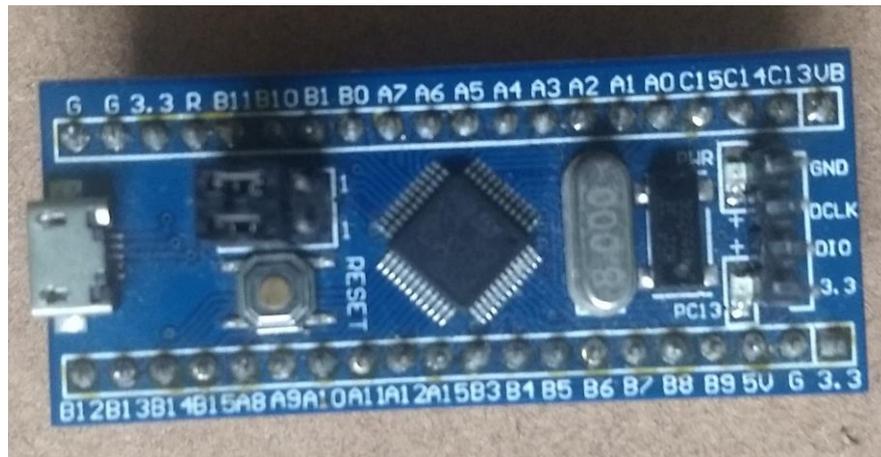
Figura 36 - Câmera Raspberry



Fonte: Lifehacker (2016)

O projeto contém duas placas de desenvolvimento, uma delas chamada de *bluepill*, que pode ser vista na Figura 37, esta placa contém um microcontrolador ARM-M3 o Stm32F103, ela é responsável pelo controle do robô delta, ou seja, pelos algoritmos de cinemática inversa, planejamento de trajetória e controle dos PWM para o acionamento dos servomotores.

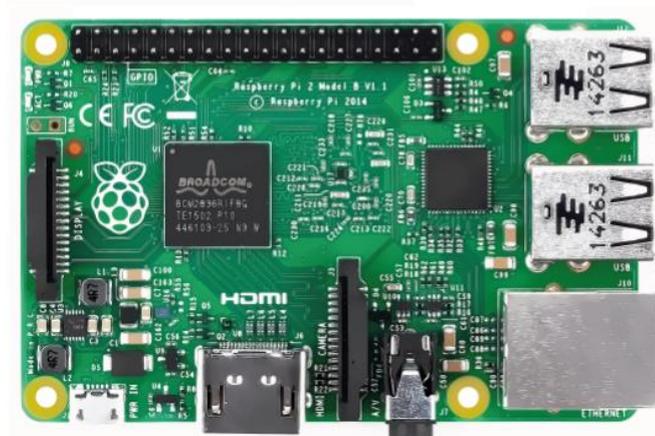
Figura 37 - Placa de desenvolvimento BluePill



Fonte: Elaborado pelo Autor

Enquanto isso a placa *Raspberry Pi 3 Model B* da Figura 38, é responsável pelo processamento de imagem utilizando a biblioteca *OpenCv* e enviando as coordenadas das peças para a *bluepill* através de interface serial.

Figura 38 - Raspberry Pi 3 Model B+



Fonte: Rhydolabz (2015)

Os servomotores possuem tensão nominal de 6V sendo aplicável até 6,8V caso necessite de um aumento no torque do motor. A plataforma contém um conversor DC-DC do tipo *buck* para rebaixar a tensão da fonte de 12V. A corrente máxima do dispositivo é 9 A. O conversor pode ser visto na Figura 39

Figura 39 - Conversor DC-DC



Fonte: Elaborado pelo Autor

4.2 Sistema de visão computacional

O reconhecimento de padrões em sistemas de visão computacional é um processo complexo, para facilitar esse processo quando possível os ambientes são controlados, tentando evitar situações não previstas na etapa de treinamento. O sistema proposto é capaz de reconhecer uma determinada peça padrão em ambiente controlado. A ênfase do sistema de detecção de peça concentra-se na coordenada resultante do sistema, não foi avaliada a qualidade da detecção do objeto padrão, ou seja, o sistema pode detectar de forma equivocada objetos que sejam semelhantes ao objeto padrão.

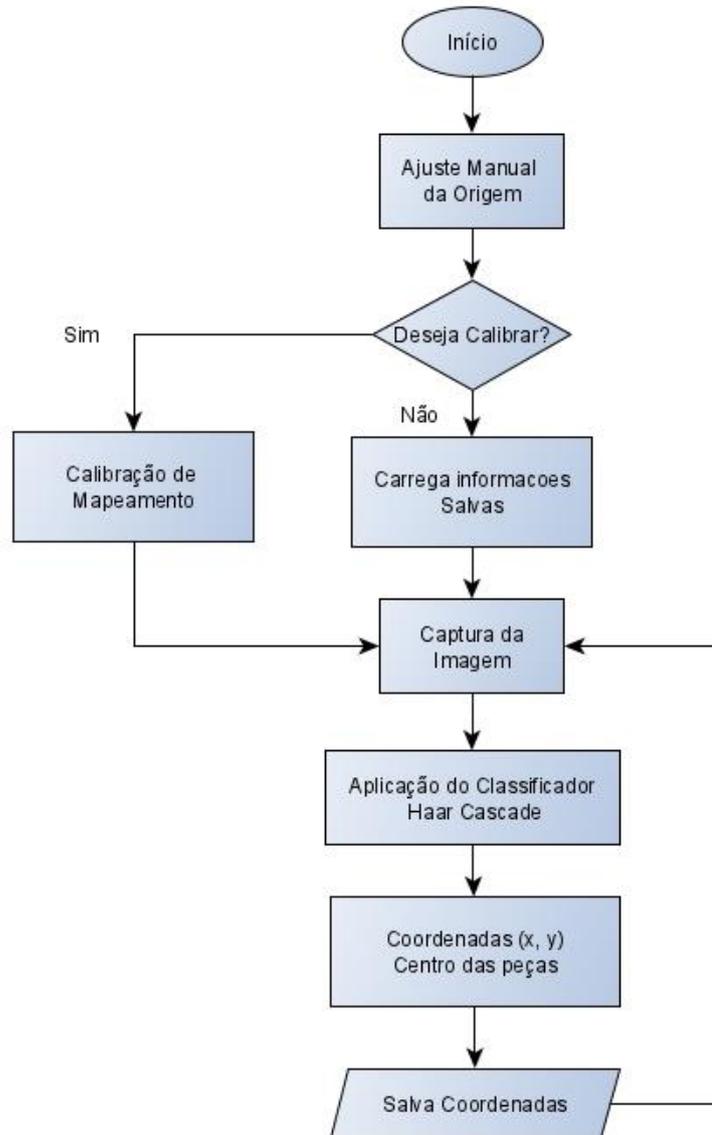
O sistema de detecção de objetos tem duas etapas bem distintas o treinamento e a detecção. Para o treinamento do sistema diversas imagens foram

adquiridas através da planta, essas imagens são utilizadas para o treinamento do classificador de *Haar Cascade*. Foram treinados diversos classificadores para obter bons resultados de detecção das peças através da variação dos parâmetros de treinamento.

Para a etapa de execução o sistema segue o fluxograma da Figura 40:

- a) Primeiramente é realizada uma configuração manual tendo que o usuário indicar as coordenadas da localização da origem do sistema;
- b) como segundo passo foi inserido uma rotina para realizar a calibração, para encontrar a relação entre pixel e milímetros, caso o usuário já tenha essa relação essa etapa poderá ser ultrapassada;
- c) após é realizada a aquisição da imagem através da câmera;
- d) a imagem adquirida será enviada para o classificador, para extrair as coordenadas das peças;
- e) como saída o sistema de visão computacional deve enviar ao programa principal as coordenadas de todas as peças localizadas dentro do campo de visão.

Figura 40 - Fluxograma do Processamento de detecção de peças



Fonte: Elaborado pelo Autor.

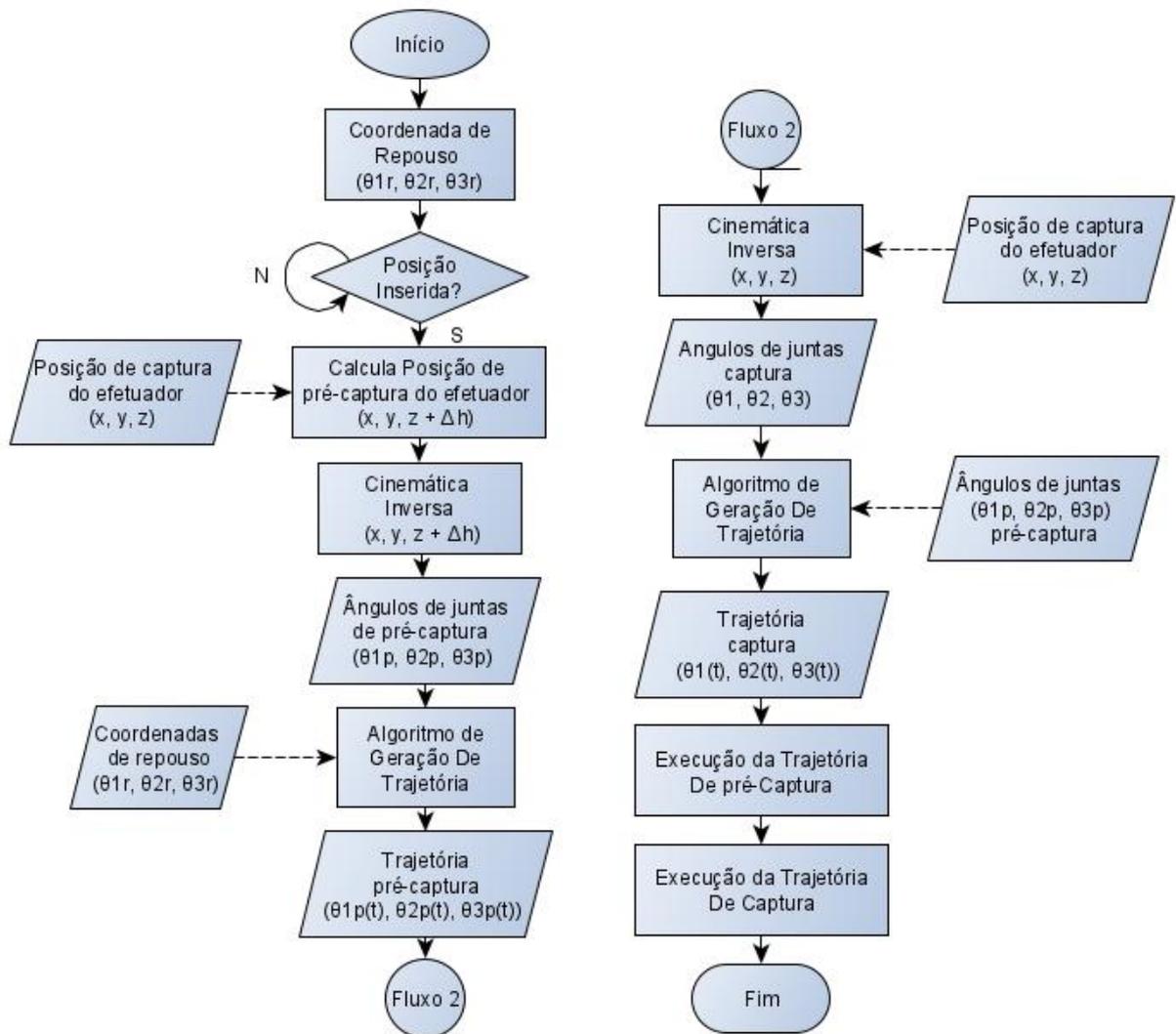
4.3 Sistema de controle do robô delta

O sistema de controle de robô é baseado em duas equações básicas para a movimentação do robô delta, são elas a cinemática inversa e a equação de geração de trajetória, sabemos que o sistema de detecção de peças nos apontará uma coordenada (x, y, z) com essa informação podemos aplicar a cinemática inversa para encontrar o conjunto de ângulos que os servomotores devem se posicionar para que o efetuador final esteja na coordenada desejada, após saber onde estamos

e onde queremos chegar precisa-se calcular o caminho que conduzirá o efetuator entre esses dois pontos, essa etapa é responsabilidade da equação de geração de trajetória, ou seja, a equação que define os ângulos dos servomotores em todo o instante t , essa equação é definida por um polinômio cúbico envolvendo aceleração, velocidade e posição. Essa combinação de cinemática inversa e planejamento de trajetória são os dois passos cruciais para a execução de movimentos de um robô do tipo delta. O fluxograma da Figura 41 mostra o processo para gerar a movimentação do robô delta:

- a) No início o robô sai de um determinado ponto de repouso;
- b) para movimentar o robô é necessário que seja inserido no sistema um conjunto de coordenadas (x, y, z) onde se deseja posicionar o efetuator, caso o contrário nada acontece no sistema;
- c) nessa coordenada é somada uma altura de pré-captura necessária para capturar do objeto pelo topo, caso essa altura não fosse estipulada o efetuator poderia colidir com o objeto que deseja-se capturar;
- d) a coordenada de pré-destino deve ser transformada em ângulos no espaço das juntas através da cinemática inversa, que resultará nos ângulos de pré-captura $(\theta_{1p}, \theta_{2p}, \theta_{3p})$;
- e) os ângulos atuais e desejadas do efetuator são as variáveis de entrada para a geração da trajetória, será utilizado um polinômio cúbico como perfil de geração de trajetória, esta etapa nos resultará na trajetória $(\theta_{1p}(t), \theta_{2p}(t), \theta_{3p}(t))$;
- f) após inicia-se o fluxograma 2 onde é basicamente o primeiro fluxograma, porém dessa vez é calculada a trajetória do ponto de pré-captura até o ponto de captura $(\theta_1(t), \theta_2(t), \theta_3(t))$;
- g) após o sistema ira realmente executar as duas funções de trajetórias finalizando o processo;

Figura 41 - Fluxograma robótica



Fonte: Elaborado pelo Autor.

4.4 Métricas para avaliação de resultados

Após o desenvolvimento dos algoritmos de controle do robô delta e de visão computacional foram feitos um conjunto de ensaios para levantar suas características técnicas.

4.4.1 Erro absoluto de detecção

Para o sistema de detecção de peças foi construída uma tabela com o erro absoluto em diversas coordenadas (x, y), foi fixado um papel milimetrado na plataforma de trabalho e inserida a peça padrão em diversas posições conhecidas e anotado os valores da detecção.

4.4.2 Erro absoluto de posição

O sistema robótico assim como o sistema de detecção de peças também teve seu erro absoluto avaliado, foi colada novamente a folha de papel milimetrado na plataforma de trabalho do robô e foi movimentado o efetuador final com o intuito de marcar a posição de captura, o teste de erro absoluto será feito em apenas uma altura, a altura exata para a captura da peça padrão.

4.4.3 Teste *Pick and Place*

Como teste final onde foi avaliada a integração entre os dois sistemas, o de detecção e o robótico. O teste consiste em uma simulação da tarefa de *pick and place*, foi posicionada na plataforma de trabalho uma peça por vez e quantas vezes o sistema falha em pegar e largar a peça no lugar determinado. O interessante desse teste é que temos a soma dos erros da detecção e posicionamento. Os dados do teste estão apresentados em formato de tabela contendo o número de tentativas totais, número de tentativas bem sucedidas, e o percentual de efetividade.

5. DESENVOLVIMENTO

Esta secção explica como foi realizado o desenvolvimento do trabalho de forma detalhada e será dividido em três subcapítulos hardware onde é apresentada imagens, explicações da montagem física, alguns esquemas elétricos, o sistema de detecção de peças que apresenta todas as etapas necessárias para a detecção de objetos e extração das coordenadas para entregá-las ao programa principal, e o sistema de controle do robô contendo equação de cinemática inversa e planejamento de trajetória.

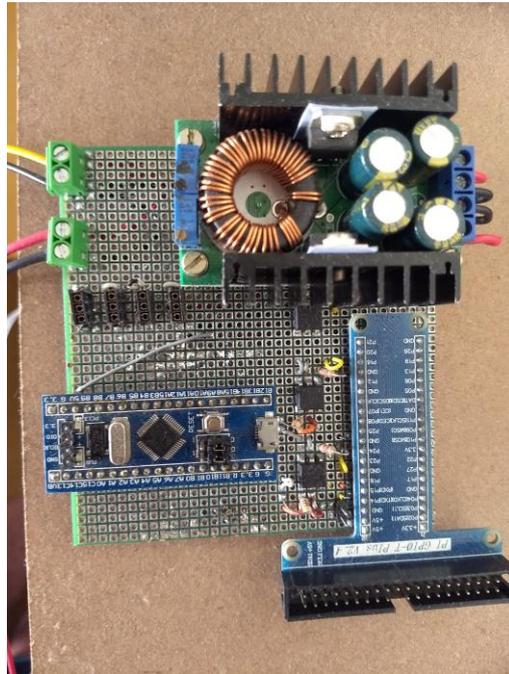
5.1. Hardware

O sistema foi montado com uma *raspberry pi 3 model B* onde é realizado a análise da imagem, extração da coordenada e envio da mesma pela interface serial. A placa *bluepill* fica responsável pelo cálculo da cinemática inversa e algoritmo de geração de trajetória esta placa também atua sobre o PWM que realiza o controle do servo motores.

A planta também possui uma fonte ATX 300W, onde estão ligados a fita LED (*Light Emissor Diode* ou diodo emissor de luz) e o conversor DC-DC para rebaixar a tensão para os servos.

A Figura 42 mostra a placa montada com o conversor Buck entrada 12 V da fonte ATX e saída regulada em 6 V, o conector adaptador da *raspberry*, nele passa os pinos de comunicação serial da *raspberry* que estão optoacoplados com a comunicação serial da *bluepill* também é possível ver a *bluepill* e os conectores para ligação dos 4 servomotores do robô delta. No Apêndice A é possível ver o esquemático do projeto.

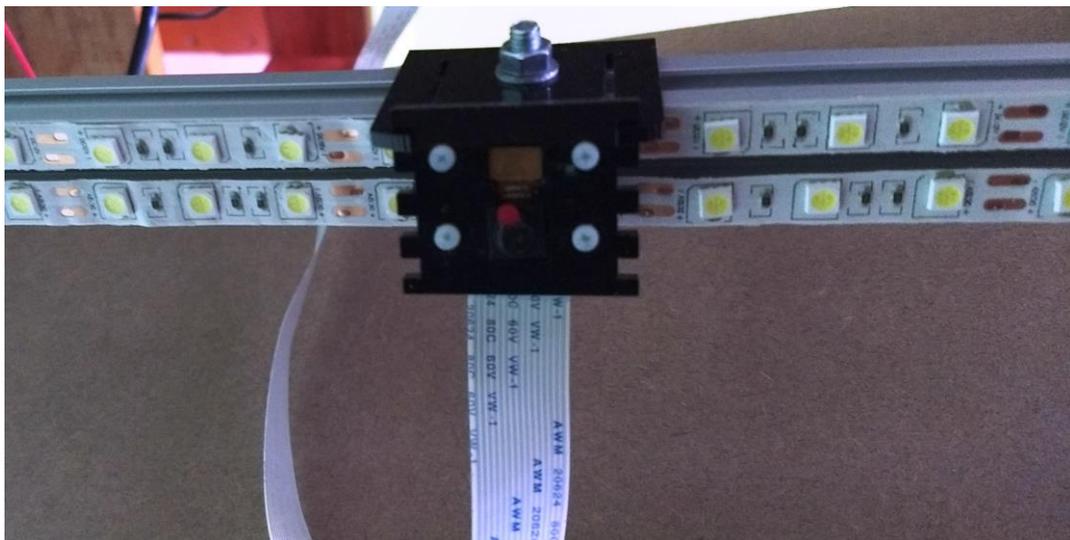
Figura 42 - Placa utilizada



Fonte: Elaborado pelo Autor.

Também foi instalado na plataforma uma iluminação artificial com fita LED branca para evitar problema com sombras, elas foram fixadas atrás da câmera como pode ser visto na Figura 43.

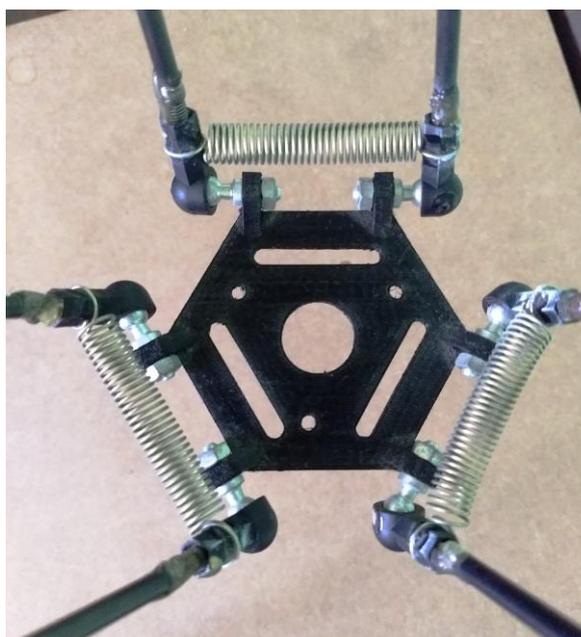
Figura 43 - Câmera e iluminação



Fonte: Elaborado pelo Autor.

A Figura 44 mostra o resultado da base móvel impressa em 3D com a furação necessária para o acoplamento com a garra e a fixação das juntas esféricas. Para isso foram necessárias a instalação de molas de tração porque a junta esférica precisa ser fixada com pressão.

Figura 44 - Base impressa em 3D



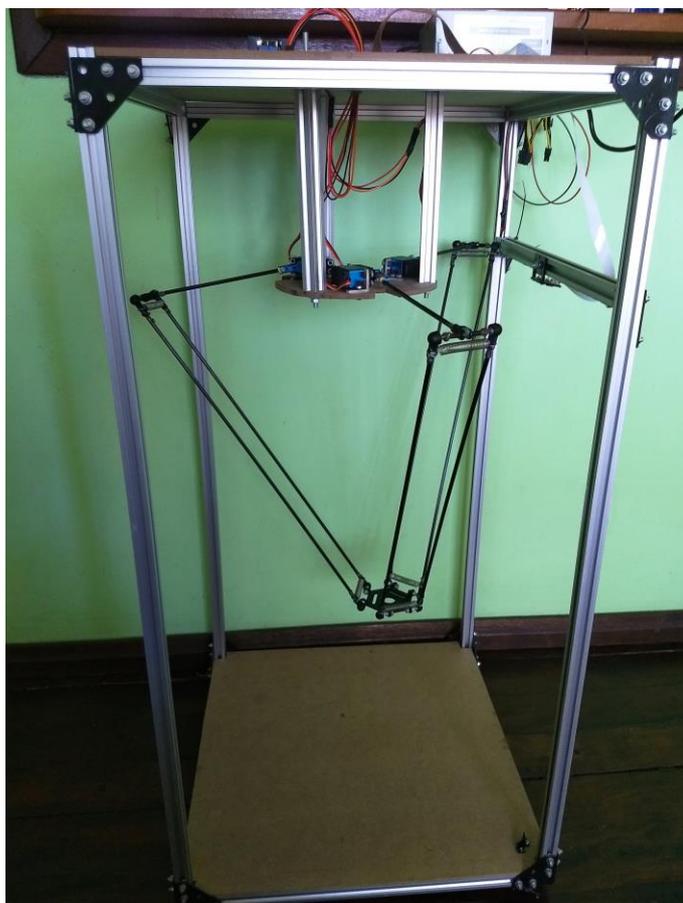
Fonte: Elaborado pelo Autor.

A Figura 45 apresenta a proposta do protótipo onde é possível ver a câmera, os três servomotores na base fixa, as juntas esféricas, as guias lineares, e a base móvel. Na Construção robótica que será utilizada posteriormente nos cálculos de cinemática inversa os parâmetros construtivos do robô delta foram:

- Braço Superior – 210 milímetros;
- Braço Inferior – 550 milímetros;
- Raio Base Fixa – 72 milímetros;
- Raio Base Móvel – 43 milímetros;

Lembrando que os raios são considerados os raios de acoplamentos no caso base fixa refere-se a distância do centro ao eixo dos motores enquanto da base móvel refere-se a distância do centro até as juntas articulares.

Figura 45 - Montagem do Protótipo



Fonte: Elaborado pelo Autor.

5.2. Sistema de detecção de peças

O sistema de detecção de peças consiste em cinco etapas, o treinamento do classificador, o ajuste do ponto de origem, a calibração do sistema, a conversão de pixel para milímetro e obviamente a detecção propriamente dita, porém após todas as etapas anteriores concluídas a detecção se torna simplesmente uma rotina de funções.

5.2.1 Treinamento classificador de *Haar Cascade*

Para desenvolver o sistema de detecção de peças foi utilizado o algoritmo de *Haar Cascade* através da interface gráfica *Cascade Trainer GUI*, ela é uma aplicação que disponibiliza recursos gráficos, porém utiliza os códigos fontes da biblioteca *OpenCv*, essa ferramenta foi utilizada para facilitar o processo de

treinamento. Foram capturadas diversas fotos da peça padrão na plataforma de atuação do robô como mostra a Figura 46.

Figura 46 - Foto capturada da plataforma de trabalho do robô delta



Fonte: Elaborado pelo Autor.

Para o treinamento do classificador foram utilizadas 169 imagens positivas semelhantes a Figura 47, também foram utilizadas 4000 imagens negativas, ou seja, imagens que não possuíam o objeto padrão.

Figura 47 - Peça padrão



Fonte: Elaborado pelo Autor.

Foram realizados diversos treinamentos de classificadores, variando os parâmetros de treinamento e observando a influência nos critérios de análise de desempenho que serão apresentados no capítulo seis. Esta etapa é demorada devido ao fato que o treinamento de um classificador pode levar várias horas e depois de terminado ainda é necessário testá-lo.

5.2.2 Ponto de origem

Ambos os sistemas tanto a detecção de peças e o sistema robótico precisam se comunicar utilizando um mesmo sistema de coordenadas. Para o sistema de visão computacional a coordenada de origem (X_{origem} , Y_{origem}) será definida de forma manual como mostra a Figura 48 o *software* solicitando ao usuário a coordenada (X_{origem} , Y_{origem}).

Figura 48 - Diálogo de ajuste de origem

```
Digite a coordenada de origem X0  
218  
Digite a coordenada de origem Y0  
440  
Gostaria de aplicar esse ponto como origem? [S,N]  
S
```

Fonte: Elaborado pelo Autor.

Após o usuário indicar a coordenada ela é atualizada na tela através de um quadrado branco, é perguntado ao usuário se esse é o ponto que ele deseja utilizar de referência de origem, o usuário deve posicionar o quadrado no delimitador fixado na plataforma, como se pode observar na Figura 49.

Figura 49 - Escolha coordenada de origem



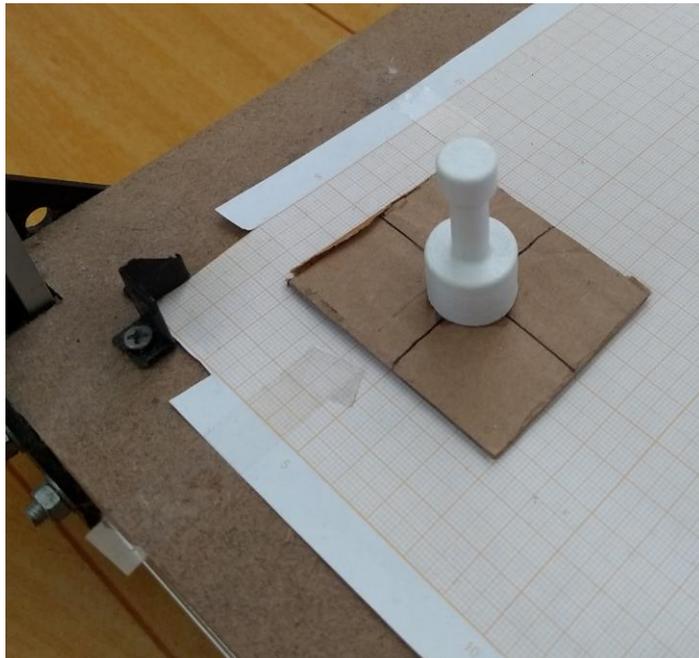
Fonte: Elaborado pelo Autor.

5.2.3 Calibração do sistema de detecção

Para que a conversão de pixel para milímetros seja possível foi desenvolvido um algoritmo para executar a calibração do sistema de detecção.

Primeiramente como pode ser visto na Figura 50, foi fixado uma folha de papel milimetrado sobre a plataforma de trabalho e o usuário posiciona a peça padrão em coordenadas notáveis (regiões de contorno e múltiplos de cinquenta milímetros). Para que o sistema de detecção funcionasse foi necessário a utilização de um fundo escuro devido peça padrão não ser identificada em cima de superfícies muito claras.

Figura 50 - Sistema de calibração



Fonte: Elaborado pelo Autor.

Após como pode ser visto na Figura 51 o classificador detecta a peça como é possível ver na marcação em azul gerada pelo software, logo após o usuário confirmar a detecção da coordenada e o sistema indica a próxima coordenada que deve ser posicionada a peça padrão até completar todo o mapeamento da superfície.

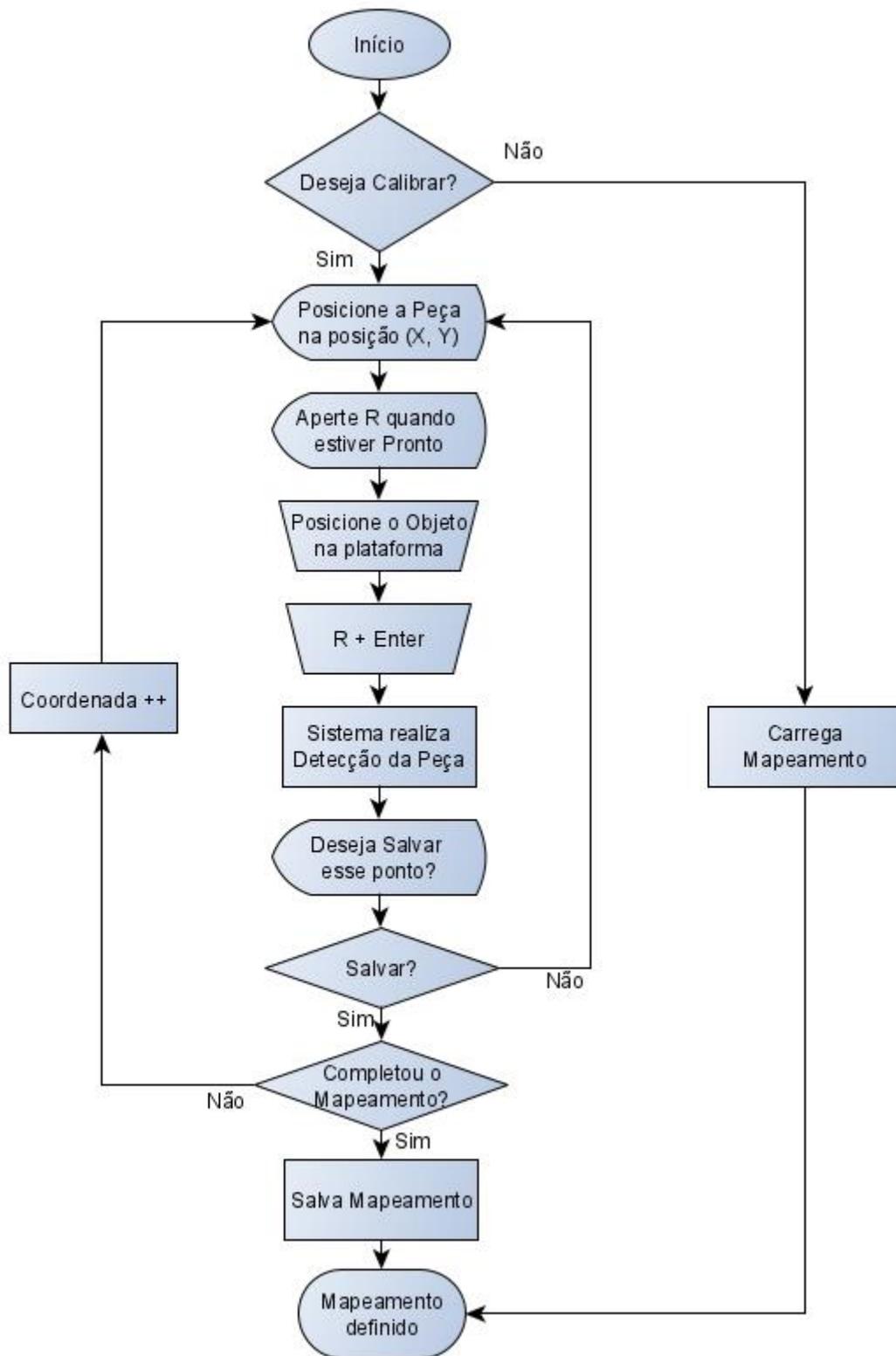
Figura 51 - Sistema de calibração em execução



Fonte: Elaborado pelo Autor.

A Figura 52 apresenta o fluxograma de calibração do sistema, caso o usuário não deseje realizar a calibração o sistema irá buscar um arquivo de calibração realizado anteriormente.

Figura 52 - Fluxograma de calibração



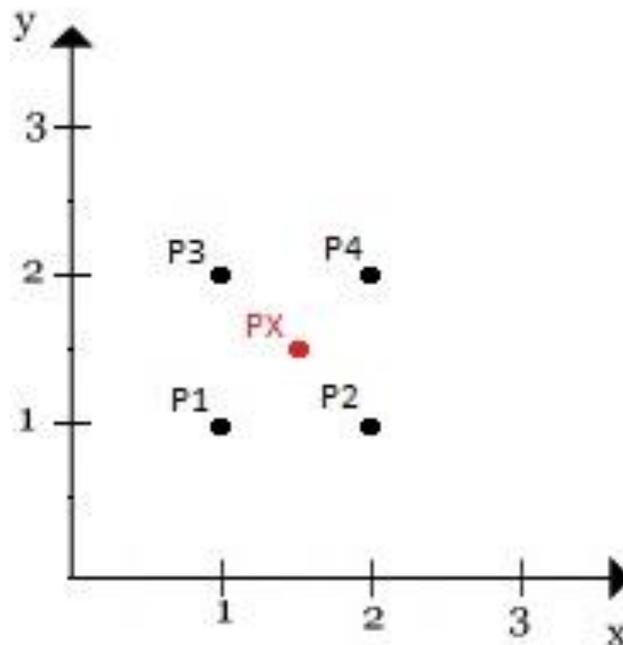
Fonte: Elaborado pelo Autor.

5.2.4 Extração de coordenadas

A detecção do objeto realizada pelo algoritmo de *Haar Cascade* retorna as coordenadas em pixel e essa medida precisa ser convertida para milímetro. A partir do mapeamento realizado previamente um ponto qualquer P_x pode ser convertido de pixel para milímetro.

A Figura 53 mostra um ponto qualquer P_x , onde primeiramente o algoritmo encontra as quatro coordenadas notáveis mais próximas (P_1, P_2, P_3, P_4).

Figura 53 - Calculo de conversão de coordenada



Fonte: Elaborado pelo Autor.

Após é calculado a distância absoluta de cada ponto (P_1, P_2, P_3, P_4) até P_x , utilizando a equação 5.1. Onde $d_{p_1p_x}$ é a distância entre o ponto P_1 e o ponto P_x e X_{p_x} e X_{p_1} são as coordenadas X de seus respectivos pontos e Y_{p_x} e Y_{p_1} são as coordenadas Y de seus respectivos pontos.

$$d_{p_1p_x} = \sqrt{(X_{p_x} - X_{p_1})^2 + (Y_{p_x} - Y_{p_1})^2} \quad (5.1)$$

Após é calculado os respectivos ganhos (K_1, K_2, K_3, K_4) com a equação 5.2, para ambas as coordenadas X e Y que convertem as coordenadas notáveis de pixel para milímetros obtidas através do mapeamento de coordenadas no processo de calibração.

$$K_x = \frac{Coord_{pixel}}{Coord_{mm}} \quad (5.2)$$

Por último é calculado o valor da coordenada em milímetros utilizando a equação 5.3. Onde é calculado um ganho para multiplicar o valor da Coordenada em pixel pela média ponderada dos ganhos nas coordenadas notáveis próximas ao ponto de interesse dando maior peso ao ganho mais próximo da coordenada do sistema.

$$Coord_{mm} = Coord_{pixel} \times \left(\frac{k_1 \times d_1^{-1} + k_2 \times d_2^{-1} + k_3 \times d_3^{-1} + k_4 \times d_4^{-1}}{d_1^{-1} + d_2^{-1} + d_3^{-1} + d_4^{-1}} \right) \quad (5.3)$$

5.3. Sistema Robótico

O controle do robô delta conforme visto anteriormente é realizado a partir de dois itens, a cinemática inversa e a geração de trajetória. Primeiramente é necessário informar ao sistema a coordenada que se deseja posicionar o efetuador do manipulador. Utilizando a cinemática inversa será encontrado um conjunto de juntas que satisfaçam esse sistema. Conhecendo as coordenadas de chegada e de fim passa-se para a geração de trajetória, onde é gerado o caminho que os ângulos de juntas irão fazer para guiar o efetuador do ponto inicial até o ponto final.

5.3.1 Cinemática Inversa

A cinemática inversa consiste no cálculo que transforma as coordenadas cartesianas em coordenadas no domínio das juntas que devem ser aplicados aos motores para obter a posição desejada no efetuador, diferentes dos manipuladores séries que normalmente utilizam as equações de Denavit–Hartenberg para as topologias em paralelo essa metodologia não se enquadra de forma muito efetiva devido a alta complexidade, atualmente também existem abordagens utilizando

redes neurais artificiais para executar essa operação, porém para o robô delta foi utilizado uma abordagem matemática baseada em geometria especial. O sistema de equações escolhido para o presente trabalho é do estudo de (YAN; LIU; JIAO, 2016), estudo que também foi utilizado pela dissertação de mestrado de (LENSARIN, 2017) que propôs algumas alterações no cálculo da cinemática inversa.

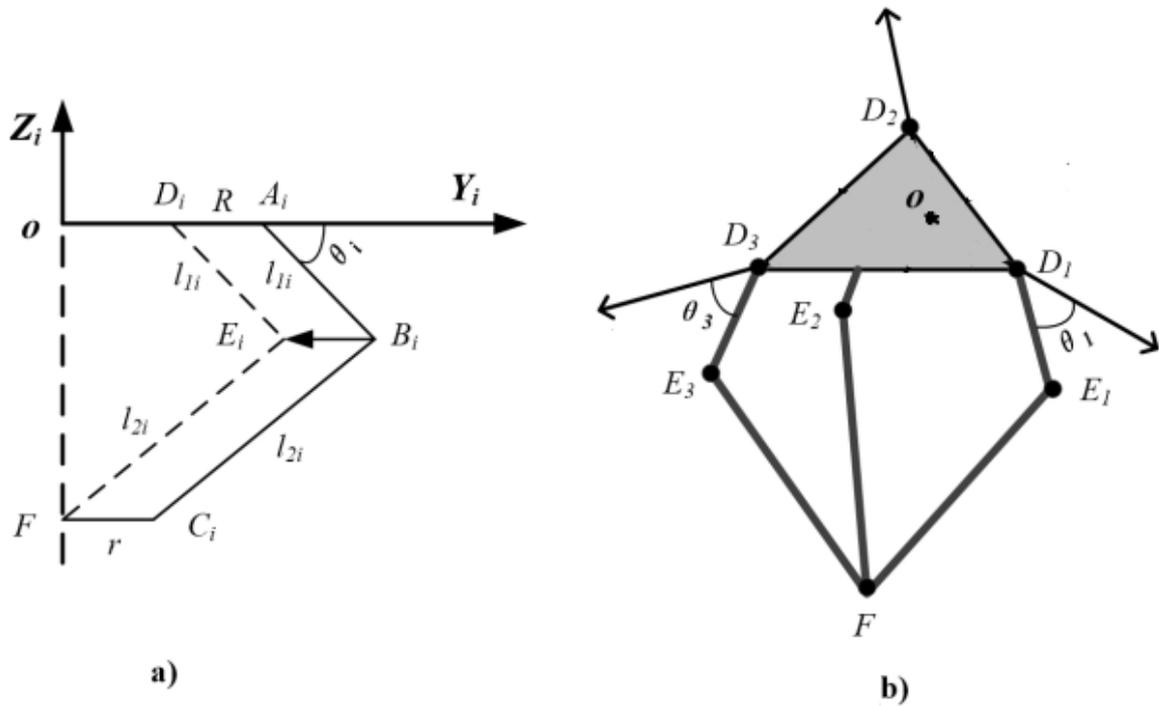
O primeiro passo é realizar uma simplificação do manipulador robótico, simplificando sua geometria como é possível observar na Figura 54. (YAN; LIU; JIAO, 2016).

Principais parâmetros do robô (YAN; LIU; JIAO, 2016):

- l_1 - comprimento do braço superior;
- l_2 - comprimento do braço inferior;
- R - raio da base fixa;
- r - raio da base móvel;
- F - centro da plataforma móvel;
- B_i - coordenada real da junta intermediária;
- D_i - coordenada simplificada da posição de rotação do eixo do motor;
- E_i - coordenada simplificada da junta intermediária;
- Θ_i - Ângulo do servomotor;

Um fato relevante a se destacar nessa abordagem é que no processo de simplificação são alteradas as posições das juntas e dos elos para a resolução matemática do sistema, o que significa que o sistema perde a capacidade de evitar colisões.

Figura 54 - Simplificação do Robo Delta



Fonte Adaptado de Yan, Liu e Jiao (2016, p. 743)

A simplificação é baseada na alteração das posições das juntas originais para juntas imaginárias ou simplificadas. Primeiramente é calculado a diferença entre o diâmetro real da base fixa e da base móvel conforme equação (5.4). (YAN; LIU; JIAO, 2016, p. 744).

$$h = R - r \quad (5.4)$$

Como o robô é composto por três articulações foi necessário calcular a rotação de coordenadas com base na coordenada F do centro da plataforma móvel desejada, para cada plano de articulação haverá coordenadas diferentes. Este cálculo se repete para os outros braços articulados do sistema substituindo o pelos valores gerados pela equação de rotação do sistema, utilizando ângulos α (0, 120°, 240°). (YAN; LIU; JIAO, 2016, p. 744).

$$\begin{bmatrix} X_{fi} \\ Y_{fi} \\ Z_{fi} \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_f \\ Y_f \\ Z_f \end{bmatrix} \quad (5.5)$$

Com os comprimentos dos braços do robô I1 e I2 é possível inferir equações para determinar parciais do ponto E_i , que representa a coordenada intermediária, que é possível ver nas equações (5.6), (5.7) e (5.8). (YAN; LIU; JIAO, 2016, p. 744).

$$X_{ei} = 0 \quad (5.6)$$

$$l_1^2 = Z_{ei}^2 + (Y_{ei} - h)^2 \quad (5.7)$$

$$l_2^2 = X_{fi}^2 + (Y_{ei} - Y_{fi})^2 + (Z_{ei} - Z_{fi})^2 \quad (5.8)$$

Após é necessário resolver as equações para obter as demais parciais das coordenadas intermediárias. (YAN; LIU; JIAO, 2016, p. 744).

$$Y_{ei} = \frac{(h - \rho_i \delta_i) \pm \sqrt{l_1^2 (1 + \rho_i^2) - (\rho_i h + \delta)^2}}{1 + \rho_i^2} \quad (5.9)$$

$$Z_{ei} = \rho_i Y_{ei} + \delta_i \quad (5.10)$$

onde,

$$\rho_i = \frac{h - Y_{fi}}{Z_{fi}} \quad (5.11)$$

$$\delta_i = \frac{X_{fi}^2 + Y_{fi}^2 + Z_{fi}^2 + l_1^2 - l_2^2 - h^2}{2Z_{fi}} \quad (5.12)$$

Com os parâmetros calculados para finalizar é necessário calcular o ângulo necessário a ser aplicado ao motor com a equação (5.12). (YAN; LIU; JIAO, 2016, p. 744).

$$\theta_i = \sin^{-1} \frac{|Z_{ei}|}{l_1} \quad (5.13)$$

Como é possível ver na equação (5.9) o sistema possui mais de uma solução, ao todo matematicamente existem oito soluções possíveis, o sistema desenvolvido calcula uma coordenada no domínio das juntas que soluciona o problema caso essa solução apresente coordenadas de juntas fora de um determinado intervalo considerado seguro para as juntas ela é descartada e o algoritmo calcula mais uma solução possível.

5.3.2 Planejamento da Trajetória

A técnica de geração de trajetória utilizada foi a de polinômio cúbico. A entrada do sistema de geração de trajetórias são as coordenadas iniciais e finais no espaço das juntas. O equacionamento de trajetória é realizado através do polinômio cúbicos conforme (CRAIG, 2012 p. 194-195). Quatro equações são evidentes duas, delas são que conhecemos os valores iniciais de finais de posição (5.14), e outras duas que a velocidade nesse ponto deve ser nula (5.15). Assim obtemos as quatro condições de contorno do sistema de equações. (CRAIG, 2012 p. 194-195),

$$\theta(t_0) = \theta_i \quad \theta(t_f) = \theta_f \quad (5.14)$$

$$\dot{\theta}(t_0) = 0 \quad \dot{\theta}(t_f) = 0 \quad (5.15)$$

A equação característica de uma função cúbica é dada pela equação (5.16)

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \quad (5.16)$$

Calculando a derivada primeira equação (5.16), e a derivada segunda (5.17) encontramos as equações de velocidade e aceleração. Lembrando que essas grandezas estão no espaço de juntas, ou seja, não representam as grandezas cartesianas do efetuador, e sim do eixo dos servomotores. (CRAIG, 2012 p. 194-195),

$$\dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2 \quad (5.17)$$

$$\ddot{\theta}(t) = 2a_2 + 6a_3 t \quad (5.18)$$

Combinando as equações de posição velocidade e aceleração com as equações de restrição e resolvendo o sistema de equações de forma algébrica utilizando o método dos determinantes são obtidas as seguintes equações. (CRAIG, 2012 p. 194-195),

$$\begin{cases} a_0 = \theta_0 \\ a_1 = 0 \\ a_2 = \frac{3}{t_f^2} (\theta_f - \theta_i) \\ a_3 = -\frac{2}{t_f^3} (\theta_f - \theta_i) \end{cases} \quad (5.19)$$

Para cada movimentação é necessário resolver o sistema de equações (5.19), e gerar a quantidade de pontos desejados. O sistema de detecção de peças envia um conjunto de coordenadas (x, y), onde deseja capturar uma peça padrão. Após será calculado a partir da cinemática inversa o conjunto de juntas que satisfazem (x, y, $Z_{peça} + \Delta h$). Onde Δh é a altura acima da peça que será posicionado o efetuador na pré-captura. Será aplicada outra trajetória entre os pontos de pré-captura e captura da peça, ou seja, são geradas duas trajetórias a partir da solução dos polinômios cúbicos.

5.3.3 Conversão de coordenadas

O sistema de visão computacional possui um sistema de coordenadas enquanto o sistema robótico possui um sistema de coordenada completamente diferente. Para isso foi necessário a utilização de um algoritmo para fazer essa função de remapeamento para os dois sistemas utilizarem a mesma orientação. Foi aplicado um algoritmo que é invocado antes do cálculo de cinemática inversa, inserindo uma coordenada corrigida (X_c, Y_c, Z_c) no sistema.

O sistema por ser um protótipo e não ter sido construído com avançados processos de fabricação ele possui algumas folgas, erros nas medidas de fixação

dos motores, diferença de comprimento de um braço para o outro, acabaram resultando em um erro de posicionamento do efetuador final devido a essas divergências. O algoritmo de remapeamento também será responsável por realizar uma pequena correção nas coordenadas para conseguir posicionar o efetuador na posição desejada.

O fluxograma da Figura 55 ilustra a ordem dos algoritmos, o algoritmo utilizado é similar ao utilizado para realizar a transformação de pixel para milímetros, onde foi realizado um mapeamento das coordenadas (X, Y, Z) e (X_c, Y_c, Z_c) gerando um mapa nos pontos notáveis. Porém dessa vez será inserida uma coordenada cartesiana e a saída também será uma coordenada cartesiana.

Figura 55 - Fluxograma Conversão de coordenada



Fonte: Elaborado pelo Autor.

Para a calibração foi novamente utilizada a folha de papel milimetrado. Foi inserido as coordenadas considerando o sistema de coordenadas do sistema robótico. Essas coordenadas resultantes foram anotadas em seus respectivos pontos no sistema de coordenada do robô e de visão computacional. Desta forma foram armazenadas essas duas coordenadas. e realizado o mesmo processo das equações (5.1), (5.2) porém desta vez com o sistema de coordenadas contendo as 3 coordenadas. Foi necessário o uso das três coordenadas para realizar a correção na altura da peça que apresentava erro em algumas posições. Então é utilizada a equação (5.20) para calcular a conversão da coordenada robótica para coordenada de visão computacional, é necessário realizar o cálculo uma vez para cada ponto da coordenada.

$$Coord_{VC} = Coord_{Rb} \times \left(\frac{k_1 \times d_1^{-1} + k_2 \times d_2^{-1} + k_3 \times d_3^{-1} + k_4 \times d_4^{-1}}{d_1^{-1} + d_2^{-1} + d_3^{-1} + d_4^{-1}} \right) \quad (5.20)$$

6. ANÁLISE DOS RESULTADOS

Este capítulo será dividido em 3 etapas contendo a avaliação do sistema de detecção de peças, a avaliação do sistema robótico, e o resultado do teste *pick and place* do sistema como um todo.

6.1 Avaliação do sistema de detecção de peças

O software Trainer Cascade GUI também disponibiliza uma aba de teste, onde é possível aplicar o classificador num conjunto de imagens. As imagens devem ser diferentes das imagens utilizadas para o treinamento.

Para análise do sistema de detecção foi utilizado os conceitos de verdadeiro positivo (VP), falso positivo (FP) e falso negativo (FN). Verdadeiro positivo significa o número de vezes que o sistema detectou uma peça que realmente estava presente na posição. Falso positivo significa que o sistema detectou uma peça onde ela não existe. E por último falso negativo (FN) significa que o sistema não detectou uma peça que estava presente na plataforma.

O arquivo com imagens de teste possui 10 imagens, sendo que em todas elas existem 6 peças, ou seja, o sistema possui um total de 60 peças de testes. Como comentado anteriormente a qualidade da detecção das peças não é o foco da aplicação. Então a métrica para a escolha do classificador foi selecionar o classificador que possui maior VP e menor FN. Como estamos trabalhando um ambiente controlado o FP será um critério mais folgado, devido ao fato que se uma peça está na área de trabalho do robô, provavelmente ela é uma peça que deseja ser manipulada.

O classificador escolhido entre os treinados possui as seguintes características de treinamento:

- Imagens positivas: 169
- Imagens negativas: 4000
- *Features*: Todas
- Número de Estágios: 12
- Algoritmo: BOOST

Na Figura 56 temos o exemplo da imagem que foi analisada pelo software utilizando o classificador escolhido resultando na detecção de seis verdadeiros positivos.

Figura 56 - Resultado do sistema de teste de detecção de peças



Fonte: Elaborado pelo Autor.

Como a resposta do sistema de visão computacional é dada em milímetros, foi gerada uma tabela com o mapeamento do erro absoluto de medição. O maior erro encontrado foi 4,47 mm. Os valores das medições estão apresentados no Apêndice B, para o sistema de calibração para mapeamento foram utilizadas coordenadas diferentes das coordenadas utilizadas para o cálculo do erro.

6.2 Avaliação do sistema robótico

A avaliação do erro de posicionamento do sistema robótico consistiu na inserção manual das coordenadas desejadas do sistema considerando a origem do sistema de visão e anotando em tabela as coordenadas obtidas, essa tabela de resposta pode ser visualizada no Apêndice C, é possível verificar os erros de posicionamento individual de cada coordenada (X, Y) e o erro absoluto da distância

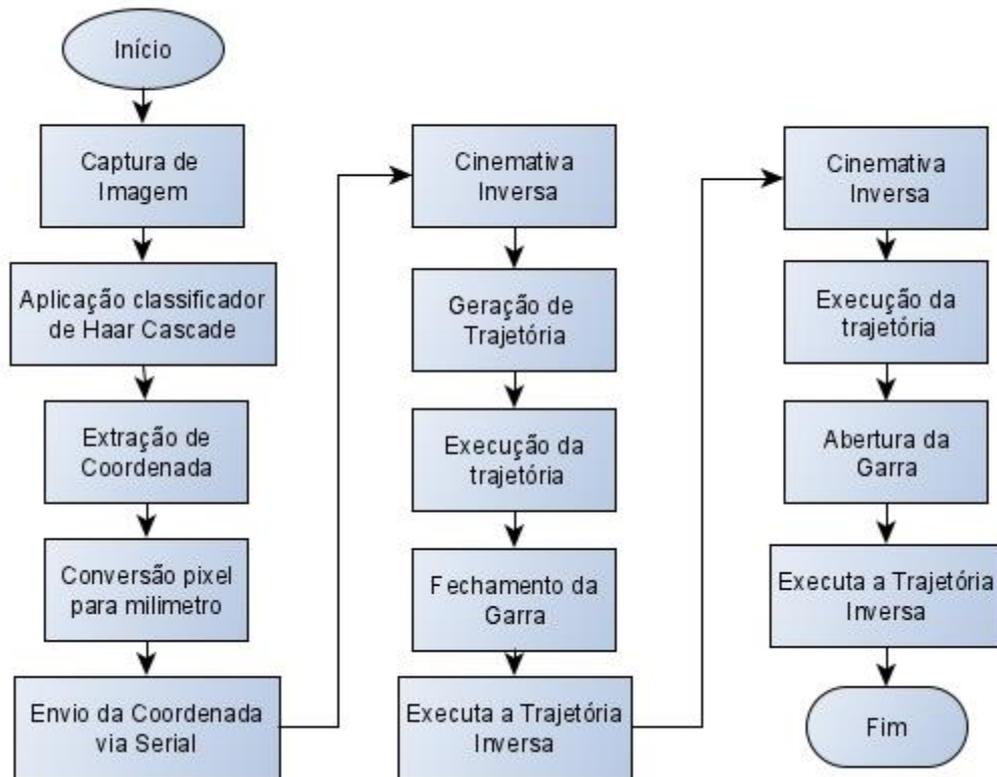
em milímetros do ponto desejado até o ponto medido. O sistema robótico de posicionamento teve um erro máximo de 19,8mm e a média de erro ficou em 11,7 milímetros. Todas as medidas foram tomadas considerando a altura da peça padrão.

O alcance testado do sistema robótico permitiu locomover o efetuador final na área de uma circunferência mínima de 350 milímetros de diâmetro, considerando a altura padrão da peça

6.3 Teste *Pick and Place*

Para o teste de *pick and place* o algoritmo foi conforme a Figura 57. O sistema extrai a coordenada e envia através da comunicação serial para o microcontrolador, o microcontrolador por sua vez recebe a coordenada aplica o algoritmo de correção, calcula as cinemáticas inversas dos pontos de pré-captura e captura e gera as duas trajetórias, após o sistema executa as trajetórias para capturar a peça, fecha a garra e executa a trajetória inversa para voltar ao ponto de repouso, para finalizar o sistema leva a peça até um ponto de entrega predeterminado, depositando a peça padrão.

Figura 57 - Algoritmo para teste *pick and place*



Fonte: Elaborado pelo Autor.

A Figura 58 mostra o protótipo executando a função de *pick and place* capturando uma peça com sucesso.

Figura 58 - Teste de *pick and place* sendo executado



Fonte: Elaborado pelo Autor.

A Tabela 2 foi construída segundos os seguintes parâmetros de análise:

- Captura com Sucesso (CSS): Peça que foi detectada pelo sistema de visão e capturada pela garra do robô delta;
- Erro na captura (EC): Peça que foi detectada pelo sistema de visão, porém houve colisão entre a garra e a peça ou o erro de posicionamento do robô acabou resultado numa considerável distância da peça, de tal modo, que ao realizar o fechamento da garra não ocorreu a captura da peça;
- Largada com sucesso (LC): Peça que foi capturada com sucesso e levada até o ponto de entrega com sucesso;
- Erro na largada (EL): Peças que foram capturadas com sucesso, porém ao depositar na coordenada de entrega a abertura da garra ou movimento até o ponto de pré-captura resultou na peça sendo derrubada.

Tabela 2 - Resultados do teste do *pick and place*

Captura com sucesso (CSS)	Erro na captura (EC)	Largada com sucesso (LC)	Erro na largada (EL)
36	14	32	4

Fonte: Elaborado pelo Autor.

A Tabela 2 apresenta o resultado do teste de *pick and place*, o teste foi realizado com um total de 50 peças (Total) depositadas em posições diferentes da plataforma uma por vez, foram anotadas o número de vezes que se obteve sucesso na execução das funções. O percentual de erro de captura é definido pela equação (6.1) que é a razão entre o número de peças com erro na captura sobre o número total de tentativas. O percentual de erro de captura do sistema foi 28%.

$$\text{Erro Captura}(\%) = \left(\frac{EC}{Total} \right) \times 100 \quad (6.1)$$

A percentual de erro de largada é definido pela equação (6.2) que é a razão entre o número de peças largadas com erro sobre o número total de tentativas que nesse caso é o número de peças que foram capturadas com sucesso. O percentual de erro do sistema foi 11,1%.

$$\text{Erro Largada}(\%) = \frac{EL}{CSS} \times 100 \quad (6.2)$$

A eficiência total do sistema foi calculada pela equação (6.3) que é a razão entre o número de peças capturadas e largadas com sucesso sobre o número total de tentativas. A eficiência do sistema ficou em 64%.

$$\text{Eficiência Total}(\%) = \frac{LC}{Total} \times 100 \quad (6.3)$$

7. CONCLUSÕES

Robótica e sistemas de visão computacional estão sendo muito utilizados na indústria atendendo os mais distintos segmentos, o engenheiro deve estar preparado para se deparar com esses dois tipos de tecnologia. Esse foi o principal motivo para a escolha do tema deste trabalho o aprofundamento do conhecimento nestes dois assuntos tão importantes.

O objetivo do trabalho foi dividido em três etapas principais, sendo elas:

- Treinar e aplicar um algoritmo de visão computacional capaz de detectar e extrair uma coordenada (x, y) de uma peça num determinado plano e levantar o erro de detecção do sistema;
- Construir um protótipo de robô delta para aplicar os algoritmos de cinemática inversa e geração de trajetória com a finalidade de movimentar o robô para a posição desejada, e levantar o erro de posicionamento do efetuador do sistema;
- Executar a função de *pick and place* a função mais comum do robô delta na indústria fazendo uma análise da efetividade da tarefa;

O protótipo foi capaz de atender a todos os objetivos para o qual foi desenvolvido.

Como dificuldades pode se destacar a alta precisão necessária para a usinagem e montagem das peças para garantir que todos os ângulos, posições e distância sejam atendidos. No protótipo foi necessário retrabalhar algumas peças para eliminar folgas e angulações indesejadas, qualquer tipo de desvio da construção física do robô pode resultar em desvios dos resultados de posicionamento esperado.

Outra dificuldade foi o uso da garra mecânica de três dedos, se ela fosse substituída por uma ventosa e uma válvula geradora de vácuo seria possível aumentar a efetividade de robô. Sem alterar em nada o algoritmo fazendo o uso desta ferramenta aumentaria a efetividade do protótipo mostrando o quanto a ferramenta é importante para a tarefa de *pick and place*.

Como trabalhos futuros gostaria de ressaltar uma melhoria no protótipo com foco na redução de custos da aplicação. Que seria a eliminação da Raspberry do sistema, o microcontrolador também ficaria responsável pela aquisição de imagens, detecção de peças e extração de coordenada. Outra oportunidade de trabalho futuro

é testar múltiplos algoritmos de geração de trajetórias, existem técnicas utilizando algoritmos de inteligência artificial como redes neurais para resolução da trajetória uma vez que existem inúmeras trajetórias possíveis para o robô executar, mas escolher a trajetória que consome menos energia ou a trajetória mais rápida ou até mesmo uma trajetória para desviar de outros sólidos presentes no volume de trabalho é de veras interessante.

REFERÊNCIAS

ABB. **IRB 360 FlexPicker**. [S.I.], 2018. Disponível em: <<https://new.abb.com/products/robotics/industrial-robots/irb-360>>. Acesso em: 31 maio 2018.

ABB. **IRB 4400**. [S.I.], 2018. Disponível em: <<http://new.abb.com/products/robotics/industrial-robots/irb-4400>>. Acesso em: 10 maio 2018.

ABB. **Robô IRB 360 – FlexPicker da ABB**. [S.I.], 2018. Disponível em: <<http://new.abb.com/food-beverage/pt/alimentos-bebidas/references/flexpicker-irb-360>>. Acesso em: 10 maio 2018.

ANATOMIA DO CORPO HUMANO. **Olho humano | Globo ocular – Anatomia do olho**. [S.I.], [2018?]. Disponível em: <<http://www.anatomiadocorpo.com/visao/olho-humano-globo-ocular/>>. Acesso em: 12 abr. 2018. Blog: Anatomia do corpo humano.

ANGELES, Jorge. **Fundamentals of robotic mechanical systems: Theory, methods and algorithms**. 1ª edição. Editora Springer, 1997.

ARDUINO. **Whats is Arduino?** [S.I.], [2018?]. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>. Acesso em: 15 maio 2018.

ATMEL. **ATmega328/P**. [S.I.], [2018?]. Disponível em: <http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Summary.pdf>. Acesso em: 15 maio 2018.

BARRIENTOS, Antonio. **Fundamentos de Robótica**. 2ª edição: McGraw-Hill, 2007.

BINARY UPDATES. **Introduction raspberry pi model B**. Umesh Lokhand [S.I.], [2018?]. Disponível em: <<http://binaryupdates.com/introduction-of-raspberry-pi-3-model-b/>>. Acesso em: 16 maio 2018.

CABRAL, Eduardo L.L. **PMR2560 – Robótica efetadores e atuadores**. Notas de aula. São Paulo, USP. Disponível em: <https://edisciplinas.usp.br/pluginfile.php/1692695/mod_folder/content/0/R5_Efetadores%20e%20atuadores.pdf?forcedownload=1>. Acesso em: 1 maio 2018.

CHRISTIE, Dennis A; KUSUMA, Tubagus M; MUSA, Purnawarman. **Chess piece movement detection and tracking, A vision system framework for autonomous chess playing robot**. IEEE. Second International Conference on Informatics e computing (ICIC), Jayapura, 2017, p. 1-6.42

CONDIT, Reston; JONES Douglas W. **AN907 Stepping Motors Fundamentals**. MICROCHIP. Notas de aplicação. [S.l.], 2004. Disponível em: <<http://ww1.microchip.com/downloads/en/AppNotes/00907a.pdf>>. Acesso em: 2 maio 2018.

CRAIG, John J. **Robótica**. Tradução de Heloísa Coimbra de Souza; Revisão técnica Reinaldo Augusto da Costa Bianchi. 3ª edição. São Paulo: Pearson Education do Brasil, 2012.

DANGETI, Pratap. **Statics For Machine Learning**. Packet Publishing, Birmingham - Muambai, 2017. Disponível em: <https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781788295758/4/ch04lvl1sec32/adaboost-classifier>. Acesso em: 4 setembro 2019.

DHGATE. **Kossel mini 3D printer plastic injection molding delta rostock fisheye end effector hot end for jhead E3D hotend**. [S.l.], [2018?]. Disponível em: <<https://www.dhgate.com/store/product/kossel-mini-3d-printer-plastic-injection/259355111.html>>. Acesso em: 5 jun. 2018.

EMBARCADOS. **Sistema embarcado - O que é? Qual a sua importância?** [S.l.], 16 nov. 2013. Disponível em: <<https://www.embarcados.com.br/sistema-embarcado/>>. Acesso em: 15 maio 2018. Blog: EMBARCADOS.

ÉPOCA NEGÓCIOS. **Indústria instala 1,5 mil robôs por ano.** São Paulo, 14 out. 2017. Disponível em:

<<https://epocanegocios.globo.com/Tecnologia/noticia/2017/08/epoca-negocios-industria-instala-15-mil-robos-por-ano.html/>>. Acesso em: 24 mar. 2018.

FABIONES. **O que é mecatrônica?** [S.l.], [2018?]. Disponível em: <<http://www.fabiones.com.br/o-que-e-macatronica.htm>>. Acesso em: 1 maio 2018.

FILIPEFLOP. **Placa de Desenvolvimento Stm32 F103C8T6.** Santa Catarina, [2020?]. Disponível em:

<https://www.filipeflop.com/produto/placa-de-desenvolvimento-stm32-para-arduino/?gclid=Cj0KCQjw5eX7BRDQARIsAMhYLP8RSSF7qF9iESc6rWbzQwyGs9yFtFx5VR58iUIb_P9mbEbfrw93Wi0aAvqxEALw_wcB>. Acesso em: 04 outubro 2020.

FOCUS, Escola de Fotografia. **Uma ilusão de ótica.** [S.l.], [2018?]. Disponível em: <<https://focusfoto.com.br/uma-ilusao-de-optica/>>. Acesso em: 11 maio 2018. Blog: FOCUS, Escola de Fotografia.

FORSYTH, David A.; PONCE, Jean. **Computer Vision a modern approach.** New Jersey: Prentice Hall, 2003.

GOMES, Sinésio. **Aula 45 - Anatomia dos braços mecânicos industriais.** [S.l.], 22 mai. 2014. Disponível em: <<http://controleeautomacaoindustrial3.blogspot.com.br/2014/05/aula-125-anatomia-dos-bracos-mecanicos.html>>. Acesso em: 1 maio 2018. Blog: Controle e Automação industrial III.

GONZALEZ, Rafael C.; WOODS, Richard E. **Processamento de imagens digitais.** Tradução de Roberto Marconi Cesar Junior; Luciano da Fontoura Costa. 1ª edição. São Paulo: Edgard Blücher Ltda, 2000.

IGUS Plastics for long life®. **União articulada angular Igubal® de baixo custo.** São Paulo - Jundiaí, [2018?]. Disponível em:

<https://www.igus.com.br/wpck/2469/igubal_Low_cost_Winkelgelenk>. Acesso em: 5 jun. 2018.

JUBELLI, Fabio J; LIMA, Marcelo de; HORTSMANN, Robson L. **Manipulador paralelo em configuração delta**. 95 f. Trabalho de Conclusão de Curso (Tecnólogo em mecânica industrial) – Curso superior de tecnologia em mecânica industrial, Instituto federal de educação ciência e tecnologia de Santa Catarina, Joinville, 2013. Disponível em: <<http://joinville.ifsc.edu.br/~bibliotecajoi/arquivos/tcc/mecind/108459.pdf>>. Acesso em: 31 maio 2018.

LAZZARI, Felipe dos S; OLIVEIRA, Henrique R de; PAIXÃO, Teilor S. de Q. **Desenvolvimento de um robô paralelo tipo delta associado com visão computacional para aplicações pick and place**. 66 f. Trabalho de Conclusão de Curso (Tecnólogo em mecânica industrial) – Curso superior de tecnologia em mecânica industrial, Universidade tecnológica federal do Paraná, Curitiba, 2012. Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/929/1/CT_COMET_2012_1_02.pdf>. Acesso em: 31 maio 2018.

LENSARIN, Ricardo G. **Análise do volume de trabalho de um robô de arquitetura paralelo do tipo Delta**. Dissertação de mestrado. Universidade Federal do Rio Grande Do Sul. Porto Alegre, 2017.

LIENHART, Rainer; MAYDT, Jochen. **An Extended Set of Haar-Like Features for Rapid Object Detection**. IEEE ICIP. Intel Labs, Santa Clara, 2002, p. 900-903.

LIFEHACKER. **The Raspberry Pi Camera Module Gets Upgraded to 8-Megapixels, Still Costs \$25**. KLOSOWSKI, ThOrin. [S.l.], 25 abr. 2016. Disponível em: <<https://lifehacker.com/the-raspberry-pi-camera-module-gets-upgraded-to-8-megap-1772898924>>. Acesso em: 7 jun. 2018. Blog: Life Hacker.

MAIA, Matheus. **A era dos robôs: tecnologia amplia produtividade, transformação educação e salva vidas**. São Paulo, fev. 2017. Disponível em:

<<http://www.portaldaindustria.com.br/agenciacni/noticias/2017/02/a-era-dos-robos-tecnologia-amplia-produtividade-transforma-educacao-e-salva-vidas/>>. Acesso em: 24 mar. 2018.

MECÂNICA INDUSTRIAL. **Solda e tratamentos industriais**. [S.l.], 2018. Disponível em: <<https://www.mecanicaindustrial.com.br/solda-e-tratamentos-superficiais/>>. Acesso em: 1 maio 2018. Blog: Mecânica Industrial.

MOUSSA, Simhon. **Robótica Industrial**. São Paulo: 1ª Edição, 2011.

MUNDO ESTRANHO. **Como funciona o foco automático das câmeras fotográficas?** São Paulo, 18 março 2011, Editora abril. Disponível em: <<https://mundoestranho.abril.com.br/tecnologia/como-funciona-o-foco-automatico-das-cameras-fotograficas/>> Acesso em: 12 abr. 2018.

NEON CÂMARA. **Tipos de lentes fotográficas, suas funções e aplicações**. 13 Agosto 2013. Disponível em: <<http://www.cameraneon.com/acessorios/tipos-de-lentes-fotograficas-suas-funcoes-e-aplicacoes/>> Acesso em: 12 abr. 2018.

NERYS, José W. L. **Microcontroladores e Microprocessadores**. Notas de aula. Goiás, Universidade Federal de Goiás, [2018?]. Disponível em: <https://inf.ufes.br/~zegonc/material/Introducao_a_Computacao/Microprocessadores%20-%20Parte%201.pdf>. Acesso em: 15 maio 2018.

ProMotion. **Entenda como o robô delta pode revolucionar a produção na indústria**. Santa Catarina, 2017. Disponível em: <<http://www.group-promotion.com/entenda-como-o-robo-delta-pode-revolucionar-a-producao-na-industria/>>. Acesso em: 15 abr. 2018.

RHYDOLABZ. **An Arduino-RPi 2 Interface using Serial Communication**. [S.l.], 26 maio 2015. Disponível em: <<http://www.rhydolabz.com/wiki/?p=6101>>. Acesso em: 7 jun. 2018. Blog: Rhydolabz.

ROBOTICS delta. **Inverse Kinematics of Parallel Manipulator (Delta Robot)**. [S.I.], jul. 2014. Disponível em: <<https://robotics.stackexchange.com/questions/3144/inverse-kinematics-of-parallel-manipulator-delta-robot>>. Acesso em: 1 maio 2018.

SANTACATARINA, Moisés A. **Projeto e construção de um robô paralelo tipo delta controlador por arduino**. 55 f. Trabalho de Conclusão de Curso (Bacharel em engenharia mecânica) – Curso de Engenharia Mecânica, Faculdade de Horizontina (FAHOR), Horizontina, 2015. Disponível em: <<http://www.fahor.com.br/publicacoes/TFC/EngMec/2015/MoisesAugustoSantacatarina.pdf>>. Acesso em: 31 maio 2018.

SANTOS, Rafael. **Processamento de imagens digitais** Notas de aula. São Paulo, INPE (Instituto Nacional de Pesquisa Espaciais), jul. 2009. Disponível em: <<http://www.lac.inpe.br/~rafael.santos/Docs/CursoInverno/2009/Proclmagens-CI2009.pdf>>. Acesso em: 19 maio 2018.

SENSOR SENSORES INOVADORES. **Sistemas de visão industrial**. [S.I.], [2018?]. Disponível em: <<http://www.sensordobrasil.com.br/sistema-visao-industrial>>. Acesso em: 1 maio 2018.

SG90 9g Micro servo. Manual. [S.I.], [2018?]. Disponível em: <<http://akizukidenshi.com/download/ds/towerpro/SG90.pdf>>. Acesso em: 2 mai. 2018.

SIKU, Saeed B. **Introdução à Robótica** Análise, controle, aplicações. Tradução e Revisão Técnica: Sérgio Gilberto Taboada. 2ª edição. Rio de Janeiro: LTC, 2013.

SILVEIRA, Cristiano B. **Servo Motor**: veja como funciona e quais os tipos. [S.I.], [2018?]. Disponível em: <<https://www.citisystems.com.br/servo-motor/>>. Acesso em: 2 mai. 2018. Blog: Citisystems.

TECHTUDO. **Como funciona a raspberry pi? Entenda a tecnologia e sua aplicabilidade**. Filipe Garretti. [S.I.], 20 nov. 2014. Disponível em:

<<http://www.techtudo.com.br/noticias/noticia/2014/11/como-funciona-o-raspberry-pi-entenda-tecnologia-e-sua-aplicabilidade.html>>. Acesso em: 16 maio 2018. Blog: TechTudo.

TECHTUDO. **Entenda o formato de arquivos de imagens**. Claudia Sardinha. [S.l.], 5 jul. 2012. Disponível em: <<http://www.techtudo.com.br/artigos/noticia/2012/07/entenda-os-formatos-dos-arquivos-de-imagem.html>>. Acesso em: 16 maio 2018. Blog: TechTudo.

TEKKNO Mecatrônica. **Estrutura Frame para 3D printer e afins**. São Paulo - Bauru, [2018?]. Disponível em: <<https://www.tekkno.com.br/produto/3922/Estrutura-Frame-para-3D-Printer-e-Afins>>. Acesso em: 5 jun. 2018.

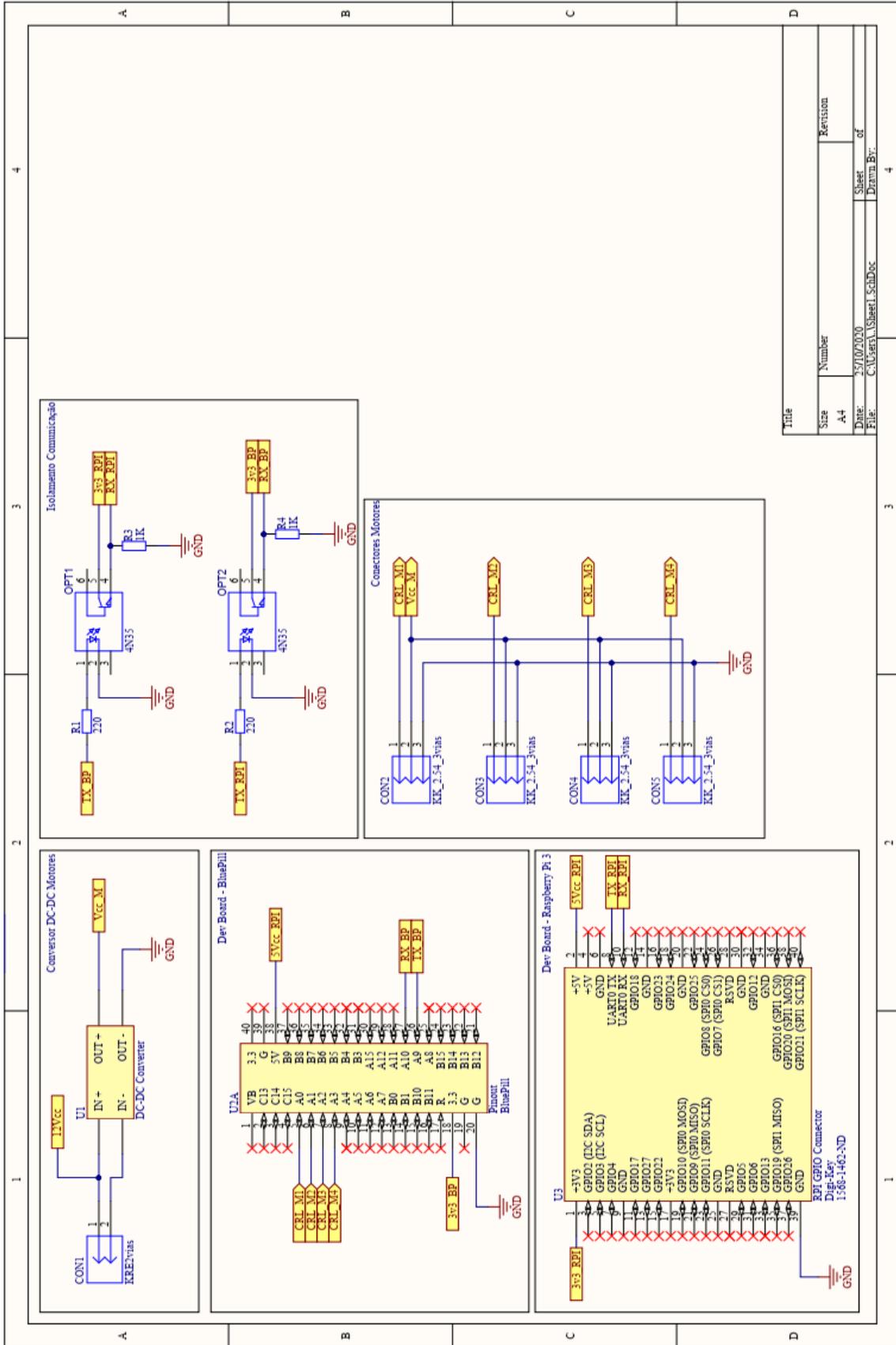
TIRANOS DAS CORES. **Modelo RGB**. André: Eduardo. [S.l.], 18 fev. 2011. Disponível em: <<http://tiranosdascors.blogspot.com.br/2011/02/modelo-rgb.html>>. Acesso em: 24 maio 2018. Blog: Tiranos das Cores.

VESTIBULAR. **A decomposição da luz branca e suas cores originárias**. [S.l.], [2018?]. Disponível em: <<https://www.vestibular.com.br/dica/a-decomposicao-da-luz-branca-e-suas-cores-originarias/>>. Acesso em: 24 maio 2018. Blog: Vestibular - Física.

VIOLA, Paul; JONES, Michael. **Rapid Object Detection Using a Boosted Cascade of Simple Feature**. IEEE. Accepted Conference On ComputerVision and Pattern Recognition. Cambridge, 2001.

YAN, Liang; LIU, Delong; JIAO, Zongxia. **Novel design and kinematics modeling for delta robot with improved end effector**. School of automation science and electrical Engineering. Beihang University, 2016.

APÊNDICE A – DIAGRAMA ESQUEMÁTICO



APÊNDICE B – TABELA DE ERROS DE DETECÇÃO

X Real (mm)	Y real (mm)	Média X Medido (mm)	Média Y Medido (mm)	Erro X (mm)	Erro Y (mm)	Erro abs (mm)
25	25	26	26	1	1	1,41
25	75	22	75	-3	0	3,00
25	125	24	123	-1	-2	2,24
25	175	26	174	1	-1	1,41
25	225	26	223	1	-2	2,24
25	275	26	274	1	-1	1,41
25	325	25	325	0	0	0,00
25	375	26	375	1	0	1,00
75	25	75	24	0	-1	1,00
75	75	78	73	3	-2	3,61
75	125	75	122	0	-3	3,00
75	175	75	173	0	-2	2,00
75	225	75	223	0	-2	2,00
75	275	75	271	0	-4	4,00
75	325	75	322	0	-3	3,00
75	375	73	371	-2	-4	4,47
125	25	127	24	2	-1	2,24
125	75	126	73	1	-2	2,24
125	125	126	122	1	-3	3,16
125	175	125	172	0	-3	3,00
125	225	127	222	2	-3	3,61
125	275	126	272	1	-3	3,16
125	325	123	321	-2	-4	4,47
125	375	125	374	0	-1	1,00
175	25	173	24	-2	-1	2,24
175	75	175	75	0	0	0,00
175	125	177	123	2	-2	2,83
175	175	175	172	0	-3	3,00
175	225	175	221	0	-4	4,00
175	275	176	272	1	-3	3,16
175	325	173	322	-2	-3	3,61
175	375	173	373	-2	-2	2,83
225	25	223	24	-2	-1	2,24
225	75	226	72	1	-3	3,16
225	125	226	122	1	-3	3,16
225	175	225	171	0	-4	4,00
225	225	226	221	1	-4	4,12
225	275	225	271	0	-4	4,00
225	325	225	325	0	0	0,00
225	375	225	373	0	-2	2,00

275	25	274	24	-1	-1	1,41
275	75	274	72	-1	-3	3,16
275	125	275	123	0	-2	2,00
275	175	274	171	-1	-4	4,12
275	225	274	221	-1	-4	4,12
275	275	276	272	1	-3	3,16
275	325	276	321	1	-4	4,12
275	375	276	373	1	-2	2,24
325	25	327	24	2	-1	2,24
325	75	324	74	-1	-1	1,41
325	125	325	122	0	-3	3,00
325	175	324	171	-1	-4	4,12
325	225	325	221	0	-4	4,00
325	275	327	271	2	-4	4,47
325	325	325	322	0	-3	3,00
325	375	325	372	0	-3	3,00
375	25	378	25	3	0	3,00
375	75	376	71	1	-4	4,12
375	125	376	123	1	-2	2,24
375	175	374	172	-1	-3	3,16
375	225	372	223	-3	-2	3,61
375	275	375	273	0	-2	2,00
375	325	375	322	0	-3	3,00
375	375	372	373	-3	-2	3,61

APÊNDICE C – TABELA DE ERROS DE POSICIONAMENTO EFETUADOR

Calibração posição Delta						
SetPoint x (mm)	SetPoint y (mm)	Real x (mm)	Real y (mm)	Erro x (mm)	Erro y (mm)	Erro Absoluto(mm)
75	75	82	86	-7	-11	13,0
75	125	87	116	-12	9	15,0
75	175	77	182	-2	-7	7,3
75	225	85	215	-10	10	14,1
75	275	77	285	-2	-10	10,2
75	325	87	320	-12	5	13,0
75	375	80	366	-5	9	10,3
125	75	118	77	7	-2	7,3
125	125	117	120	8	5	9,4
125	175	116	185	9	-10	13,5
125	225	117	218	8	7	10,6
125	275	120	288	5	-13	13,9
125	325	118	319	7	6	9,2
125	375	120	370	5	5	7,1
175	75	185	86	-10	-11	14,9
175	125	184	120	-9	5	10,3
175	175	186	186	-11	-11	15,6
175	225	184	220	-9	5	10,3
175	275	184	284	-9	-9	12,7
175	325	189	339	-14	-14	19,8
175	375	180	380	-5	-5	7,1
225	75	219	84	6	-9	10,8
225	125	223	119	2	6	6,3
225	175	221	185	4	-10	10,8
225	225	220	220	5	5	7,1
225	275	219	289	6	-14	15,2
225	325	217	314	8	11	13,6
225	375	218	368	7	7	9,9
275	75	268	84	7	-9	11,4
275	125	268	134	7	-9	11,4
275	175	269	184	6	-9	10,8
275	225	273	237	2	-12	12,2
275	275	270	289	5	-14	14,9
275	325	271	327	4	-2	4,5
275	375	274	368	1	7	7,1
325	75	319	86	6	-11	12,5
325	125	321	137	4	-12	12,6
325	175	320	186	5	-11	12,1
325	225	319	237	6	-12	13,4

325	275	319	282	6	-7	9,2
325	325	317	337	8	-12	14,4
325	375	317	388	8	-13	15,3
375	75	365	87	10	-12	15,6
375	125	366	138	9	-13	15,8
375	175	367	187	8	-12	14,4
375	225	367	232	8	-7	10,6
375	275	370	282	5	-7	8,6
375	325	369	336	6	-11	12,5
375	375	368	387	7	-12	13,9