



## Pós-Graduação em Robótica e Sistemas Autônomos

### Relatório Final do Programa de Formação em Robótica e Sistemas Autônomos.

Apresentada por: Jean Paulo Silva

Dezembro de 2020



Jean Paulo Silva

**Relatório Final do Programa de Formação em  
Robótica e Sistemas Autônomos.**

Salvador  
Centro Universitário SENAI CIMATEC  
2020

---

## Agradecimentos

---

Agradeço a todos os orientadores e pesquisadores do Centro de Competência em Sistemas Autônomos pelo suporte e por serem intemediadores do conhecimento durante todo o processo da formação. Agradeço ao SENAI CIMATEC por fomentar o programa Novos Talentos 2020 e a Especialização em Robótica e Sistemas Autônomos. Por fim, agradeço aos professores das disciplinas da Especialização pela didática empregada durante as aulas.

Salvador, Brasil  
30 de Dezembro de 2020

Jean Paulo Silva

---

## Resumo

---

Este relatório foca em trazer os resultados obtidos de estudos e projetos realizados durante a pós-graduação em Especialização em Robótica e Sistemas Autônomos. Este documento trás os objetivos, metodologia e os resultados de cinco desafios e dois estudos estatíticos realizados durante o período como bolsista do programa de formação Novos Talentos 2020 realizado pelo SENAI CIMATEC.

**Palavras-chave:** Robótica, Automação, Pesquisa, Inovação, Manipulador

---

## **Abstract**

---

This report focuses on bringing the results obtained from studies and projects carried out during the graduate course in Robotics and Autonomous Systems. This document presents the objectives, methodology and results of five challenges and two statistical studies carried out during the period as a fellow of the Novos Talentos 2020 program by SENAI CIMATEC.

**Keywords:** Robotics, Automation, Research, Innovation, Manipulator

---

# Sumário

---

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetivos . . . . .	1
1.2	Justificativa . . . . .	1
1.3	Organização do documento . . . . .	1
<b>2</b>	<b>Desenvolvimento</b>	<b>3</b>
2.1	Desafio 1.0 - Simulação do ambiente externo do CIMATEC . . . . .	3
2.2	Desafio 2.0 - Entrega parcial: simulação do manipulador robótico RAJA . . . . .	5
2.3	Desafio 2.2 - Entrega final do manipulador JeROTIMON . . . . .	6
2.4	Desafio 2.5 - Análise estatística R&R do robô Darwin-OP . . . . .	7
2.5	Desafio 3.0 - Curupira . . . . .	10
2.6	Estudo estatístico de <i>DOE (Design of Experiments)</i> . . . . .	12
<b>3</b>	<b>Metodologia</b>	<b>13</b>
<b>4</b>	<b>Resultados</b>	<b>15</b>
4.1	Resultado do Artigo RAJA: MANIPULADOR ROBÓTICO DE 5 DOF COM DETECÇÃO VISUAL INTEGRADA . . . . .	15
4.2	Resultado do Artigo PERSPECTIVES ON AUTONOMOUS UNMANNED GROUND VEHICLES: A SURVEY . . . . .	15
<b>5</b>	<b>Conclusão</b>	<b>16</b>
<b>A</b>	<b>MANIPULADOR ROBÓTICO RAJA</b>	<b>17</b>
<b>B</b>	<b>MANIPULADOR ROBÓTICO JEROTIMON</b>	<b>84</b>
<b>C</b>	<b>Avaliação de medição para o desafio 2.5</b>	<b>219</b>
<b>D</b>	<b>SISTEMA PARA IDENTIFICAÇÃO E MANIPULAÇÃO DE OBJETOS EM AMBIENTES EXTERNOS UTILIZANDO PLATAFORMA MÓVEL</b>	<b>225</b>
<b>E</b>	<b>Estudo estatístico - Análise de Regressão</b>	<b>341</b>
<b>F</b>	<b>RAJA: MANIPULADOR ROBÓTICO DE 5 DOF COM DETECÇÃO VISUAL INTEGRADA</b>	<b>361</b>
<b>G</b>	<b>Certificado de participação no V SAPCT</b>	<b>365</b>
<b>H</b>	<b>PERSPECTIVES ON AUTONOMOUS UNMANNED GROUND VEHICLES: A SURVEY</b>	<b>367</b>
<b>I</b>	<b>Certificado de participação no VI SIINTEC</b>	<b>376</b>

---

## Lista de Figuras

---

2.1	Vista superior da área externa do CIMATEC 4 . . . . .	3
2.2	Representação em <i>CAD</i> utilizando o <i>software OnShape</i> . . . . .	4
2.3	Robô <i>Clearpath Husky</i> simulado em ambiente <i>Gazebo</i> . . . . .	4
2.4	Bola amarela no ambiente de trabalho simulado. . . . .	5
2.5	Manipulador RAJA finalizando a missão. . . . .	6
2.6	Manipulador JeRoTIMON no ambiente de trabalho e a caixa objetivo. . .	7
2.7	<i>Darwin-OP</i> em ambiente simulado. . . . .	8
2.8	<i>Darwin-OP</i> em posição de marcha liderada. . . . .	9
2.9	<i>Darwin-OP</i> em posição de revezamento. . . . .	9
2.10	Pista e suas dimensões. . . . .	10
2.11	“Bomba” identificada pela rede neural no ambiente de trabalho. . . . .	11
2.12	Integração física. . . . .	11
3.1	Metodologia do Programa de Formação em Robótica e Sistemas Autônomos	14

---

## Lista de Siglas

---

<i>ArUco</i> .....	<i>Augmented Reality by University of Cordoba</i>
<i>CAD</i> .....	<i>Computer Aided Design</i>
<i>DOE</i> .....	<i>Design of Experiments</i>
<i>DOF</i> .....	<i>Degrees of Freedom</i>
<i>GPS</i> .....	<i>Global Positioning System</i>
<i>ICPAD</i> .....	Integração e Capacitação em Processamento de Alto Desempenho
<i>LIDAR</i> .....	<i>Light Detection and Ranging</i>
<i>PD&amp;I</i> .....	Pesquisa, Desenvolvimento e Inovação
<i>DOF</i> .....	<i>Degrees of Freedom</i>
<i>RGB</i> .....	<i>Red, Green and Blue</i>
<i>SAPCT</i> .....	Seminário de Avaliação de Pesquisa Científica e Tecnologia
<i>SIINTEC</i> ...	Simpósio Internacional de Inovação e Tecnologia
<i>UGV</i> .....	<i>Unmanned Ground Vehicle</i>

---

## Introdução

---

Neste capítulo estarão expostos os objetivos e a justificativa do programa de formação e os conhecimentos adquiridos durante o desenvolvimento dos desafios.

### 1.1 *Objetivos*

Este relatório tem por objetivo mostrar todos os desafios e trabalhos desenvolvidos durante o programa de formação Novos Talentos 2020 em Robótica e Sistemas Autônomos e demonstrar a estrutura do curso de especialização e os conhecimentos adquiridos durante o processo.

### 1.2 *Justificativa*

Este programa de formação profissional e de pós-graduação buscou formar especialistas em robótica que possam atuar futuramente em projetos industriais e de desenvolvimento. O programa trouxe uma metodologia de aprendizagem ativa para o bolsista, onde este estava realizando constantes desafios simulados e práticos, utilizando de tecnologias recentes na área de automação, robótica e visão computacional para o desenvolvimento de habilidades e competências necessárias para futuros trabalhos.

### 1.3 *Organização do documento*

Este relatório possui 5 capítulos e é estruturado da seguinte forma:

- **Capítulo 1 - Introdução:** Em que são descritos os objetivos gerais, a justificativa e a organização dos documentos
- **Capítulo 2 - Desenvolvimento:** Estão descritos os projetos realizados durante a graduação, consistindo em desafios simulados, desafios físicos e estudos estatísticos.
- **Capítulo 3 - Metodologia:** Está descrita a metodologia empregada durante a formação em Robótica e Sistemas Autônomos.

- **Capítulo 4 - Resultados:** Resultados apresentados para os desafios e artigos empregados.
- **Capítulo 5 - Conclusão:** Considerações finais em relação aos projetos e ao o programa de graduação.

## Desenvolvimento

Neste capítulo serão demonstrados os desafios e trabalhos realizados durante o programa de formação em Robótica e Sistemas Autônomos.

### 2.1 Desafio 1.0 - Simulação do ambiente externo do CIMATEC

4

Este desafio foi realizado de maneira individual, e consiste na simulação de um robô UGV (*Unmanned Ground Vehicle*) da *Clearpath Robotics* chamado *Husky* (Figura 2.3), no ambiente simulado *Gazebo*. O ambiente de trabalho é uma representação em 3D *CAD* (*Computer Aided Design*) da área externa do CIMATEC 4 (Figura 2.1), como visto na Figura 2.2. Sua missão era achar uma bola amarela de 1 metro de raio (Figura 2.4) que possuía localização aleatória no ambiente de trabalho. Durante a missão, o robô deve navegar autonomamente e realizar a detecção da bola via câmera. O objetivo da missão era concluído quando o robô se aproximasse até 1 metro da bola e enviasse uma mensagem de conclusão.

Figura 2.1: Vista superior da área externa do CIMATEC 4.



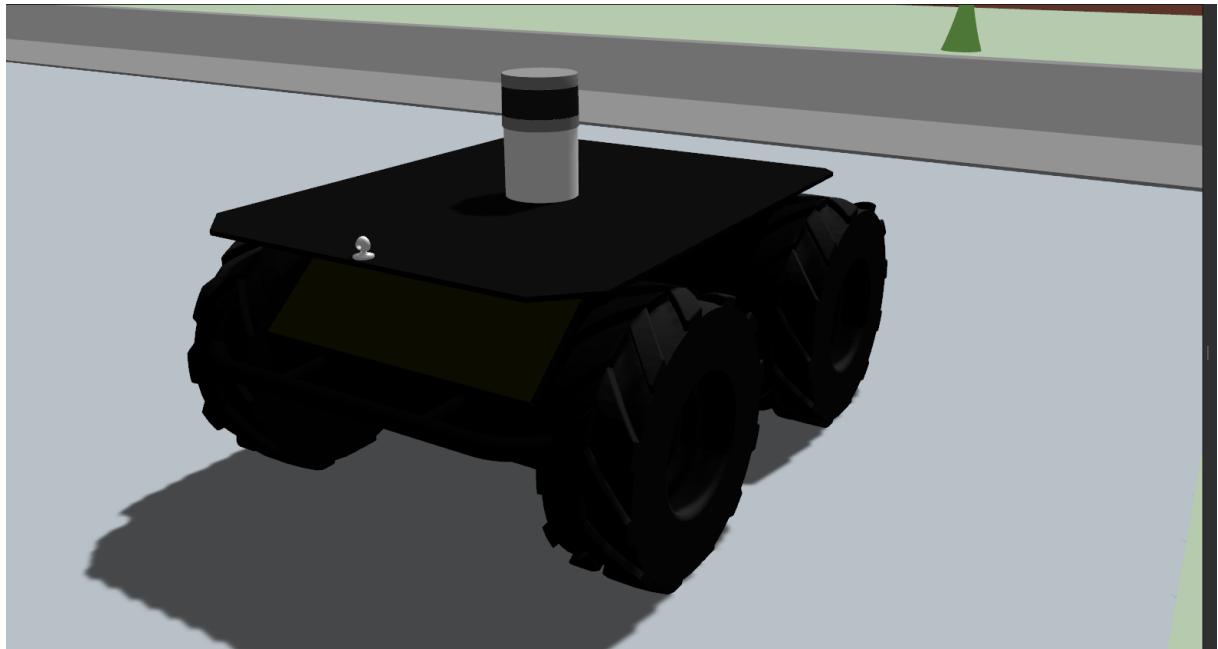
Fonte: *Google Earth*

Figura 2.2: Representação em *CAD* utilizando o software *OnShape*.



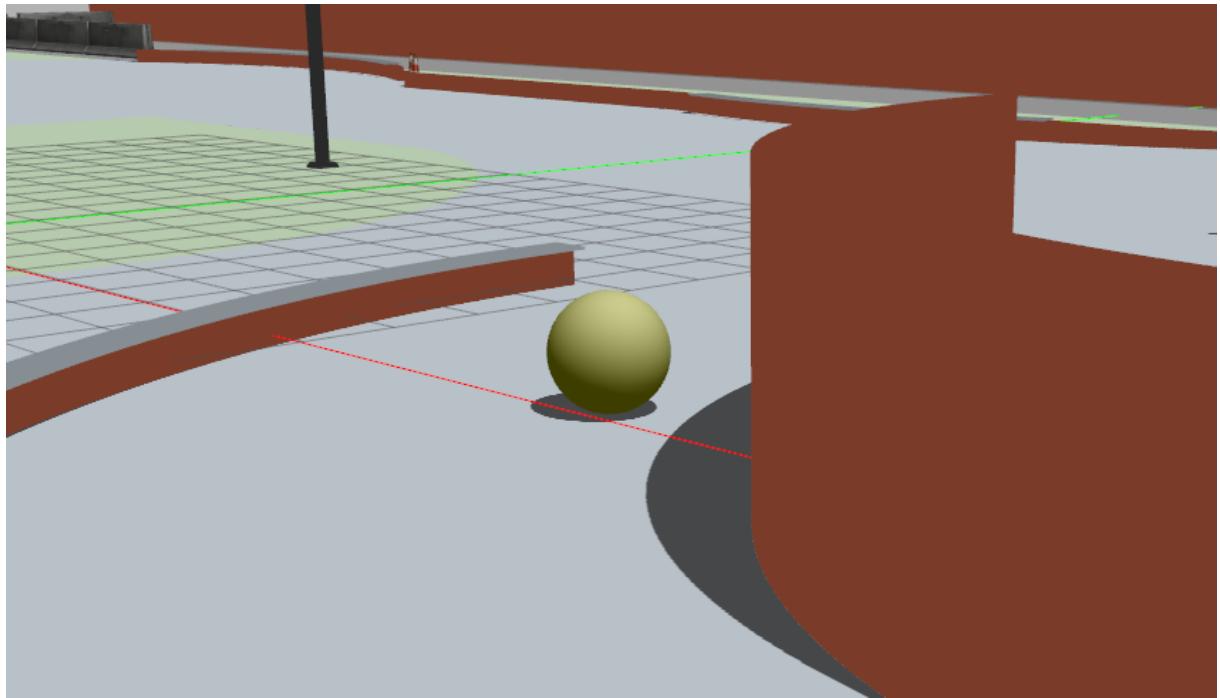
Fonte: Grupo de formação em Robótica e Sistemas Autônomos

Figura 2.3: Robô *Clearpath Husky* simulado em ambiente *Gazebo*.



Fonte: Grupo de formação em Robótica e Sistemas Autônomos

Figura 2.4: Bola amarela no ambiente de trabalho simulado.

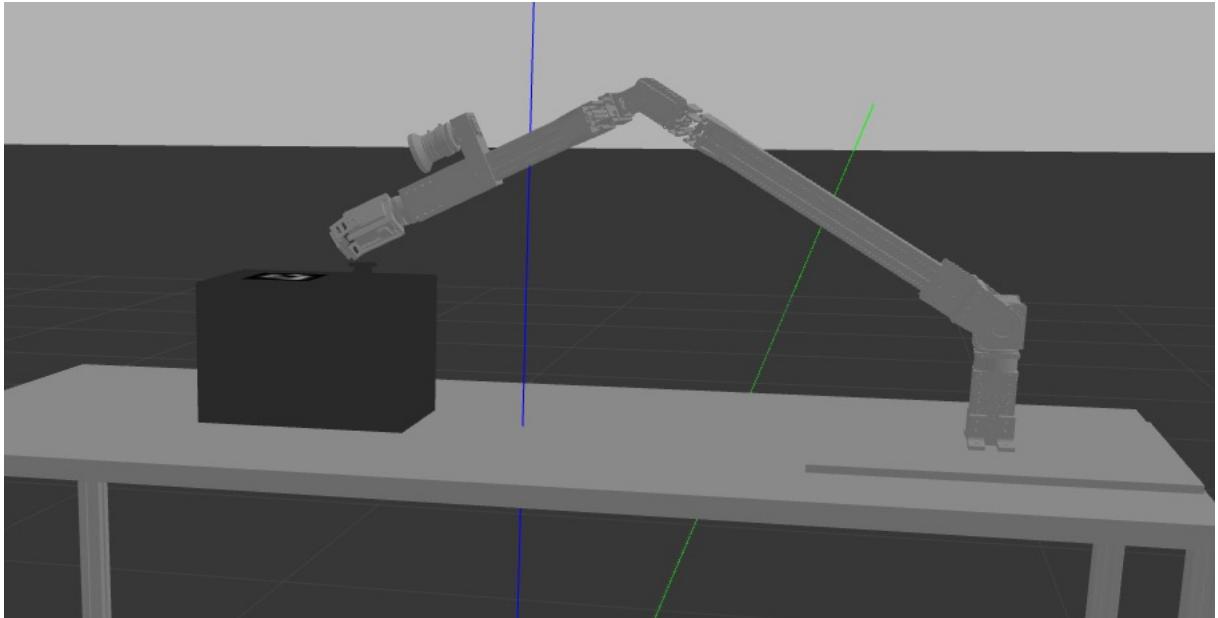


Fonte: Grupo de formação em Robótica e Sistemas Autônomos

## 2.2 Desafio 2.0 - Entrega parcial: simulação do manipulador robótico RAJA

Este desafio foi realizado em grupo de 4 pessoas e consistiu na simulação em *Gazebo* de um manipulador robótico com 5 graus de liberdade de nome RAJA. Ele possui uma câmera *RGB (Red, Green and Blue)* que tem finalidade em encontrar a posição de um botão de uma caixa no seu ambiente de trabalho. Esta posição é definida por um marcador fiducial *ArUco (Augmented Reality by University of Cordoba)* que a caixa possui. Com a posição encontrada, o manipulador pressiona o botão com sua ferramenta. Todos os modelos simulados foram feitos no *OnShape*. A Figura 2.5 representa o manipulador em seu ambiente de trabalho ao chegar até o botão. O Apêndice A representa o relatório deste desafio com suas especificações, arquitetura, desenvolvimento, metodologia e resultados.

Figura 2.5: Manipulador RAJA finalizando a missão.



Fonte: Grupo de formação em Robótica e Sistemas Autônomos

### 2.3 Desafio 2.2 - Entrega final do manipulador JeRoTIMON

Este desafio realizado em um grupo de 6 pessoas, visou construir um modelo físico de um manipulador robótico baseado nos conceitos aplicados no desafio 2.0 (Subseção 2.2). Os objetivos continuaram os mesmos: obter a posição de um botão em uma caixa baseado em um marcador fiducial *ArUco* utilizando uma câmera *RGB* modelo *Teledyne Genie Nano C2590* e pressioná-lo utilizando o manipulador robótico. A construção física do manipulador foi realizada com os materiais disponíveis no laboratório de Robótica e Sistemas Autônomos do SENAI CIMATEC. Sua base foi feita de madeira com suportes de aço, e os elos foram feitos com alumínio. Os motores são de marca ROBOTIS e modelos Dynamixel PLUS. A peça de suporte para a câmera foi feita a partir de impressão 3D com material ABS. A Figura 2.6 mostra o manipulador em seu ambiente de trabalho com a caixa objetivo. No Apêndice B está disponível o relatório com maiores detalhes sobre o desenvolvimento, metodologia e resultados.

Figura 2.6: Manipulador JeRoTIMON no ambiente de trabalho e a caixa objetivo.



Fonte: Grupo de formação em Robótica e Sistemas Autônomos

## 2.4 Desafio 2.5 - Análise estatística R&R do robô Darwin-OP

Este desafio foi realizado em um grupo de 4 pessoas, as mesmas do desafio 2.2. O objetivo deste trabalho era desenvolver um ambiente simulado em *Gazebo*, e realizar ações autônomas de marcha liderada e corrida de revezamento com 4 robôs humanóides *Darwin-OP* (Figura 2.7). Na marcha liderada, os 4 robôs devem andar em velocidade sincronizada durante 2 metros, como mostra a Figura 2.8. Na corrida de revezamento, cada robô está posicionado em uma parte específica da pista de corrida (Figura 2.9), em que ele deve aguardar o robô anterior se aproximar, manter-se próximo em velocidade sincronizada por cerca de 5 segundos (representando a passagem de bastões) e depois seguir em maior velocidade para alcançar o robô à sua frente. A pista projetada pode ser visualizada na Figura 2.10.

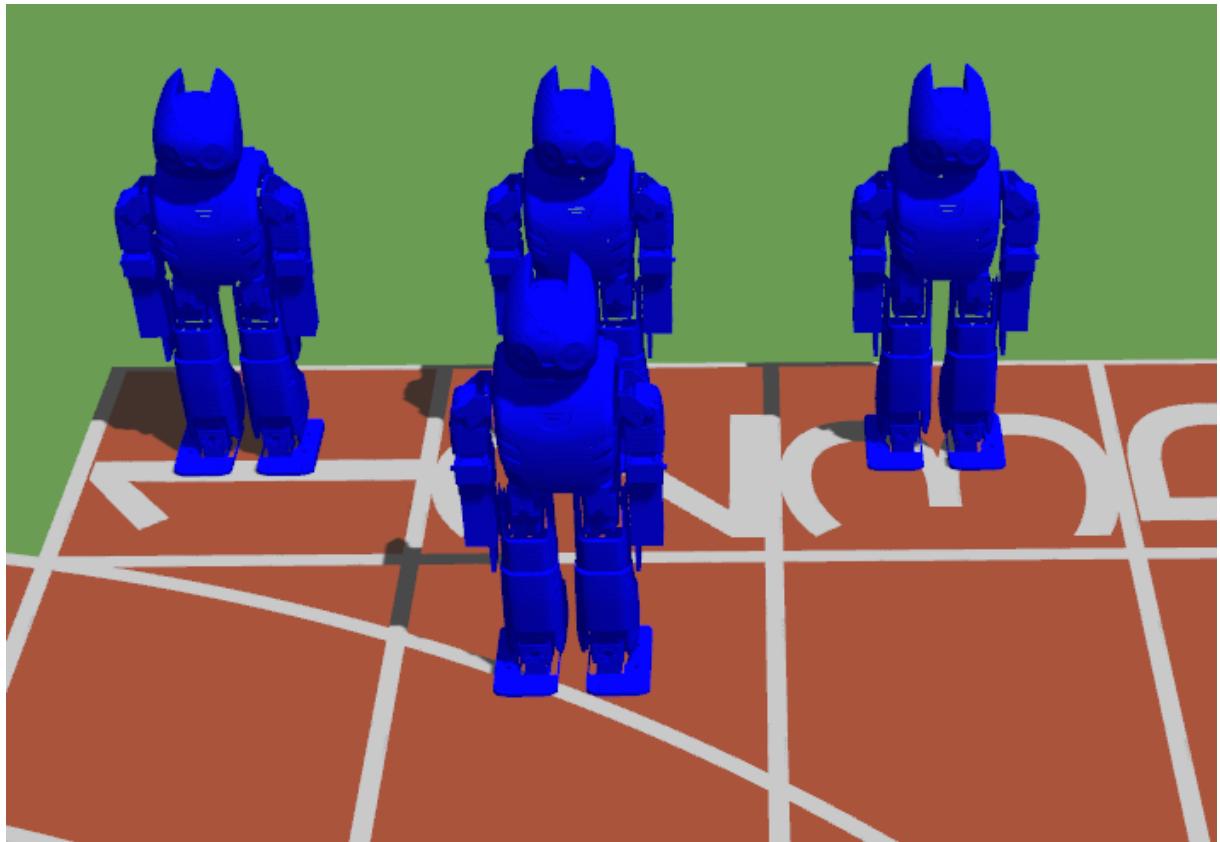
Este projeto teve um estudo estatístico que pode ser visto no Apêndice C, e tem por objetivo analisar a medição dos dados utilizando o método de análise de variância.

Figura 2.7: *Darwin-OP* em ambiente simulado.



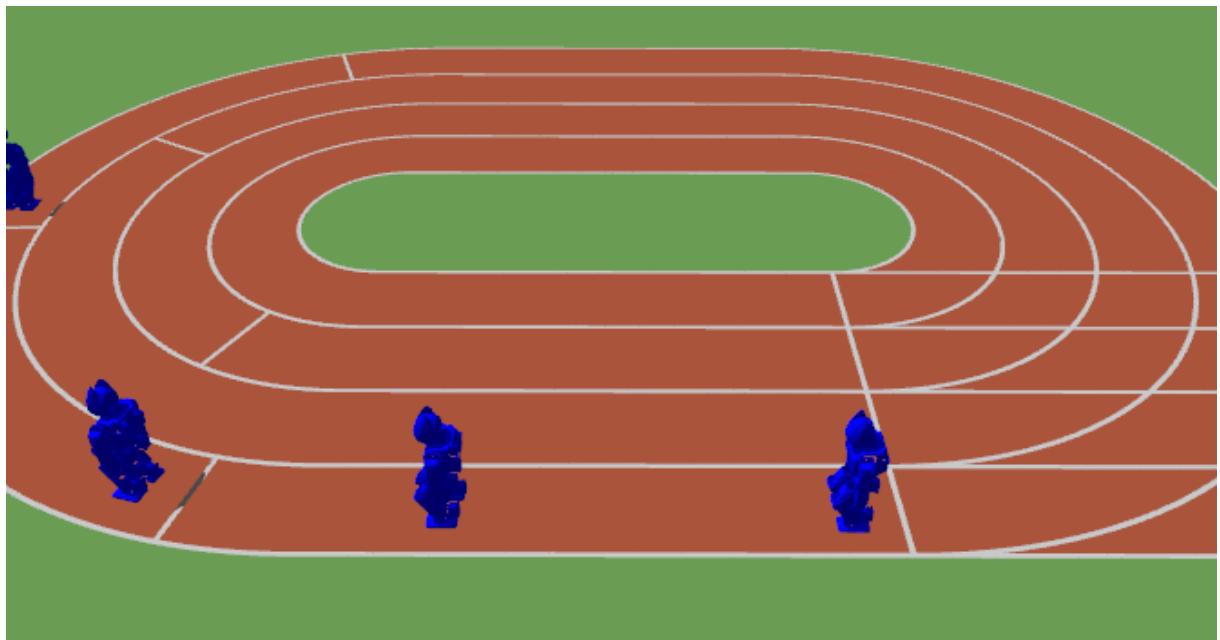
Fonte: Grupo de formação em Robótica e Sistemas Autônomos

Figura 2.8: *Darwin-OP* em posição de marcha liderada.



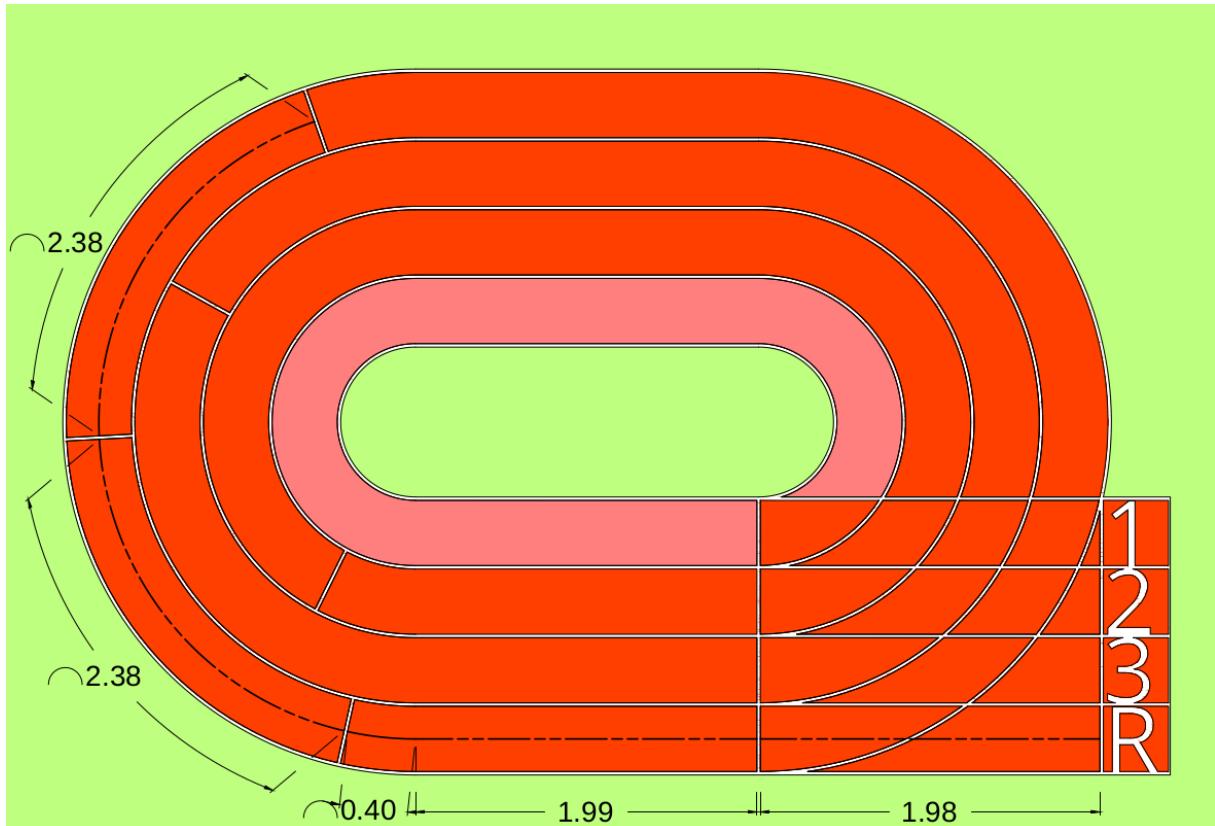
Fonte: Grupo de formação em Robótica e Sistemas Autônomos

Figura 2.9: *Darwin-OP* em posição de revezamento.



Fonte: Grupo de formação em Robótica e Sistemas Autônomos

Figura 2.10: Pista e suas dimensões.



Fonte: Grupo de formação em Robótica e Sistemas Autônomos

## 2.5 Desafio 3.0 - Curupira

Este desafio foi realizado por um grupo de 3 pessoas. O projeto Curupira representa a integração do *UGV Clearpath Robotics Warthog* e o manipulador JeRoTIMON, visto na Subseção 2.3. Equipado com sensores para as funcionalidades de percepção de localização, como câmera *RGB*, *LIDAR* (*Light Detection and Ranging*) e *GPS* (*Global Positioning System*), o objetivo é dar autonomia ao sistema para que ele consiga identificar uma “bomba” (Figura 2.11) presente em um lugar aleatório da área externa ao CIMATEC 4 (visto na Figura 2.1 na Subseção 2.1). A posição do objetivo é adquirida pela câmera, e o objetivo é considerado completo quando o manipulador tocar nos fios da “bomba”.

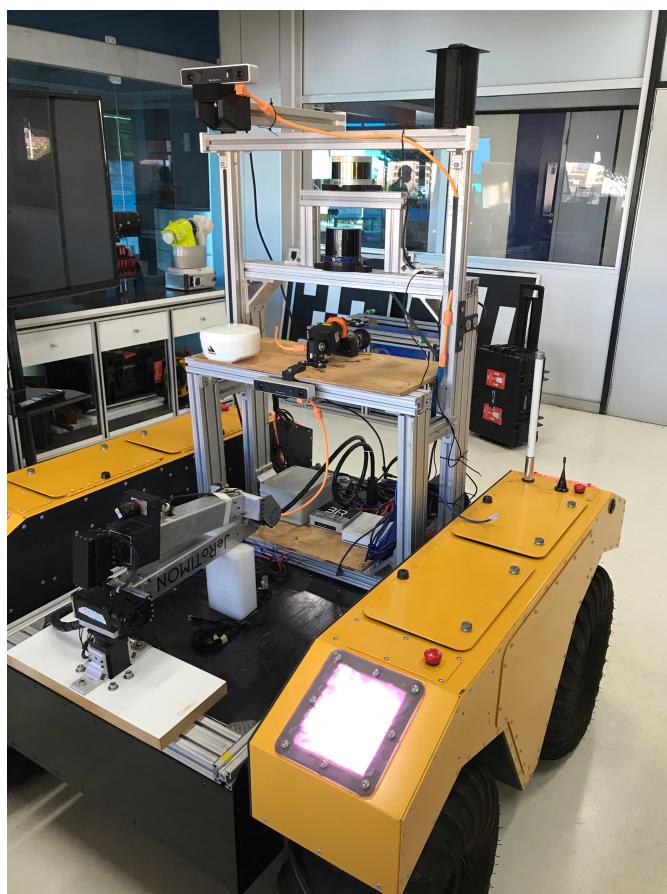
Este projeto foi dividido em duas partes: simulada em ambiente *Gazebo*; e a construção real da integração, visto na Figura 2.12. O desenvolvimento, metodologia e resultados podem ser vistos no Apêndice D.

Figura 2.11: “Bomba” identificada pela rede neural no ambiente de trabalho.



Fonte: Grupo de formação em Robótica e Sistemas Autônomos

Figura 2.12: Integração física.



Fonte: Grupo de formação em Robótica e Sistemas Autônomos

## 2.6 *Estudo estatístico de DOE (Design of Experiments)*

Este desafio foi realizado em um grupo de 4 pessoas, as mesmas do desafio 2.2. O objetivo deste experimento foi aplicar os conceitos de estatística aprendidos durante a especialização. Foi utilizado um modelo de helicóptero de papel em um experimento de medição do seu tempo de queda. Durante o processo de medição, foram adicionados adesivos e clipes ao helicóptero, modificando seu tempo de queda. Os conceitos de *DOE* foram aplicados para identificar quais os fatores mais influenciam a variável controlada: tempo de vôo. Esta análise foi feita utilizando o *software* estatístico *R Studio* para tratamento dos dados e visualização final. Os resultados e interpretações estão no Apêndice E.

## Metodologia

Um dos pontos característico do programa de Formação em Robótica é a sua metodologia, onde buscará a aprendizagem ativa do estudante, com a construção dos seus conhecimentos, complementando as aulas expositivas com atividades e dinâmicas de grupo, elaboração e apresentação de trabalhos e pesquisas, emprego de meios audiovisuais, estudos individualizados, pesquisa de artigos técnicos e científicos, entre outros condicionantes ao programa. A metodologia em si é um caminho para o sucesso na formação dos estudantes, pois esta convergência baseia-se em 5 pontos principais:

1. **Criatividade:** espaço que estimula a criatividade, possibilitando interação e acesso à diferentes tecnologias.
2. **Engajamento:** atividades práticas de aprendizado aumentam os níveis de concentração
3. **Programação:** a inteligência artificial se torna cada vez mais presente nas escolas e escritórios
4. **Trabalho em equipe:** robótica incorpora uma gama de habilidades e promove um ambiente de aprendizagem para pessoas com diferentes talentos
5. **Diversão:** aprender sobre robótica deve ser divertido, e a medida que os estudantes continuem melhorando sua interação com eles, isso aumenta mais ainda o nível de diversão

Basicamente o caminho para o sucesso compreende em 4 fases distintas, demonstradas na Figura 3.1:

**Assimilação:** desenvolver habilidades de codificação e lógica;

**Simulação:** testar as missões de um robô de forma eficiente;

**Integração:** garantir informações sobre o ambiente e o robô;

**Criação:** elaborar um projeto aplicado a tecnologia.

As três primeiras fases são compreendidas em 6 meses, ficando a fase Criação com 6 meses para finalizar o programa. Entre cada fase, desafios são lançados para que o estudante

obtenha maior sucesso na assimilação dos conceitos ministrados, para a última fase um projeto final é lançado para que o estudante possa realizar a demonstração de seu sistema para os avaliadores. Bom salientar que durante as fases, temas serão tratados e discutidos de forma expositiva e prática.

Figura 3.1: Metodologia do Programa de Formação em Robótica e Sistemas Autônomos



Fonte: Programa de Formação em Robótica e Sistemas Autônomos

Com uma abordagem inovadora, o programa tenta realçar a busca por um aprendizado mais real e excitante para isso o conceito de professor facilitador que estimule a experiência nas várias tecnologias se faz necessário. Isso promoverá vários feedbacks mais intensos aos estudantes. Em resumo o professor será o facilitador de uma experiência de aprendizado, criando recursos e experiência para a formação do aprendiz.

---

## Resultados

---

Neste capítulo são expostos os resultados provenientes dos artigos desenvolvidos baseados nos desafios e entregas que foram expostos nos eventos SAPCT (Seminário de Avaliação de Pesquisa Científica e Tecnologia) e SIINTEC (Simpósio Internacional de Inovação e Tecnologia) de 2020.

### **4.1 Resultado do Artigo RAJA: MANIPULADOR ROBÓTICO DE 5 DOF COM DETECÇÃO VISUAL INTEGRADA**

Para o V SAPCT e IV ICPAD (Integração e Capacitação em Processamento de Alto Desempenho), o artigo baseado no projeto do manipulador robótico RAJA, visto no Apêndice F, foi enviado e aceito pelo evento. Este projeto recebeu o prêmio de melhor trabalho na categoria PD&I (Pesquisa, Desenvolvimento e Inovação), e o certificado pode ser visto no Apêndice G.

### **4.2 Resultado do Artigo PERSPECTIVES ON AUTONOMOUS UNMANNED GROUND VEHICLES: A SURVEY**

Este resultado foi baseado no projeto conceitual de dois UGVs. Os veículos possuem dois portes: o de porte médio, de nome Capivara; e outro de porte pequeno, de nome Mocó - ambos homenageando a fauna brasileira. Ambos possuem a finalidade de aspersão de ambientes, tendo em vista o contexto da pandemia causada pelo Coronavírus. O Seminário foi apresentado no evento SIINTEC 2020. Seu artigo pode ser visto no Apêndice H e a certificação gerada pode ser visualizada no Apêndice I.

## Conclusão

---

O programa de formação Novos Talentos 2020 em Robótica e Sistemas Autônomos possibilitou a aprendizagem de habilidades e competências requeridas para trabalhar em projetos de robótica. Isso foi possível graças à metodologia empregada, que foca em fazer com que o bolsista aprenda os conhecimentos necessários por atividades práticas de maneira contínua e gradual. A metodologia trabalhada durante o programa contribuiu para uma maneira fluída de assimilamento de informações e os resultados apresentados no documento confirmam sua eficácia. A presença imediata de vários orientadores especialistas também é responsável por grande parte dos resultados obtidos durante a formação. Esta abordagem necessita de estímulos, correções e avaliações constantes; características presentes durante todo o processo.

Além dos conteúdos técnicos passados, houveram também ensinamentos sobre competências esperadas de um pesquisador trabalhando em projetos, como liderança, planejamento, trabalho em equipe, criatividade e gerenciamento. Os desafios e trabalhos desenvolvidos durante a formação contribuíram para a publicação de artigos e a participação de seminários, resultados fundamentais para a formação de um pesquisador, tornando o processo de formação extremamente gratificante e motivador.

---

## MANIPULADOR ROBÓTICO RAJA

---



## **MANIPULADOR ROBÓTICO COM DETEÇÃO VISUAL INTEGRADA**

Relatório Parcial do Projeto RAJA

### **Autores:**

Anderson Queiroz do Vale  
Aziel Martins de Freitas Júnior  
Jean Paulo Silva  
Rodrigo Formiga Farias

### **Facilitadores:**

Rebeca Tourinho Lima  
Marco Antônio dos Reis

**Salvador  
Bahia, Brasil**

**Abril de 2020**



Título: MANIPULADOR ROBÓTICO COM DETECÇÃO VISUAL INTEGRADA	
<b>PROD. TEC. BIR - 001 / 2020</b>	Versão
<b>Classificação:</b> ( ) Confidencial (X) Restrito ( ) Uso Interno ( ) Público	01

**Informações Confidenciais** - Informações estratégicas para o BIR e Senai Cimatec.

Seu manuseio é restrito a usuários previamente autorizados pelo Gestor da área.

**Informações Restritas** - Informação cujo conhecimento, manuseio e controle de acesso devem estar limitados a um grupo restrito de pesquisadores que necessitam utilizá-la para exercer suas atividades profissionais.

**Informações de Uso Interno** - São informações destinadas à utilização interna por pesquisadores e parceiros.

**Informações Públicas** - Informações que podem ser distribuídas ao público externo, o que, usualmente, é feito através dos canais apropriados.

#### Dados Internacionais de Catalogação na Publicação (CIP)

Anderson Queiroz do Vale  
Aziel Martins de Freitas Júnior  
Jean Paulo Silva  
000 Rodrigo Formiga Farias

Rebeca Tourinho Lima  
Marco Antônio dos Reis

MANIPULADOR ROBÓTICO COM DETECÇÃO VISUAL INTEGRADA  
Salvador  
Bahia, Brasil  
Abril de 2020

Keywords:  
1. Manipulator. 2. Simulation. 3. Computer vision.

000



## SUMÁRIO EXECUTIVO

O projeto de Manipuladores - Desafio.2, também conhecido como **RAJA Robotics** se configura sob o Programa de Formação de Novos Talentos do Serviço Nacional de Aprendizagem Industrial, Departamento Regional da Bahia - Senai/DR/BA, sendo este o principal fomentador do programa.

O projeto teve início no dia 27 de fevereiro de 2020 e o prazo de execução foi de 50 dias.

## RESUMO

Este trabalho tem por finalidade desenvolver um manipulador robótico automatizado capaz de localizar um marcador e realizar a ação de pressionamento de um botão. Para tanto, utiliza-se da detecção e identificação de marcadores fiduciais, do estudo da cinemática direta e inversa, a fim de planejar e executar o movimento do manipulador, e do estudo da ferramenta *MoveIt* a fim de conseguir alcançar um ponto de interesse.

## ABSTRACT

This work aims to develop an automated robotic manipulator capable of locating a marker and performing the action of pressing a button. For that, it uses the detection and identification of fiducial markers, the study of direct and inverse kinematics, in order to plan and execute the manipulator's movement, and the study of the MoveIt tool in order to reach a point of interest.

## LISTA DE FIGURAS

Figura 1:	Volume de trabalho de robô cartesiano.	16
Figura 2:	Volume de trabalho de robô de coordenadas cilíndricas.	16
Figura 3:	Áreas de trabalho de robô de coordenadas esféricas.	17
Figura 4:	Volume de trabalho de robô SCARA.	17
Figura 5:	Volume de trabalho de robô antropomórfico.	18
Figura 6:	Modelagem utilizando matrizes homogêneas e a notação de D-H.	20
Figura 7:	Ambiente de operação simulado.	23
Figura 8:	Representação visual do sistema.	26
Figura 9:	Visão geral dos elementos do sistema.	27
Figura 10:	Estrutura dos elos.	27
Figura 11:	Juntas do manipulador.	28
Figura 12:	Atuadores Dynamixel.	29
Figura 13:	Teledyne Genie Nano C2590.	30
Figura 14:	Área de trabalho projetada no plano XZ.	30
Figura 15:	Área de trabalho projetada no plano YZ.	31
Figura 16:	Área de trabalho projetada no plano XY.	31
Figura 17:	Estrutura analítica do protótipo.	32
Figura 18:	Fluxograma do sistema.	33
Figura 19:	Detalhamento da funcionalidade de busca.	33
Figura 20:	Detalhamento da funcionalidade de aproximação.	34
Figura 21:	Manipulador em curso ao objetivo - caixa horizontal.	38
Figura 22:	Manipulador em curso ao objetivo - caixa vertical.	38
Figura 23:	Histograma e densidade das amostras da movimentação da posição <i>Home</i> até identificação da <i>tag</i> - caixa horizontal.	40
Figura 24:	Histograma e densidade das amostras da movimentação do fim do ciclo de escaneamento até o botão - caixa horizontal.	40
Figura 25:	Histograma e densidade das amostras da movimentação da posição <i>Home</i> até identificação da <i>tag</i> - caixa vertical.	41
Figura 26:	Histograma e densidade das amostras da movimentação do fim do ciclo de escaneamento até o botão - caixa vertical.	41

## LISTA DE TABELAS

Tabela 1:	Análise de esforços das juntas. . . . .	28
Tabela 2:	Especificações do fabricante para os atuadores. . . . .	29
Tabela 3:	Testes do manipulador em curso ao objetivo - caixa horizontal. . . . .	37
Tabela 4:	Testes do manipulador em curso ao objetivo - caixa vertical. . . . .	39
Tabela 5:	Classificação de severidade de falhas. . . . .	43
Tabela 6:	Classificação de facilidade de detecção de falhas. . . . .	44
Tabela 7:	Classificação da probabilidade de ocorrência de falhas. . . . .	44
Tabela 8:	Matriz de modos e efeitos de falhas para os subsistemas do manipulador	45
Tabela 9:	Condensação de eventos marcantes do projeto. . . . .	47



## LISTRA DE SÍMBOLOS E ABREVIATURAS

**AR** *Augmented Reality*

**ARUCO** *Augmented Reality University of Cordoba*

**CAD** *Computer Aided Design*

**DC** *Direct Current*

**GUI** *Graphic User Interface*

**PWM** *Pulse Width Modulation*

**QR** *Quick Response*

**ROS** *Robot Operating System*

**RViz** *ROS Visualizer*

**SCARA** *Selective Compliance Articulated Robot Arm*

**URDF** *Universal Robotic Description Format*

**XML** *Extensible Markup Language*

**D-H** *Denavit-Hartenberg*

**OpenCV** *Open Source Computer Vision Library*

**XACRO** *XML macros*

**FMECA** *Failure Mode and Effect Criticality Analysis*

# SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>11</b>
1.1 Objetivos . . . . .	12
1.2 Justificativa . . . . .	12
1.3 Organização do relatório . . . . .	12
<b>2 CONCEITO DO SISTEMA</b>	<b>13</b>
2.1 Parâmetros básicos . . . . .	13
2.1.1 Manipulador mecânico . . . . .	13
2.1.2 Atuadores . . . . .	13
2.1.3 Sensores . . . . .	14
2.1.4 Unidade de controle . . . . .	14
2.1.5 Unidade de potência . . . . .	14
2.1.6 Efetuador . . . . .	14
2.2 Requisitos do cliente . . . . .	14
2.3 Requisitos técnicos . . . . .	15
2.4 Fundamentação Teórica . . . . .	15
2.4.1 Robô Manipulador . . . . .	15
2.4.1.1 Robô de coordenadas cartesianas . . . . .	16
2.4.1.2 Robô de coordenadas cilíndricas . . . . .	16
2.4.1.3 Robô de coordenadas esféricas . . . . .	16
2.4.1.4 Robô SCARA . . . . .	17
2.4.1.5 Robô antropomórfico . . . . .	17
2.4.2 Motor . . . . .	18
2.4.2.1 Motor de passo . . . . .	18
2.4.2.2 Motor de corrente contínua . . . . .	18
2.4.2.3 Motor de corrente contínua <i>brushless</i> . . . . .	19
2.4.2.4 Motor de corrente alternada . . . . .	19
2.4.3 Cinemática . . . . .	19
2.4.3.1 Cinemática direta . . . . .	19
2.4.3.2 Cinemática Inversa . . . . .	21

2.5 ROS . . . . .	21
2.6 Gazebo . . . . .	22
2.7 OpenCV . . . . .	22
2.8 MoveIt . . . . .	22
2.9 Ambiente de operação . . . . .	23
<b>3 DESENVOLVIMENTO DO SISTEMA</b>	<b>25</b>
3.1 Descrição do sistema . . . . .	25
3.1.1 Arquitetura geral . . . . .	25
3.1.2 Especificação técnica . . . . .	26
3.1.2.1 Estrutura do manipulador . . . . .	26
3.1.2.2 Atuadores . . . . .	28
3.1.2.3 Câmera . . . . .	29
3.1.2.4 Volume de trabalho . . . . .	29
3.1.2.5 Unidade de potência . . . . .	30
3.1.3 Estrutura analítica do protótipo . . . . .	31
3.2 Especificação funcional . . . . .	32
3.2.1 Funcionalidade de busca . . . . .	32
3.2.1.1 Descrição . . . . .	32
3.2.1.2 Premissas necessárias . . . . .	33
3.2.2 Funcionalidade de aproximação . . . . .	33
3.2.2.1 Descrição . . . . .	34
3.2.2.2 Premissas necessárias . . . . .	34
3.3 Arquitetura de software . . . . .	34
3.4 Simulação do sistema . . . . .	34
3.4.1 <i>CAD</i> . . . . .	35
3.4.2 <i>ROS</i> . . . . .	35
3.4.3 <i>MoveIt</i> . . . . .	35
3.4.4 Pacote <i>bir_marker_localization</i> . . . . .	35
3.5 Testes realizados e resultados . . . . .	36
<b>4 CONFIABILIDADE DO SISTEMA</b>	<b>43</b>
4.1 Análise dos modos e efeitos de falhas . . . . .	43

---

<b>5 GESTÃO DO CONHECIMENTO</b>	<b>47</b>
5.1 Lições aprendidas . . . . .	47
5.2 Guia de uso . . . . .	47
<b>6 CONCLUSÃO</b>	<b>51</b>
<b>REFERÊNCIAS</b>	<b>53</b>
<b>APÊNDICE A Tabelas</b>	<b>57</b>
<b>APÊNDICE B Script em Octave para o cálculo do <i>workspace</i></b>	<b>59</b>

# 1 INTRODUÇÃO

A robótica é um campo relativamente jovem da tecnologia moderna que atravessa os limites da engenharia tradicional ([SPONG; HUTCHINSON; VIDYASAGAR, 2005](#)). O estudo da robótica é um ramo da tecnologia que engloba área de Mecânica, Eletrônica e Computação, com graus de teoria de controle, microeletrônica, inteligência artificial, fatores humanos e de produção ([PIMENTA, 2009](#)). Segundo ([ERTHAL, 1992a](#)), o crescimento da robótica na indústria é justificado em face das exigências de maior qualidade, produtividade e flexibilidade nos processos fabris. Na área industrial, a robótica evoluiu devido ao aumento de uso de robôs e manipuladores industriais.

O estudo do desenvolvimento de manipuladores robóticos foi iniciado por volta de 1954 com George Devol, quando foi desenvolvido o primeiro robô programável e desde então grandes desenvolvimentos nessa área foram atingidos. Como resultado desse avanço, o investimento de empresas mundiais em torno de 16,5 bilhões de dólares em 2018, chegando a marca de 420 mil unidades enviadas globalmente, com perspectiva de crescimento médio de 12% ao ano entre 2020 e 2022. ([INDUSTRIAL..., 2019](#)). O investimento de empresas mundiais em torno de 16,5 bilhões de dólares em 2018, chegando a marca de 420 mil unidades enviadas globalmente, com perspectiva de crescimento médio de 12% ao ano entre 2020 e 2022. ([INDUSTRIAL..., 2019](#)).

Um manipulador robótico consiste em um dispositivo mecânico composto de elementos rígidos (elos) que proporcionam a sustentação e alcance do braço. A inevitabilidade de apresentar algum grau de flexibilidade faz com que esses elos necessitem ser projetados para apresentar elevada rigidez aos esforços que o manipulador será submetido. Esses elos são conectados entre si através de articulações (juntas), que oferecem graus de liberdade ao manipulador e controle de movimento relativo entre os elos. Essas juntas podem ser basicamente divididas em dois grupos: juntas prismáticas e juntas de rotação. Neste projeto foi utilizado as juntas de rotação. A disposição dessas juntas, determina a classificação dos manipuladores, sendo classificados como: robô de coordenadas cartesianas, de coordenadas esféricas, *Selective Compliance Articulated Robot Arm* (SCARA), entre outros ([ROMANO, 2002](#)).

O presente trabalho consiste no desenvolvimento de um manipulador robótico antropomórfico de cinco graus de liberdade, com capacidade de detecção e aquisição de dados. Para isso, será abordado conceitos sobre cinemática direta e inversa, arquisição e processamentos de dados através de uma câmera e conceitos relacionado ao *Robot Operating System* (ROS). Este projeto foi realizado no Laboratório de Robótica e Sistemas Autônomos do SENAI CIMATEC.

## 1.1 Objetivos

O objetivo deste projeto é a construção, simulação e a aplicação de um manipulador robótico antropomórfico automatizado, com a finalidade de acionar uma botoeira de emergência através da leitura de um marcador quadrado, que possui um identificador, chamado *Augmented Reality University of Cordoba* (ARUCO). A modelagem estrutural e matemática, bem como o dimensionamento e criação de um robô compatível com o *framework* do *ROS*, foram etapas precedentes à simulação.

## 1.2 Justificativa

Dados da ([INTERNATION..., 2018](#)) mostram que o ritmo da automação industrial está acelerando em grande parte do mundo desenvolvido. Ele ainda aponta que nas américas a taxa de crescimento atingiu 20% a mais do que no ano anterior. O Brasil, segundo ([EXECUTIVE..., 2019](#)), terá um crescimento de 10% no número unidades de robôs para os anos de 2019 até 2021. O Brasil em 2017 tinha 12.373 robôs e que esse número é muito inferior se compara com a China que possui 473.429, que lidera o *rank* ([BRASIL..., 2019](#)). Segundo ([BRASIL..., 2019](#)), em 2019 o Brasil possui em torno de 16 mil robôs, sendo a maioria no setor automobilístico. Dessa forma, o Brasil precisa de mais capacitação para o evoluir desenvolvimento da robótica e poder assim se aproximar das grandes potências mundiais.

Na indústria, a utilização de um braço robótico vem da necessidade de aumentar a produtividade e melhorar a qualidade dos produtos. No campo acadêmico é uma ferramenta que auxilia nos processos de ensino e pesquisa, além de aplificar os conceitos de robótica sobre problemática da movimentação no espaço tridimensional, modelos matemáticos da cinemática e da dinâmica de movimento e aspectos de controle de movimento.

## 1.3 Organização do relatório

O trabalho está dividido da seguinte forma: O Capítulo 2 apresenta os conceitos do sistema de um manipulador mecânico, os requisitos do cliente e técnicos, toda teoria relacionada aos manipuladores e o ambiente de operação; O Capítulo 3 denota o desenvolvimento do sistema, detalhando a descrição do sistema, as especificações funcionais, a arquitetura de software, simulação do sistema e os testes realizados; O Capítulo 4 aponta a confiabilidade do sistema; O Capítulo 5 apresenta as lições aprendidas; Por fim, o Capítulo 6 apresenta as conclusões finais.

## 2 CONCEITO DO SISTEMA

Neste capítulo serão apresentados os principais conceitos relacionados aos manipuladores mecânicos.

A norma técnica ([ISO-8373, 2012](#)) criada para padronizar o vocabulário referente aos robôs e dispositivos robóticos operando em ambientes industriais e não industriais, define o manipulador como uma máquina na qual o mecanismo, geralmente, consiste em uma série de segmentos, articulados ou deslizantes entre si, com o objetivo de empunhar e/ou mover objetos (peças ou ferramentas) em vários graus de liberdade. Em outras palavras, um manipulador é um equipamento baseado em atuadores programável, com um certo grau de liberdade, projetado para realizar uma variedade de atividades, assim como realização de diversos processos industriais ([ISO-8373, 2012](#)).

### 2.1 Parâmetros básicos

Um manipulador robótico é constituído pela combinação de alguns elementos fundamentais para seu funcionamento, sustentação, movimentação e interação com o meio em que está inserido ([ROMANO, 2002](#)). Para isso, ainda ([ROMANO, 2002](#)), divide-se sua estrutura em cinco componentes principais: manipulador mecânico, atuadores, sensores, unidade de controle, unidade de potência e efetuador. O detalhamento desses conceitos serão descritos a seguir.

#### 2.1.1 Manipulador mecânico

Esse componente refere-se ao aspecto mecânico e estrutural do manipulador, sendo composto por elementos rígidos (corpos ou elos) que são projetados de forma a suportar esforços de flexão e torção. Esses elos, comumente, são compostos de perfis de alumínio ou aço, mas também podem ser compostos de materiais como plásticos reforçados, fibras de carbono ou fibras de vidro em aplicações mais nobres ([ROMANO, 2002](#)). Os elos são conectados através de articulações, que permitem girar e controlar o movimento relativo entre os elos. As articulações também determinam os graus de liberdade que um robô apresenta. Por último, a parte estrutural possui o sistema de transmissão de potência mecânica originada dos atuadores. O sistema de transmissão depende do tipo do projeto e como a potência será transmitida aos elos.

#### 2.1.2 Atuadores

São componentes que enviam potência mecânica aos elos para assim permitir a movimentação. Os atuadores podem ser do tipo hidráulicos, pneumáticos ou eletromagnéticos. Esses atuadores apresentam grande eficiência de torque em diferentes faixas de velocidade

e alguns deles possuem sensores de posição angular e de velocidade para controle de posicionamento, permitindo assim um melhor controle da posição e orientação do efetuador (WEBER et al., 2001).

### 2.1.3 Sensores

Sensores são dispositivos que podem ser adicionados a um manipulador para conceber sentidos e percepção. Um exemplo disso é a capacidade de perceber um espectro além do visível ao olho humano. Sensores são transformadores de energia e convertem alguma grandeza física em um sinal elétrico (RUOCO, 2013). São usados para influenciar o comportamento do manipulador, conferindo a capacidade de avaliar posição e velocidade dos elos ou obter características e informações do ambiente para poder realizar o planejamento de trajetórias e de forma autônoma realizar as atividades determinadas.

### 2.1.4 Unidade de controle

O gerenciamento e monitoração dos parâmetros de operação para realizar as atividades são realizados na unidade de controle. Os comandos de movimentação enviados aos atuadores são originados através de uma integração da unidade de controle e os sensores.

### 2.1.5 Unidade de potência

A energia necessária para a movimentação dos atuadores e consequentemente das juntas é proveniente da unidade de potência.

### 2.1.6 Efetuador

O efetuador é a ferramenta do robô, sendo chamado pela comunidade da robótica de “*end effector*”. Esse elemento, importante para realização da tarefa do manipulador, é um dispositivo que se localiza no final do braço robótico e é responsável por interagir com o ambiente. Aspersores de tinta e geradores de arco de solda são exemplos comuns na indústria.

## 2.2 Requisitos do cliente

Requisitos predeterminados pelo cliente consistem em exigências de funcionamento que devem ser observadas ao final do projeto, para que se considere um sucesso a concepção deste. Para tal, foram determinadas algumas características desejáveis no projeto:

1. O manipulador deve estar acoplado em uma base fixa;
2. Ele deve acionar uma botoeira de emergência posicionada a uma distância de

- aproximadamente um metro da base do manipulador;
3. A posição e orientação da botoeira serão estimadas a partir da leitura de uma *tag*;
  4. Deverão ser utilizados câmera Teledyne Genie Nano C2590 e motores Dynamixel.

## 2.3 Requisitos técnicos

Os requisitos técnicos de um projeto são especificações necessárias para o funcionamento esperado do projeto. Podem ser sobrepostos aos requisitos do cliente em caso de conflito entre o esperado pelo cliente e o necessário para que o projeto seja bem sucedido, objetivando manter o projeto o mais eficiente dentro do escopo planejado. Os requisitos foram:

1. Apresentar pelo menos quatro graus de liberdade;
2. Não ultrapassar carga máxima de 19 N no efetuador;
3. Utilizar *MoveIt* para planejar a trajetória do manipulador;
4. Utilizar [ROS](#);
5. Capacidade de identificar a *tag* e calcular toda trajetória até o acionamento do botão.

## 2.4 Fundamentação Teórica

O primeiro manipulador industrial, conhecido como *UNIMATE*, foi instalado em 1961 numa planta da *General Motors* em Nova Jersey ([IEEE...](#), s.d.). Desde então vários equipamentos puderam ser agregados aos manipuladores, concedendo-lhes utilizações diversas em robôs industriais no processo produtivo, mas também expandindo-os a vários âmbitos, o que possibilitou avanços constantes nas áreas de mecânica, eletrônica digital, ciência da computação entre outras.

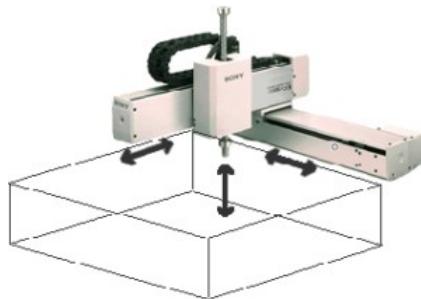
### 2.4.1 Robô Manipulador

Manipuladores podem ser teleoperados ou autônomos. São comumente utilizados em tarefas que apresentam riscos aos operadores, por repetitividade demaciada, ou onde se exige uma combinação de rapidez e precisão além capacidade humana. Soldagem, pintura, montagem, manipulação de materiais perigosos e inspeção de ambientes inóspitos são mais eficientemente realizadas por robôs. Segundo ([CARRARA, 2015](#)), os manipuladores podem ser subdivididos quanto a estrutura mecânica em cinco grupos: Robô de coordenadas cartesianas; Robô de coordenadas cilíndricas; Robô de coordenadas esféricas; Robô SCARA e Robô antropomórfico.

#### 2.4.1.1 Robô de coordenadas cartesianas

Este robô possui três juntas prismáticas, resultando em movimentos de translação. Tal nomenclatura deve-se ao movimento coincidente com um sistema cartesiano de coordenadas espaciais. Ele possui um *workspace*, ou seja, um volume de trabalho limitado ao formato de um retângulo, como mostrado na figura 1.

Figura 1: Volume de trabalho de robô cartesiano.



Fonte: ([COLLINS, 2018](#)).

#### 2.4.1.2 Robô de coordenadas cilíndricas

A configuração do movimento deste robô é descrita num sistema de coordenadas cilíndricas. É constituído de uma junta de rotação e duas juntas prismáticas, gerando um volume de trabalho limitado por superfícies de simetria cilíndrica como na figura 2.

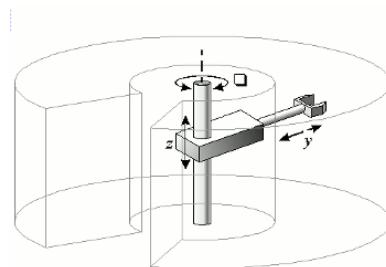


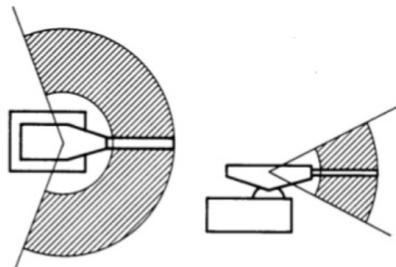
Figura 2: Volume de trabalho de robô de coordenadas cilíndricas.

Fonte: ([ALL... s.d.](#))

#### 2.4.1.3 Robô de coordenadas esféricas

O movimento dos eixos desse robô formam um sistema de coordenadas de referência polar, com uso de duas juntas de rotação e apenas uma junta prismática responsável pelo movimento de translação. Neste robô o volume de trabalho está limitado por superfícies de simetria esférica, como mostra a figura 3.

Figura 3: Áreas de trabalho de robô de coordenadas esféricas.

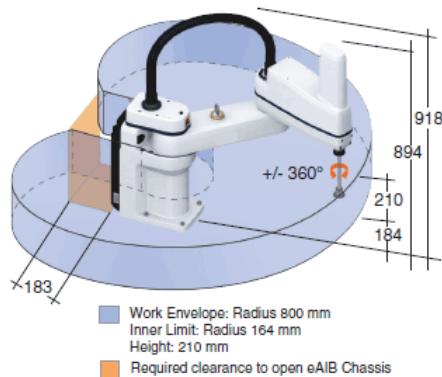


Fonte: ([C3, 2015](#))

#### 2.4.1.4 Robô SCARA

O robô [SCARA](#), assim como o robô de coordenadas esféricas, apresenta duas juntas rotacionais e uma junta prismática. Esse robô é muito utilizado em pequenas atividades e montagem de pequenos componentes. Seu volume de trabalho, como mostra a figura 4, é gerado de forma aproximadamente cilíndrica.

Figura 4: Volume de trabalho de robô SCARA.

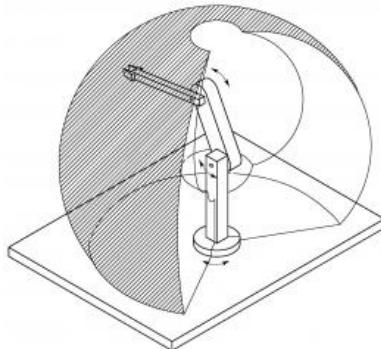


Fonte: ([AUTOMATION, 2016](#))

#### 2.4.1.5 Robô antropomórfico

O objeto de estudo deste projeto se baseia em um robô do tipo antropomórfico, que tem como configuração ao menos três juntas rotacionais. Esse manipulador possibilita diversas posições e orientações do seu efetuador devido a sua maior mobilidade, formando um volume de trabalho mais complexo em comparação as demais configurações. O volume de trabalho desse robô pode ser visto na figura 5.

Figura 5: Volume de trabalho de robô antropomórfico.



Fonte: ([FEDERICA...](#), s.d.)

## 2.4.2 Motor

Um motor é um dispositivo que converte outras formas de energia em energia mecânica, de forma a impelir movimento a uma máquina ou veículo ([WIKIPÉDIA, 2020](#)). Os tipos de motores empregados em manipuladores seguem abaixo, em ordem de utilização, da menor para a maior.

### 2.4.2.1 Motor de passo

Motores de passo são motores *Direct Current* (DC) cujo rotor gira em incrementos angulares discretos quando os enrolamentos do estator são energizados de maneira programada ([ATHANI, 2005](#)). São de fácil controle de velocidade e posição, e num primeiro momento são bastante previsíveis, pois se assume que a quantidade de passos dada é a quantidade especificada, a menos que ocorra um deslizamento das juntas por influência de forças externas. Isso demanda a presença de sensoriamento para o correto posicionamento dos elos. Além disso, não é possível controlar o torque entre passos, e comparado aos motores que seguirão, sua potência é inferior. Utilizado em poucas aplicações ([WEBER et al., 2001](#)).

### 2.4.2.2 Motor de corrente contínua

Conhecidos por motores DC são controlados por tensão, tipicamente por meio de modulação por largura de pulso, ou *Pulse Width Modulation* (PWM). O seu controle de velocidade é dependente de sensoriamento, tipicamente realizado por um encoder colocado no eixo do motor. São motores baratos mas não oferecem altas potências. Utilizam escovas para a conexão elétrica entre o rotor e a fonte de energia ([WEBER et al., 2001](#)).

#### 2.4.2.3 Motor de corrente contínua *brushless*

Não possuem as escovas dos motores DC. Tipicamente utilizam sensores de efeito Hall na medição da posição do rotor. Motores brushless tem desempenho melhor que motores DC com escovas, mas são mais caros (WEBER et al., 2001).

#### 2.4.2.4 Motor de corrente alternada

Motores de corrente alternada são os mais robustos levando em conta o custo benefício porém são mais difíceis de controlar por conta das suas características eletromagnéticas não-lineares, necessitando o uso de inversores de frequência. Seu comportamento não-linear oportuniza o uso de técnicas de controle não-linear. São usados amplamente em aplicações industriais (WEBER et al., 2001).

### 2.4.3 Cinemática

A cinemática de um manipulador é o estudo do conjunto de relações entre as posições, velocidades e acelerações de seus elos (SANTOS, 2004). A cinemática é usualmente dividida em duas partes: cinemática direta e cinemática inversa. Segundo (NAKAMURA, 1986), a cinemática possui dois problemas a estudar: a cinemática de posição e a diferencial. A cinemática de posição trata da relação entre as coordenadas de juntas e posição do efetuador, enquanto que a cinemática diferencial trata da relação entre os deslocamentos infinitesimais das juntas e os do efetuador.

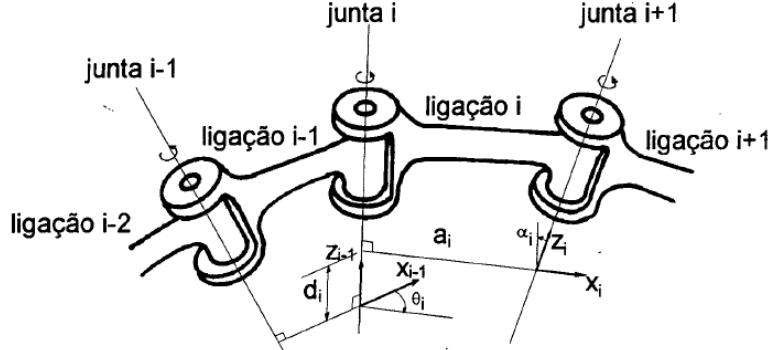
#### 2.4.3.1 Cinemática direta

Na cinemática direta visa-se obter a posição e orientação no espaço cartesiano ( $x, y, z$ ) em relação ao sistema fixo de coordenadas, a partir dos ângulos das juntas do manipulador (SANTOS, 2016). Ela determina qual a posição e orientação do efetuador, sendo conhecidas as posições das juntas do robô.

As técnicas mais comumente utilizadas para a obtenção da equação cinemática baseiam-se na geometria do robô ou em equações algébricas (ERTHAL, 1992b). Para este trabalho foi utilizado a notação *Denavit-Hartenberg* (D-H) com as matrizes homogêneas. Por esse motivo, será descrito somente essa técnica para obter a solução da cinemática direta.

Segundo (ERTHAL, 1992b), a formulação utilizando matrizes homogêneas juntamente com a notação de D-H é a mais utilizada por permitir estudos tanto em cinemática quanto em dinâmica. Os sistemas de coordenadas das ligações são fixados ao longo das juntas de modo a facilitar a definição das rotações e translações (ERTHAL, 1992b). As origens dos sistemas das ligações são localizadas na normal comum entre as juntas que a limita, como mostra a figura 6

Figura 6: Modelagem utilizando matrizes homogêneas e a notação de D-H.



Fonte: ([ERTHAL, 1992b](#))

Dessa forma, segundo ([SLOTINE HARUHIKO ASADA, 1986](#)), a transformação do sistema  $i - 1$  para o sistema  $i$  é composta por quatro transformações elementares agrupadas numa mesma matriz, como mostra a equação 2.1.

$$A_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

Como pode ser visto na equação 2.1,  $i$  representa o índice da junta,  $\theta_i$  o ângulo da junta  $i$ ,  $d_i$  o deslocamento da junta  $i$ ,  $a_i$  o comprimento da ligação  $i$  e  $\alpha_i$  a torção da ligação  $i$ . Os parâmetros  $\theta_i$ ,  $d_i$ ,  $a_i$  e  $\alpha_i$  são denominados parâmetros de D-H. Três desses parâmetros são constantes e apenas um é variável ( $\theta_i$  ou  $d_i$ ), conforme o tipo de junta seja prismática ou rotativa ([ERTHAL, 1992b](#)).

O algoritmo de D-H permite obter os sistemas de coordenadas e as transformações associadas a cada elo de manipulador ([SANTOS, 2004](#)). É utilizado principalmente quando há uma complicação na análise do manipulador por ter muitos graus de liberdade. O cálculo da cinemática direta é feito pelas transformações sucessivas entre os sistemas de coordenadas desde a base do robô até o efetuador, quando considerado que os parâmetros já são conhecidos. Isso é obtido pelas multiplicações sucessivas das matrizes  $A_i^{i-1}$  da equação 2.1, fazendo  $i$  variar de 1 até N, em que N é número total de juntas. Deste modo, a equação da cinemática direta é representada pela equação 2.2.

$$T_N = A_1^0(\theta_1) \times A_2^1(\theta_2), \dots, A_N^{i-1}(\theta_N) \quad (2.2)$$

### 2.4.3.2 Cinemática Inversa

A obtenção da cinemática inversa é mais complexa que a direta por se tratar da resolução de um sistema de equações não lineares. A cinemática inversa nem sempre é um problema de solução analítica, ou muitas vezes não possui uma solução (HERNANDES, 2014). Contudo todas as configurações conhecidas possuem soluções, somente para novas configurações é que não existe um método específico e talvez seja necessária a invenção de um método novo (SANTOS, 2004).

A cinemática inversa é o método pelo qual se procura determinar o conjunto de valores das juntas que se relaciona com uma dada configuração do espaço cartesiano ou operacional (SANTOS, 2004). Em outras palavras, segundo (ERTHAL, 1992b), a cinemática inversa trata da obtenção das coordenadas das juntas, dadas as coordenadas externas do efetuador.

Um manipulador é dito solucionável quando todos os conjuntos de variáveis de juntas, associados a um dado posicionamento do efetuador, puderem ser obtidos numa forma explícita (RAO, 1989). Existem diversos métodos descritos na literatura para solucionar a cinemática inversa, mas somente será detalhado o método inverso de Jacobiano.

Na cinemática direta tem-se uma relação do tipo  $\vec{r} = \vec{F}(\vec{q})$ , ou seja, a posição do manipulador é a função da posição das juntas, no qual  $\vec{r}$  representa a velocidade linear e angular do efetuador e  $\vec{q}$  representa a velocidade angular das juntas (SANTOS, 2004; BARINKA, s.d.). Já na cinemática inversa tem-se a relação inversa do tipo  $\vec{q} = \vec{F}_{-1}(\vec{r})$ , isto é, a posição das juntas está em função da configuração espacial do manipulador (SANTOS, 2004). Desse modo, para relacionar as velocidades de cada junta do manipulador, é necessário ter o conceito da matriz jacobiana, que se relaciona com as posições das juntas e do espaço operacional, dado pela equação 2.3.

$$d\vec{r} = J \times d\vec{q} \quad (2.3)$$

Pela equação 2.3, segundo (HERNANDES, 2014), as velocidades espaciais estão relacionadas com a velocidade das juntas através da matriz jacobiana, que é definida para  $n$  juntas e  $m$  variáveis cartesianas, como mostra a equação 2.4.

$$J = \begin{bmatrix} \frac{\partial r_1}{\partial q_1} & \dots & \frac{\partial r_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial r_m}{\partial q_1} & \dots & \frac{\partial r_m}{\partial q_n} \end{bmatrix} \quad (2.4)$$

## 2.5 ROS

O ROS é um esforço colabrativo para criar um mecanismo robusto de uso geral na criação de aplicativos para robótica (ANDERSON, s.d.). Segundo (WIKIPÉDIA, 2019), o

*ROS* é uma coleção de *frameworks* de *software* para desenvolvimento de robôs, que fornece a funcionalidade de um sistema operacional em um *cluster* de computadores heterogêneo. Ele fornece serviços padrões de um sistema operacional, tais como abstração de *hardware*, controle de dispositivos de baixo nível, a implementação de funcionalidades comumente usadas, passagem de mensagens entre processos e gerenciamento de pacotes.

## 2.6 Gazebo

O Gazebo começou o desenvolvimento em 2002 na Universidade do Sul da Califórnia, criando um conceito de simulador de alta fidelidade, no qual sugiu da necessidade de simular robôs em ambientes externos sob várias condições ([GAZEBO, 2014](#)). Segundo ([GAZEBO, 2014](#)), a simulação de robô é uma ferramenta essencial para todo desenvolvedor de robótica. Um simulador bem projetado torna possível testar rapidamente algoritmos, projetar robôs, executar testes de regressão e treinar o sistema de inteligencia artificial usando cenários realistas. Ainda ([GAZEBO, 2014](#)) que diz oferece uma capacidade de simular com precisão e eficiência populações de robôs em ambientes internos e externos complexos, onde existe um mecanismo de física robusto, gráficos de alta qualidade e interfaces gráficas e programáticas convenientes.

## 2.7 OpenCV

O *Open Source Computer Vision Library* (OpenCV) é uma biblioteca de *software* livre de visão computacional e aprendizado de máquina ([OPENCV, 2020](#)). Ela foi desenvolvida para fornecer uma infraestrutura comum para aplicativos de visão computacional e acelerar o uso da percepção da máquina nos produtos comerciais, que possui mais 2500 algoritmos otimizados ([OPENCV, 2020](#)). Segundo ([OPENCV, 2020](#)), esses algoritmos podem ser usados para detectar e reconhecer rostos, identificar objetos, classificar ações humanas em vídeos, rastrear movimentos de câmeras, rastrear objetos em movimento, extrair modelos 3D de objetos, entre outras aplicações. A biblioteca possui interfaces *C++*, *Python*, *Java* e *MATLAB* com suporte para os sistemas operacionais *Windows*, *Linux* e *MAC OS*.

## 2.8 MoveIt

Moveit é um pacote que é executado sobre o ROS, fornecendo funcionalidades para cinemática, planejamento de movimento e caminho, verificação de colisão, percepção 3D, interação do robô, entre outras aplicações ([MOVEIT, s.d.b](#)). O MoveIt é a principal fonte de muitas funcionalidades para manipuladores no ROS. Ele se baseia nos sistemas de mensagens e construção do ROS e utiliza algumas das ferramentas comuns do ROS, como o *ROS Visualizer* (*RViz*) e o formato do robô *Universal Robotic Description Format* (*URDF*). Segundo ([MOVEIT, s.d.b](#)), o MoveI está se tornando o ponto de entrada no ROS, especialmente através do uso do MoveIt Setup Assistant para configurar novos robôs. Como

nosso foco é sobre os manipuladores, para melhor compreensão sobre as funcionalidades e os conceitos do MoveIt pode ser consultado na referência ([MOVEIT, s.d.a](#)).

## 2.9 Ambiente de operação

O manipulador desenvolvido será acoplado a uma bancada que possui uma caixa com uma botoeira de emergência e uma *tag* do tipo ARUCO a ela acoplada. O centro da base quadrada do manipulador será fixado em uma extremidade da bancada e a caixa será colocada em um local qualquer dentro do volume de trabalho do manipulador. Não haverá outro tipo de obstáculo, como ilustrado na figura 7.

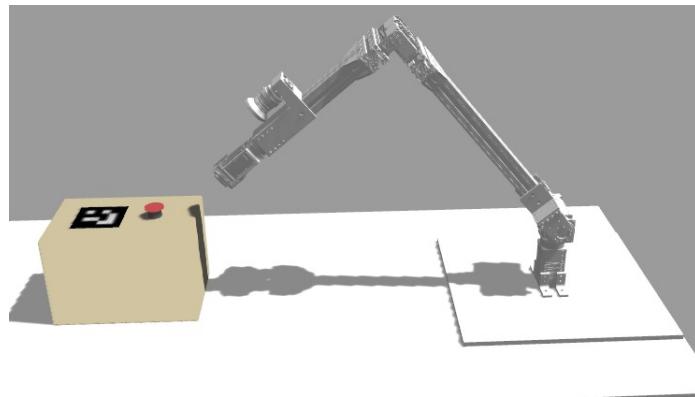


Figura 7: Ambiente de operação simulado.

Fonte: Os autores.



### 3 DESENVOLVIMENTO DO SISTEMA

Uma boa conduta no desenvolvimento de um sistema robótico é levar os seus componentes para representações em simulação. Para isso necessitou-se, em primeiro lugar, utilizar uma ferramenta de *Computer Aided Design (CAD)*. A ferramenta online gratuita *Onshape* foi escolhida e através realizou-se a montagem dos objetos que virão a compor a cena esperada do sistema. A partir disto pôde-se inserir os objetos, por meio de uma descrição em arquivos do tipo *URDF*. Estes arquivos são códigos em linguagem XML onde se fazem declarações sobre como os elementos que compõem um robô, do ponto de vista geométrico principalmente, se conectam. O simulador *Gazebo* utiliza estes arquivos como parte dos dados necessários para calcular a dinâmica do sistema para posteriormente exibir estes resultados através de uma interface gráfica.

#### 3.1 Descrição do sistema

O conjunto formado pelo manipulador, seu espaço de trabalho e os objetos com os quais ele deverá interagir compõem o sistema abordado neste trabalho. O espaço de trabalho foi representado como uma bancada de  $1,70\text{ m} \times 0,80\text{ m} \times 0,028\text{ m}$ . Há apenas um objeto com o qual o manipulador deverá interagir, uma caixa de dimensões  $0,30\text{ m} \times 0,20\text{ m} \times 0,20\text{ m}$ .

Nesta caixa há dois objetos: um marcador fiducial do tipo *ARUCO* que é a codificação gráfica de um número, tal como os *Quick Response (QR)* code já bastante populares. Essa etiqueta é responsável por sinalizar ao manipulador que uma determinada localização foi encontrada, e, a partir disto, uma tomada de decisão deve ser feita por ele. A decisão, para este trabalho, é a de pressionar a botoeira de emergência contida na caixa e referenciada pela tag.

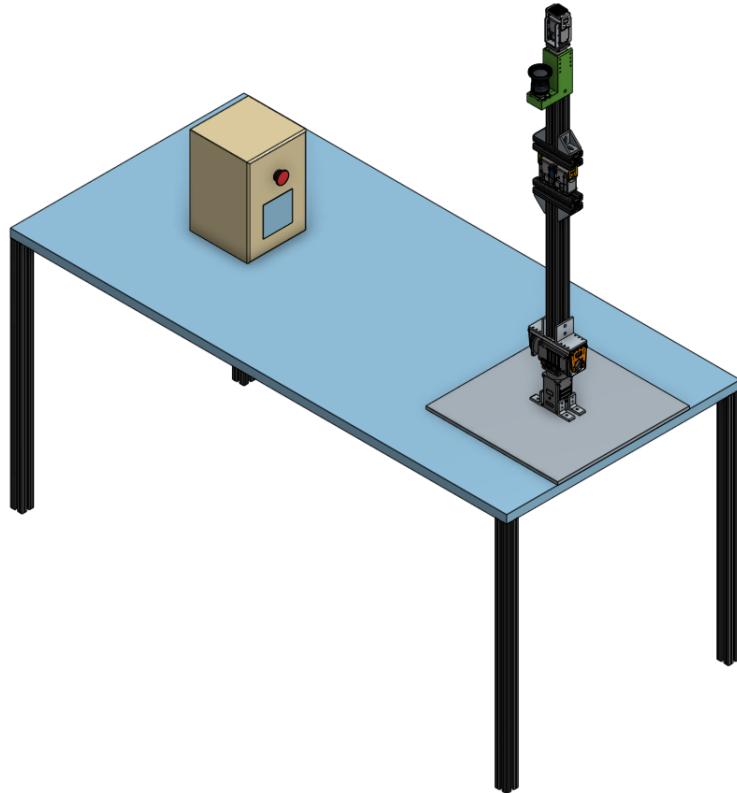
Por fim, há o próprio manipulador, que é articulado em cinco juntas onde se encontram servomotores que realizam a ação de girar, de modo controlado, os elos a eles presos, conferindo cinco graus de liberdade à ferramenta do robô: uma chapa de metal destinada a fazer contato físico com a botoeira e acioná-la.

##### 3.1.1 Arquitetura geral

A arquitetura geral do sistema consiste na integração entre os processos do *ROS* sendo operado pelo sistema principal, como mostra a figura 9. Ocorrem então o movimento do manipulador, sendo controlado pelo *MoveIt*, e a simulação, que é a integração dos componentes virtuais dos sistemas. Nesses componentes estão presentes o manipulador, que possui atuadores e sensores de estados, e uma câmera que simula a parte visual.

A interface humano-máquina pode ser controlada utilizando os softwares *RViz* e *Gazebo*,

Figura 8: Representação visual do sistema.



Fonte: Os autores

permitindo assim ter-se uma visualização geral do cenário que está sendo simulado. Para a detecção visual, é utilizado um pacote externo para interligar os componentes interessados no cenário chamado “*bir\_marker\_localization*”. O *MoveIt Setup Assistant* é responsável por realizar a configuração do manipulador, para que o movimento seja realizado. *MoveIt commander* é interligado com o [ROS](#) e ele é utilizado com a finalidade de controle do movimento do manipulador.

Representa-se na figura 9, de maneira resumida, os componentes do sistema:

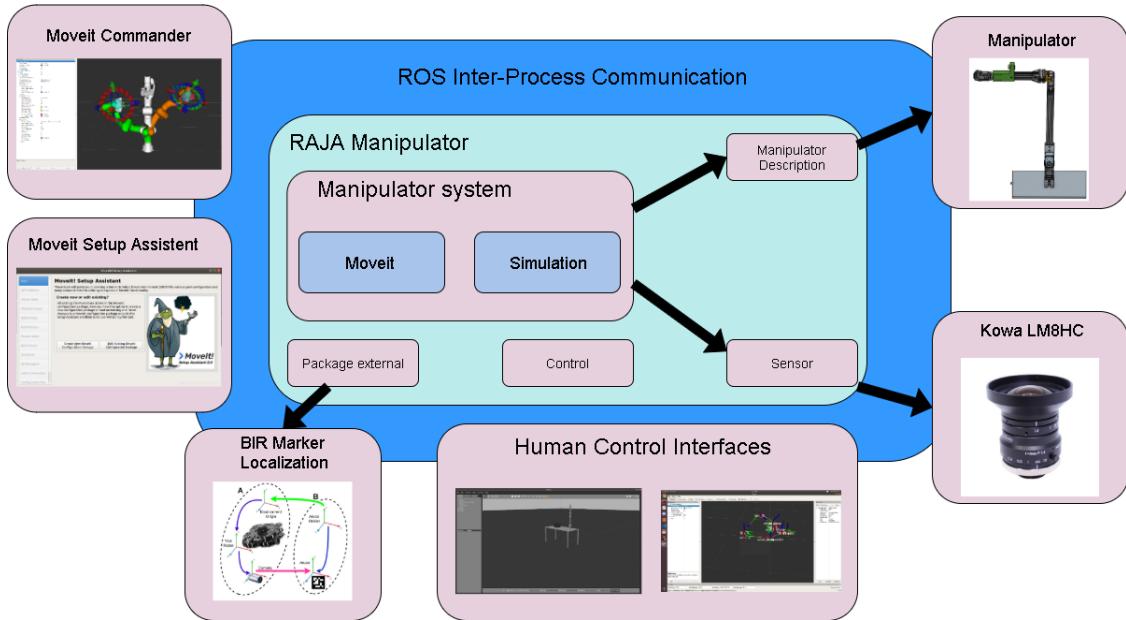
### 3.1.2 Especificação técnica

A presente especificação técnica constitui elemento fundamental para o cumprimento dos requisitos do cliente e o cumprimento do objetivo final do manipulador. Esse tópico será subdividido em alguns pontos para melhor apresentar tais especificações.

#### 3.1.2.1 Estrutura do manipulador

Para estruturação dos elos do robô, são utilizados perfis de alumínio estrutural  $40 \times 40 - 10$  mm, com rosca interna M12, na figura 10. Segundo fabricante deste tipo de perfil ([OBR...](#), [s.d.](#)), os mesmos apresentam tipicamente uma massa de 0,8 kg/m. Foram utilizados dois

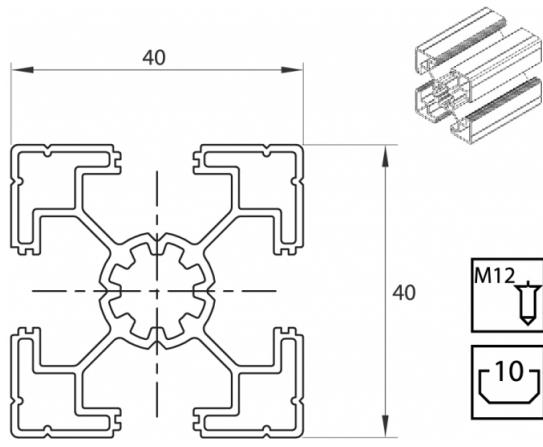
Figura 9: Visão geral dos elementos do sistema.



Fonte: Os autores

elos deste material, o primeiro possuindo 40 cm de comprimento e o segundo, 20 cm. Estas dimensões foram projetadas levando em consideração a capacidade de alcance do braço para realizar a tarefa, bem como a capacidade estrutural do manipulador, para assim suportar esforços de natureza estática e dinâmica. Para a base, foi projetado uma chapa de liga de alumínio 5000 com dimensões  $50 \times 50 \text{ cm}^2$  e espessura de 10 mm. Essa base será utilizada para fixar o manipulador, gerando estabilidade durante a execução da tarefa.

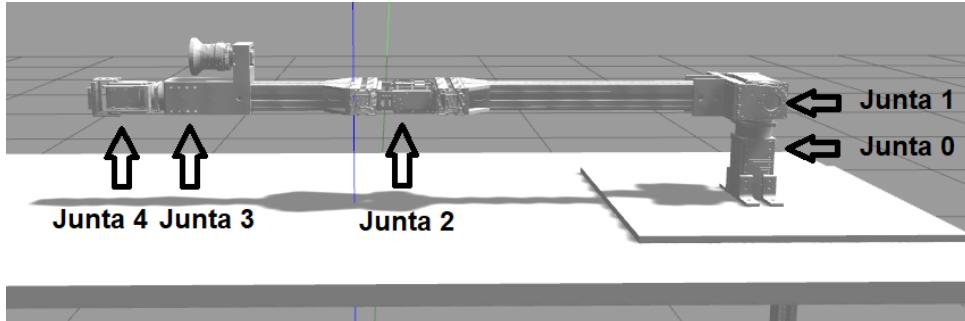
Figura 10: Estrutura dos elos.



Fonte: (OBR..., s.d.)

O manipulador foi dimensionado com o comprimento de 1 m. Levou-se em consideração o mínimo alcance necessário para que a missão possa ser realizada conforme as especificações do cliente. Então foram projetadas cinco juntas rotacionais como mostra a figura 11.

Figura 11: Juntas do manipulador.



Fonte: Os autores.

Tabela 1: Análise de esforços das juntas.

Junta	Momento Torsor	Torque de atuação dos motores	Força máxima suportada pela junta( <a href="#">ROBOTIS, s.d.</a> )
0	17 N/m	44.7 N/m	70 N
1	17 N/m	44.7 N/m	70 N
2	4.4 N/m	2 x 3 N/m	19 N
3	0.4 N/m	5.1 N/m	50 N

Fonte: Os autores.

A análise de esforços mecânicos aos quais o manipulador está exposto foi realizada para embasamento e confirmação da capacidade da estrutura e dos atuadores suportarem os esforços de momento torsor. Para isso foram analisadas individualmente as juntas em seus estados críticos. Na tabela 1 é apresentada a análise de esforços nas juntas e comparados com os máximos esforços suportados pelos atuadores, levando em consideração especificações do fabricante, para evitar falhas mecânicas.

Levando em consideração os valores obtidos nas análises de esforços, tem-se a junta 2 como junta crítica do sistema, ou seja, esta junta chegará a falha mecânica antes das demais juntas devido aos esforços que é submetida e a capacidade dos seus dois motores. Na configuração apresentada a junta 2 suporta a carga máxima de 19 N sem chegar a falha mecânica, logo o manipulador possui um limite (*payload*) de aproximadamente 1,95 kg na sua extremidade.

### 3.1.2.2 Atuadores

Os atuadores do manipulador são motores de corrente contínua, integrados com redutor de velocidade, controlador e driver. Foram utilizados atuadores *Dynamixel* da fabricante ROBOTIS. Entre os modelos figuram o MX-106R (figura 12a) para as articulações 2 e 4, porém na junta 2 eles foram acoplados em paralelo para trabalhar em sincronismo, aumentando assim a capacidade de torque na articulação. Para as juntas 0 e 1, foi

utilizado o motor PH54-200-S500-R (figura 12b). O motor PH42-020-S300-R (figura 12c) foi utilizado para a articulação 3. As especificações do fabricante mais relevantes no projeto estão apresentadas na tabela 2 para os motores utilizados. As folhas de dados estão disponíveis no *website* da ROBOTIS para os atuadores MX-106-R ([ROBOTIS...  
s.d.a](#)), PH42-020-S300-R ([ROBOTIS...  
s.d.b](#)) e PH54-200-S300-R ([ROBOTIS...  
s.d.c](#)).

Tabela 2: Especificações do fabricante para os atuadores.

Especificações dos motores em operação contínua					
Tipo	Torque (N/m)	Voltagem (V)	Corrente (A)	Velocidade de rotação (rpm)	Dimensões (mm)
PH54-200-S500-R	44.7	24.0	9.3	29.0	54.0 X 126.0 X 54.0
PH42-020-S300-R	5.1	24.0	1.5	29.2	42.0 X 84.0 X 42.0
MX-106R	2.9	12.0	5.2	(Não especificado)	40.2 X 65.1 X 46.0

Fonte: ([ROBOTIS, s.d.](#))

Figura 12: Atuadores Dynamixel.



(a) MX-106R.



(b) PH54-200-S500



(c) PH42-020-S300-R

Fonte: ([ROBOTIS, s.d.](#))

### 3.1.2.3 Câmera

Para prover o sistema com capacidade de detecção da *tag* e assim obter dados necessários para aquisição de pose e orientação relativa, utilizou-se uma câmera de video modelo Teledyne Genie Nano C2590 (figura 13). De pequenas dimensões e massa reduzida, essa câmera pode ser acoplada próxima a extremidade do manipulador robótico, facilitando a detecção da *tag* e não comprometendo a estrutura do manipulador.

### 3.1.2.4 Volume de trabalho

A área de trabalho ou *workspace* de um manipulador é o volume total alcançado pelo efetuador do manipulador, executando todos os possíveis movimentos, ou seja, é a região dentro da qual o manipulador pode posicionar o efetuador. Essa área é limitado pela geometria do manipulador, bem como as restrições mecânicas sobre as articulações. Para este projeto foi desenvolvido um script em *Octave* que calcula o *workspace* do manipulador, a partir de um modelo de um braço robótico com 3 juntas possuindo o mesmo comprimento

Figura 13: Teledyne Genie Nano C2590.

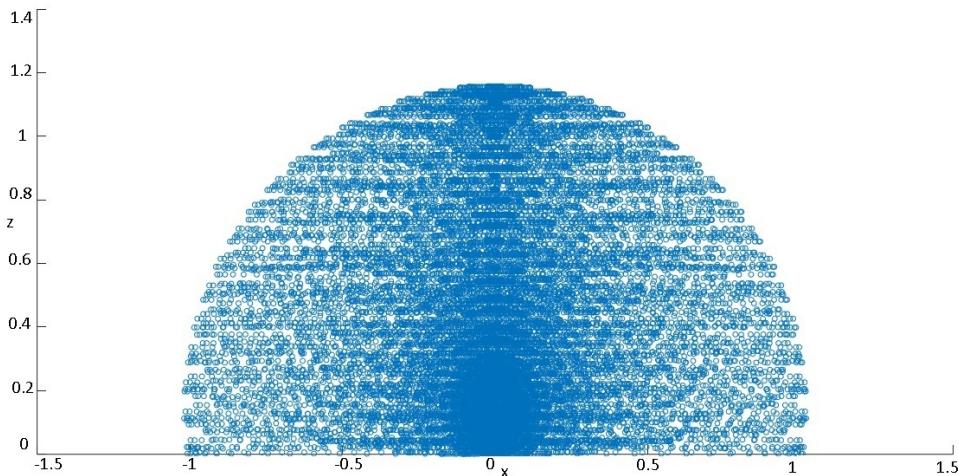


Fonte: ([TELEDYNE...](#), s.d.)

e obedecendo as distâncias entre cada junta até o efetuador. O código desse script pode ser visto no apêndice B.

A partir do script do apêndice B pode-se gerar os gráficos que representa o volume de trabalho alcançado pelo manipulador, como pode ser visto nas figuras 14, 15, 16. A figura 14 mostra o alcance máximo vertical de 1,2 m e horizontal de 1 m. A figura 15 mostra o alcance horizontal pela perspectiva lateral, porém não fica muito evidente o alcance mínimo pois há interposição dos valores. Na figura 16 apresenta uma visão de cima da amplitude do manipulador.

Figura 14: Área de trabalho projetada no plano XZ.

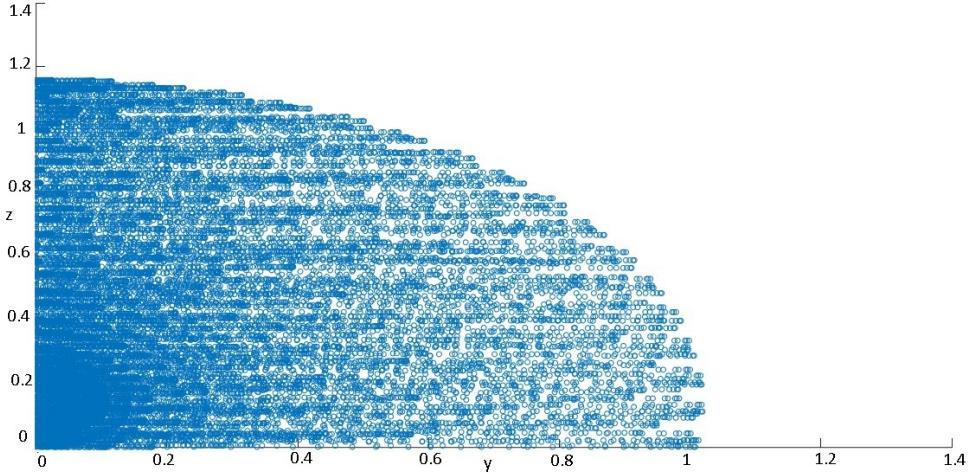


Fonte: Os autores.

### 3.1.2.5 Unidade de potência

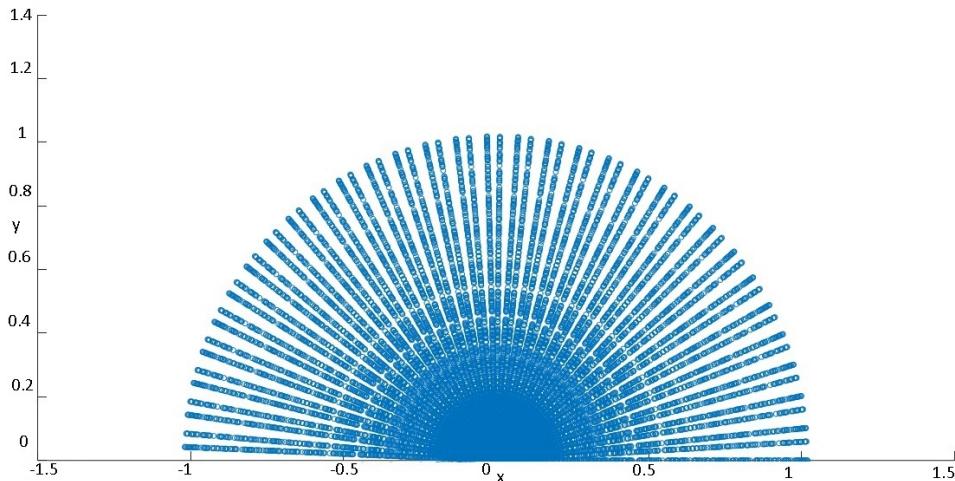
O sistema necessitará ser energizado em dois níveis de tensão, 12 V e 24 V, os mesmos níveis dos atuadores utilizados. Para isso, uma fonte de tensão será necessária juntamente com um conversor DC-DC para ou elevar ou reduzir a tensão de alimentação, caso ela seja 12 V ou 24 V, respectivamente. Pela forma como os motores serão empregados, a potência de entrega da fonte pode ficar muito abaixo da potência máxima do conjunto

Figura 15: Área de trabalho projetada no plano YZ.



Fonte: Os autores.

Figura 16: Área de trabalho projetada no plano XY.



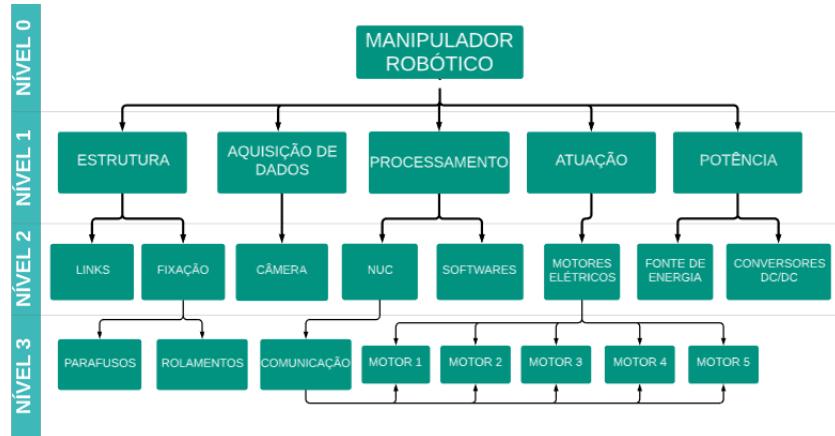
Fonte: Os autores.

de atuadores. Assim uma fonte que forneça 400 W é necessária. Um esquema elétrico é fornecido no apêndice A, indicando as conexões dos cabos entre os motores.

### 3.1.3 Estrutura analítica do protótipo

A estrutura analítica do protótipo consiste em uma subdivisão hierárquica dos seus subsistemas e componentes. Essa estrutura tem como objetivo primário a organização dos níveis que compõem o projeto, proporcionando uma visão ampla do projeto e um maior controle. Foi concebida uma análise estrutural do manipulador robótico, levando em consideração quatro níveis de hierarquia como mostrado na figura 17.

Figura 17: Estrutura analítica do protótipo.



Fonte: Os autores.

## 3.2 Especificação funcional

O manipulador simulado deve ser capaz de realizar uma inspeção visual e busca da *Augmented Reality (AR) tag*. No momento em que esta for encontrada, a busca deve ser interrompida e o manipulador inicia uma rotina de aproximação do botão. Esta rotina é realizada em duas etapas, que separam pela precisão com a qual o manipulador deve se deslocar no espaço. Até que a aproximação atinja um dado limiar, a sua movimentação pode ser realizada de maneira mais abrupta; atingido o limiar, uma aproximação mais suave conduz o efetuador até o botão e o seu pressionamento é realizado, concluindo o objetivo do sistema.

Na figura 18 é ilustrado o fluxograma da missão.

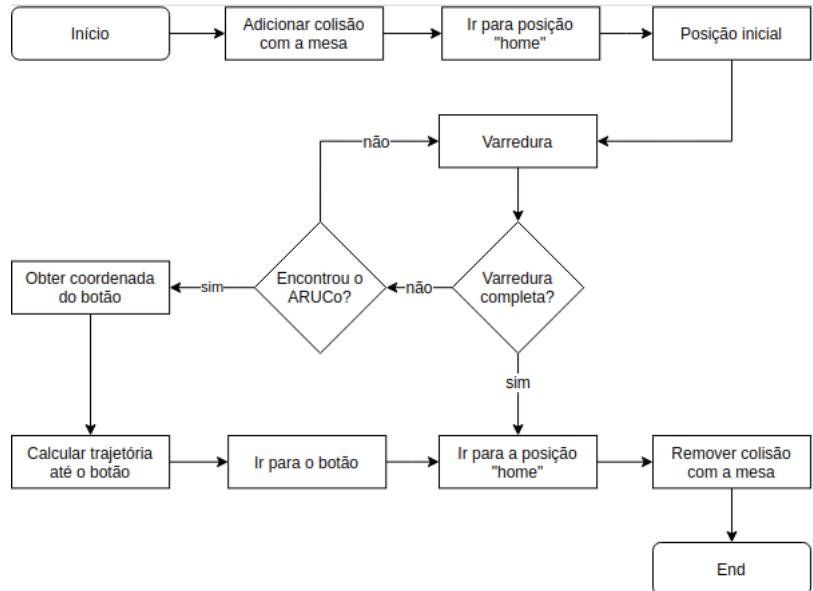
### 3.2.1 Funcionalidade de busca

Para poder localizar-se no espaço de trabalho, o manipulador percorre uma trajetória pré-definida a partir da pose inicial padrão, denominada *home pose*, a fim de detectar, por meio da câmera instalada, a *tag* pré-especificada em suas configurações.

#### 3.2.1.1 Descrição

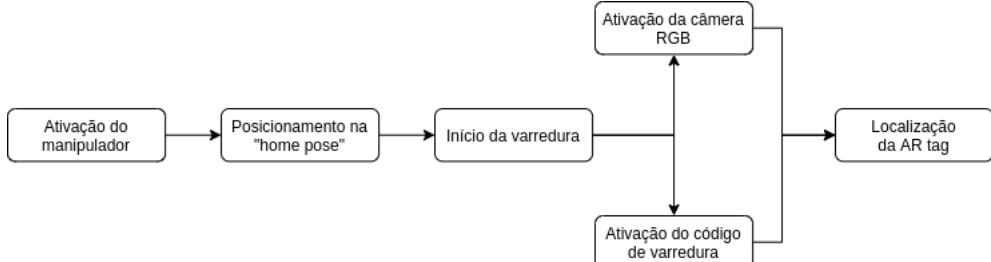
O manipulador inicia a busca a partir da sua posição *home* e realiza dois ciclos. A junta 0 realiza meia volta em cada ciclo de busca. Também em cada ciclo a junta 1, que inicia a 45° com o plano da horizontal, passa por incrementos de 22,5°. Caso o marcador fiducial não seja encontrado, o manipulador retorna à posição *home*. A figura 19 ilustra a sequência do processo de busca.

Figura 18: Fluxograma do sistema.



Fonte: Os autores.

Figura 19: Detalhamento da funcionalidade de busca.



Fonte: Os autores.

### 3.2.1.2 Premissas necessárias

Assume-se que há uma *tag* de tamanho grande o suficiente para ser detectada pela câmera empregada; também assume-se que o movimento executado na busca está adequado ao campo de visão da câmera.

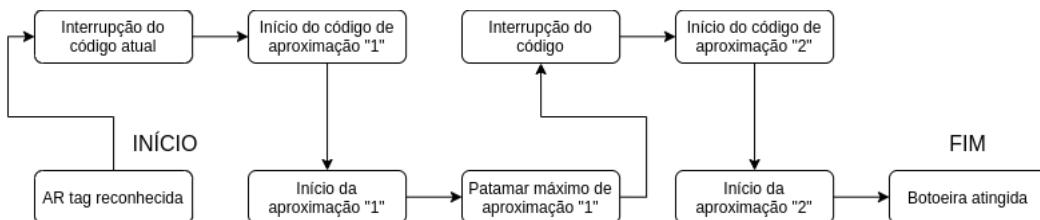
### 3.2.2 Funcionalidade de aproximação

Após a detecção da *tag* e estimativa da pose relativa do efetuador, inicia-se um procedimento de aproximação baseado na posição que o manipulador reconhece que ele ocupa no mundo. Conforme mencionado, a aproximação acontece em dois perfis diferenciados pela velocidade até que se pressione a botoeira de emergência.

### 3.2.2.1 Descrição

A figura 20 ilustra as etapas da aproximação do manipulador ao botão.

Figura 20: Detalhamento da funcionalidade de aproximação.



Fonte: Os autores.

### 3.2.2.2 Premissas necessárias

A funcionalidade aproximação depende do sucesso da aquisição de coordenadas na funcionalidade de busca. É iniciada autonomamente após o sucesso da funcionalidade anterior.

## 3.3 Arquitetura de software

O sistema é dividido em *software* e *hardware*. Os principais componentes que integram o *hardware* são o manipulador e a câmera Teledyne. Já os componentes de *software* são relacionados ao framework [ROS](#), juntamente com o pacote de movimento do sistema, incluindo a biblioteca do *MoveIt Commander*.

## 3.4 Simulação do sistema

Para realizar a simulação no *Gazebo* utilizaram-se ferramentas do framework [ROS](#), além dos modelos em [CAD](#) e os códigos concebidos. O framework ([ROS... , s.d.](#)) consiste em uma série de programas pré-concebidos e ajustáveis usados para representar em simulação dispositivos usuais encontrados no mundo da robótica, e são denominados *plugins*. Os *plugins*, após devidamente personalizados são organizados em pacotes para serem utilizados, possivelmente em conjunto com programas à parte. Neste trabalho, utilizaram-se *plugins* para simular o deslocamento angular dos elos em cada junta e a câmera. Não sendo estes *plugins* suficientes para contemplar todas as tarefas que a ele cabiam, conceberam-se os programas responsáveis por realizar a busca em busca do marcador fiducial e a lógica de movimentação do efetuador para o acionamento da batoeira.

### 3.4.1 CAD

Os modelos em CAD foram adquiridos nos sites dos fabricantes dos produtos (os atuadores *Dynamixel* e os perfis de alumínio) ou foram desenhados (as cantoneiras que conferiram maior rigidez às juntas) e montados na plataforma *Onshape*. Lá, além das dimensões e formas, a massa e a densidade dos materiais utilizados foram fornecidos e com isso foram obtidos os momentos de inércia para serem utilizados nos arquivos **URDF** possibilitando a simulação adequada da dinâmica do sistema.

### 3.4.2 ROS

O **ROS** contém duas ferramentas para a visualização da simulação. A primeira, o *Gazebo* é um software feito em duas partes, onde uma é dita *server* pois é o responsável por processar todas as informações relacionadas a dinâmica dos objetos, a luminosidade no ambiente, e todos os outros agentes físicos presentes na cena representada. A outra parte é o *graphical client*, que dispõe de uma interface gráfica **Graphic User Interface (GUI)** responsável por gerar para o usuário o visual da cena (**GAZEBO...**, s.d.). A outra ferramenta é *RViz*, destinada a fornecer informações numéricas de maneira gráfica para uma análise eficiente dos elementos dinâmicos que são geridos pelo *Gazebo server*. Dessa forma, os plugins do *Gazebo server* fornecem dados tanto para o *Gazebo client* quanto para o *RViz*. Um exemplo de exibição que foi bastante útil no projeto foram os eixos coordenados dos frames de cada uma das juntas. Ou a imagem capturada pelo plugin de câmera, permitindo ao usuário ver o que era gerado pela câmera artificial, que por sua vez permitiu avaliar o quanto bem estava funcionando a detecção da **AR tag**.

### 3.4.3 MoveIt

A descrição dos elementos físicos de todo o sistema foi feita a partir de arquivos **Extensible Markup Language (XML)**, ou mais especificamente **XML macros (XACRO)**. Estes arquivos são uma versão generalizada dos **URDF**, permitindo o uso de expressões matemáticas simples e as já mencionadas macros. Foram também empregados na atuação do software de planejamento dinâmico da movimentação do manipulador, *MoveIt*. Este software calculou as interações entre os elos, avaliando possíveis colisões. Ele também adicionou aos arquivos **XACRO** parâmetros que permitiram que as juntas fossem controladas por um plugin do *Gazebo*. Foram criados então dois pacotes - um para a descrição do manipulador e outro contendo as informações agregadas ao primeiro pelo *MoveIt*.

### 3.4.4 Pacote *bir\_marker\_localization*

Após a concepção de um modelo controlável em ambiente simulado, pôde-se programar a sua busca pela **AR tag** e implementar o pacote responsável pela sua detecção. Neste

pacote, um arquivo de configuração detém as informações sobre a *tag* utilizada, pois a mesma precisa ser previamente esperada para que a detecção ocorra. Este arquivo também contém parâmetros de câmera e as transformadas ([TF..., s.d.](#)) pertinentes aos objetos que se deseja conhecer a posição. A premissa do pacote é que avistando uma *tag* é possível estimar a posição de uma parte do manipulador em relação a ela. A versatilidade contida em “parte qualquer” é devida ao fato de que todos os elos do manipulador sendo rígidos, de comprimento fixo, é sempre possível saber a posição relativa entre todos eles. Dessa forma a pose do efetuador é determinada e o programa de posicionamento entra em ação para premir a botoeira de emergência.

### 3.5 Testes realizados e resultados

Estatística é a ciéncia que nos ajuda a fazer decisões e tirar conclusões na presença de variabilidade ([MONTGOMERY, 2012](#)). Segundo ([LENTH, 2001](#)), determinar o tamanho das amostras é um passo importante para planejar o estudo estatístico. A quantidade de amostras deve ser o suficiente para que cada uma tenha significância estatística.

Considerando a premissa que o manipulador possa interagir com uma *AR tag* independente de sua orientação no espaço e que a mesma possa ser vista pela câmera no ciclo de escaneamento, estabelecemos duas posições para os testes de localização: a caixa na horizontal e na vertical, ambas em relação à mesa. Vale salientar que a posição que o manipulador deseja alcançar está a frente do botão, para que não ocorram colisões desnecessárias.

Utilizando o tempo de simulação do *Gazebo* como recurso, foram selecionados 30 amostras para cada posição. Cada amostra pode ser dividida em duas partes: tempo em segundos da posição *Home* até o sistema identificar a *AR tag* e o tempo em segundos da identificação até o posicionamento final. Foi gerado um histograma de cada parte com uma linha que representa sua densidade. Depois, foi calculado cada média e aplicado o teste t de Student para identificar se os valores apresentam um nível de confiabilidade de 95% e uma distribuição normal a partir da mediana da população ao resultar em um *p-value* maior 0.05, segundo ([MONTGOMERY, 2012](#)).

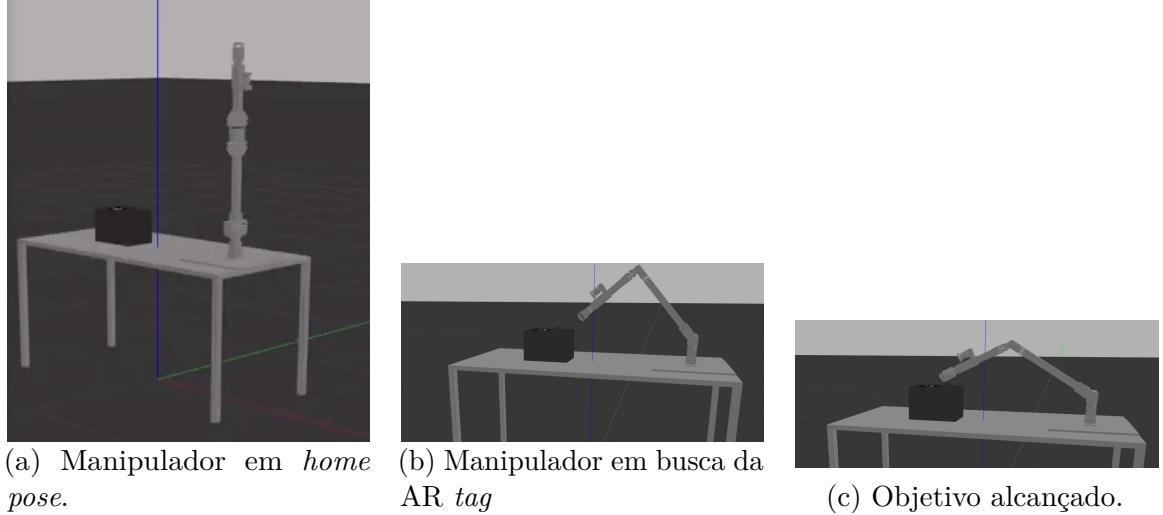
A média de tempo de deslocamento da posição *Home* até a identificação da *AR tag* é de 12.48 segundos e a sua mediana é de 12.42 segundos, a partir dos dados da tabela 3. Seu desvio padrão é de 0.2526 e seu teste t resultou em um *p-value* de 0.1888, apresentando 95% de confiabilidade entre os valores de 12.38774 segundos e 12.57639 segundos. Estes valores são aceitáveis pois o ciclo de escaneamento é programado para uma mesma movimentação, e a caixa estava na mesma posição durante as 30 amostras. Houveram dois valores atípicos nas amostras de número 5 e 21 que contribuem para o histograma, visto na figura 23, ser deslocado para a esquerda.

Tabela 3: Testes do manipulador em curso ao objetivo - caixa horizontal.

Amostra	<i>Home-Scan</i> (s)	<i>Scan-Goal</i> (s)
1	12.539	2.478
2	12.419	2.972
3	12.34	2.769
4	12.516	5.345
5	13.224	4.543
6	12.425	4.364
7	12.416	2.6
8	12.413	4.734
9	12.645	5.539
10	12.345	5.331
11	12.319	3.094
12	12.317	2.762
13	12.426	4.936
14	12.226	5.79
15	12.318	5.979
16	12.334	2.605
17	12.421	2.547
18	12.549	2.516
19	12.354	5.134
20	12.252	4.926
21	13.406	5.368
22	12.311	1.972
23	12.61	4.557
24	12.356	5.567
25	12.616	2.259
26	12.414	2.336
27	12.427	5.608
28	12.439	4.526
29	12.629	4.931
30	12.456	2.531

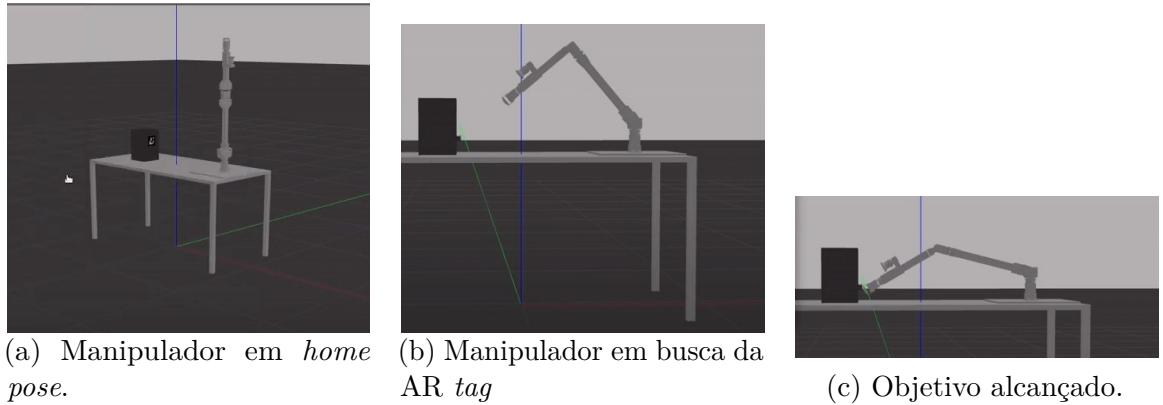
Fonte: Os autores.

Figura 21: Manipulador em curso ao objetivo - caixa horizontal.



Fonte: Os autores.

Figura 22: Manipulador em curso ao objetivo - caixa vertical.



Fonte: Os autores.

A média de tempo de deslocamento da posição de identificação da *AR tag* até o botão é de 4.021 segundos e a sua mediana é de 4.535 segundos. Seu teste t resultou em um *p-value* de 0.04597. A figura 24 mostra o histograma que não apresenta uma distribuição próxima a normal devido a cinemática do manipulador apresentar diversas soluções para resolver o problema da trajetória. As amostras apresentam um valor mínimo de 1.972 segundos e um valor máximo de 5.979 segundos. Seu desvio padrão é de 1.351131, um valor muito maior se comparado com a atividade anterior.

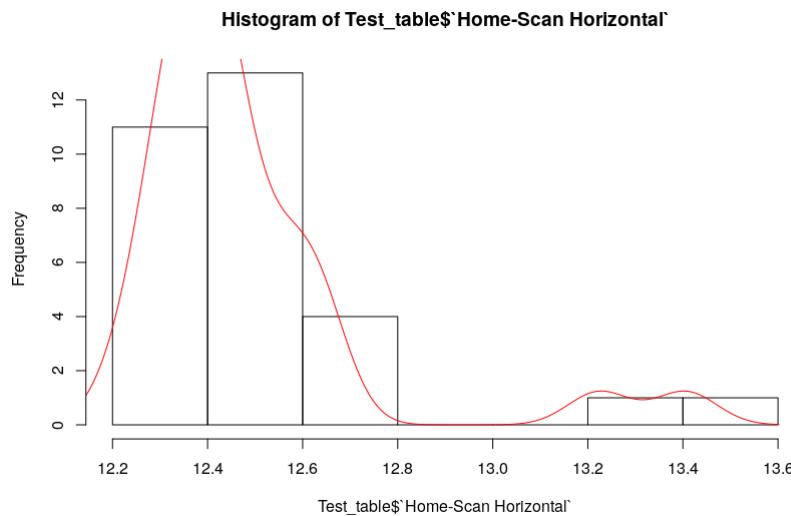
A média de tempo de deslocamento da posição *Home* até a identificação da *AR tag* é de 12.54 segundos e a sua mediana é de 12.42 segundos, a partir dos dados da tabela 4. Seu desvio padrão é de 0.3984279 e seu teste t resultou em um *p-value* de 0.101, apresentando 95% de confiabilidade entre os valores de 12.39442 segundos e 12.69198 segundos. A

Tabela 4: Testes do manipulador em curso ao objetivo - caixa vertical.

Amostra	<i>Home-Scan</i> (s)	<i>Scan-Goal</i> (s)
1	12.439	2.715
2	12.239	2.329
3	13.609	2.525
4	12.321	2.33
5	12.226	2.704
6	12.331	2.726
7	12.452	2.719
8	12.411	2.31
9	12.317	2.72
10	12.406	2.725
11	12.411	2.723
12	12.416	2.705
13	13.404	2.724
14	13.109	2.323
15	12.416	2.72
16	12.548	2.714
17	12.638	2.719
18	13.812	2.719
19	12.319	2.721
20	12.445	2.716
21	12.335	2.323
22	12.345	2.721
23	12.219	2.722
24	12.319	2.724
25	12.444	2.713
26	12.42	2.311
27	12.458	2.319
28	12.533	2.71
29	12.525	2.707
30	12.429	2.319

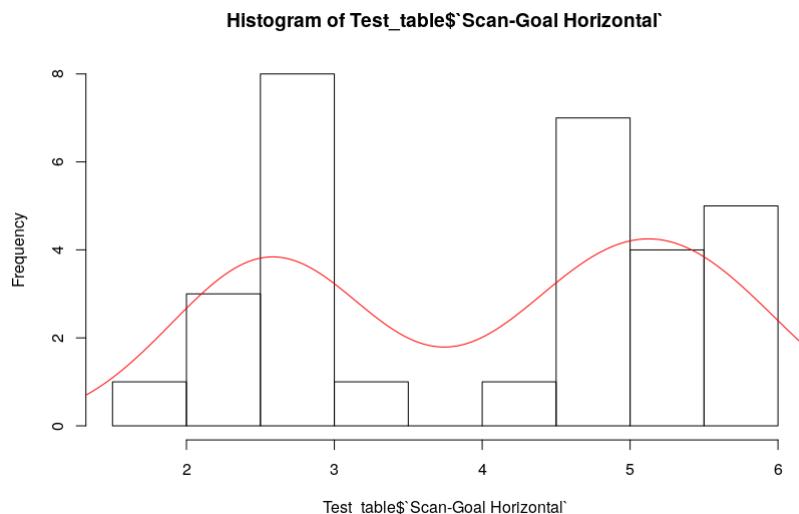
Fonte: Os autores.

Figura 23: Histograma e densidade das amostras da movimentação da posição *Home* até identificação da *tag* - caixa horizontal.



Fonte: Os autores.

Figura 24: Histograma e densidade das amostras da movimentação do fim do ciclo de escaneamento até o botão - caixa horizontal.

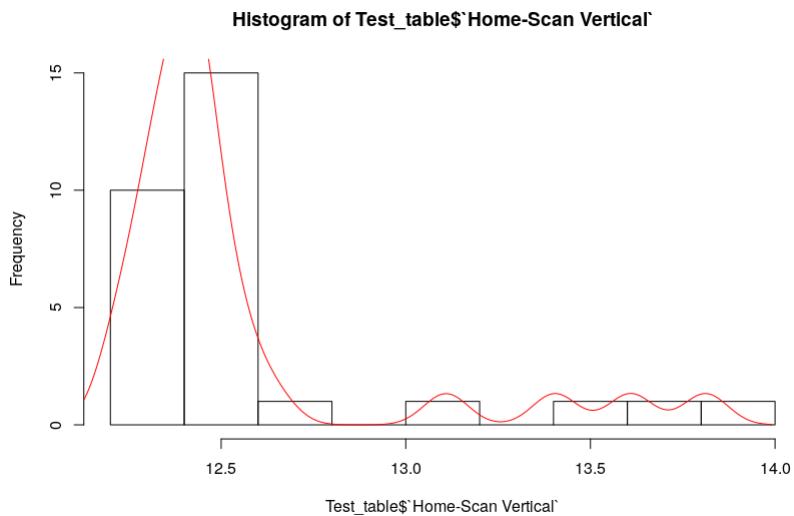


Fonte: Os autores.

justificativa dos valores se mantém semelhante com a mesma atividade representada na caixa horizontal. Porém, pode-se perceber que a quantidade de valores atípicos a mais influenciaram o valor do desvio padrão da amostra, apesar desta atividade ser mais precisa. Houveram quatro valores atípicos nas amostras de número 3, 12, 13 e 18 que contribuem para o histograma, como pode ser visto na figura 25, ser deslocado para a esquerda.

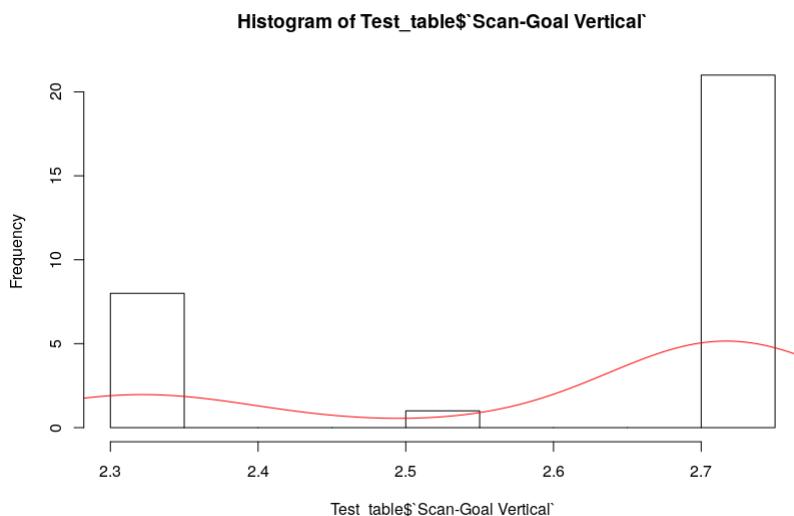
A média de tempo de deslocamento da posição de identificação da *AR tag* até o botão

Figura 25: Histograma e densidade das amostras da movimentação da posição *Home* até identificação da *tag* - caixa vertical.



Fonte: Os autores.

Figura 26: Histograma e densidade das amostras da movimentação do fim do ciclo de escaneamento até o botão - caixa vertical.



Fonte: Os autores.

é de 2.605 segundos e a sua mediana é de 2.715 segundos. Seu teste t resultou em um *p-value* de 0.002114. O histograma 26 não apresenta uma distribuição próxima a normal devido a cinemática do manipulador apresentar diversas soluções para resolver o problema da trajetória, mesma justificativa para a caixa na horizontal. As amostras apresentam um valor mínimo de 2.31 segundos e um valor máximo de 2.726 segundos. Vale ressaltar que a frequência de valores é bem maior perto do seu valor máximo, resultando em um desvio padrão considerado baixo para a atividade (0.1781923).



## 4 CONFIABILIDADE DO SISTEMA

Após conhecimento dos subsistemas presentes no desenvolvimento do manipulador robótico bem como dos seus componentes apresentados na estrutura analítica do protótipo, foi elaborado um plano de aplicação da *Failure Mode and Effect Criticality Analysis* (**FMECA**) para identificar os modos de falhas que podem ocorrer no produto ou protótipo. Segundo o formulário **FMECA** apresentado nas tabela 8 estão indicados: a) Funções do componente analisado; b) Modos de falhas que possam vir a ocorrer; c) Os possíveis efeitos causados no sistema pelas falhas; d) Uma análise de causas possíveis que gerariam os erros analisados; e) São apresentadas algumas ações recomendadas para que a falha seja corrigida ou minimizada; f) A classificação da competência em cada caso.

### 4.1 Análise dos modos e efeitos de falhas

Para uma realizar uma análise de riscos foram parametrizadas inicialmente duas avaliações dos modos e efeitos de falha. A primeira delas foi a severidade da falha, que indica o quanto ela pode comprometer o funcionamento global do sistema. Na segunda avaliação, é feita uma análise quanto a detectabilidade da falha, que é o quanto essa falha pode ser facilmente detectada, para assim aplicar o plano de ação recomendado para minimizá-la. Esses critérios são apresentados respectivamente nas tabelas 5 e 6 a seguir.

Tabela 5: Classificação de severidade de falhas.

Classificação	Severidade	Critério de classificação
1	Mínima	O cliente mal percebe que a falha ocorre
2	Pequena	Ligeira deterioração no desempenho com leve descontentamento do cliente
3	Moderada	Deterioração significativa do desempenho de um sistema com descontentamento do cliente
4	Alta	Sistema deixa de funcionar e grande é o descontentamento do cliente
5	Muito Alta	Agravamento do anterior com risco a segurança

Fonte: Os autores.

A tabela 8 condensa os estudos de **FMECA** realizados para os subsistemas que compõem o projeto, apresentando uma análise de severidade (S), detecção (D) e ocorrência da falha (O). O índice de criticidade é dado mediante cálculo dos demais índices apresentados, para assim realizar uma priorização dos modos de falha.

Tabela 6: Classificação de facilidade de detecção de falhas.

Classificação	Capacidade de Detecção	Critério de classificação
1	Muito grande	Certamente será detectado
2	Grande	Grande possibilidade de ser detectado
3	Moderada	Provavelmente será detectado
4	Pequena	Provavelmente não será detectado
5	Muito pequena	Certamente não será detectado

Fonte: Os autores.

Tabela 7: Classificação da probabilidade de ocorrência de falhas.

Valor	Ocorrência	Critério
1	Remota	1:1.000.000
2	Pequena	1:10.000
3	Moderada	1:1.000
4	Alta	1:10
5	Muito alta	>1:8

Fonte: Os autores.

Como observado na tabela 8 é apresentado o índice de criticidade mediante cálculo dos índices de severidade, detecção e ocorrência. Esse índice permite enfatizar os subsistemas de atuação e de potência como críticos para o manipulador robótico desenvolvido, pois esses apresentam elevado índice de criticidade em um ou mais modos de falha.

Tabela 8: Matriz de modos e efeitos de falhas para os subsistemas do manipulador

Função(es)	Modo(s) de falha em potencial	Efeito(s) potencial(is) da falha	S	Causa(s) potencial(is) / mecanismo(s) de falha(s)	D	O	Ação(es) recomendada(s)	Competência	S x D x O
Atuação das juntas do manipulador ( motores).	Não funciona mento dos atuadores	Perda do controle de uma ou mais juntas do manipulador	4	Problemas elétricos Falha na comunicação com o controlador	1	3	Inspeções periódicas Inspeções periódicas	Elétrica Elétrica	12 36
		Perda de movimentação do manipulador	4		3	3			
	Movimentação não compatível com a programação	Perda de desempenho do manipulador	3	Falhas no software Vibrações mecânicas Falhas na comunicação com o controlador	3	2	Inspeções periódicas Isolamento da área de trabalho Sistema de detecção de obstáculos e isolamento da área de trabalho	Elétrica Mecânica Elétrica	18 12 45
		Trajetória incompatível	3		4	1			
		Colisão com demais objetos	5		3	3			
Detecção da tag (câmera).	Não detecção da tag	Perda da aquisição de posicionamento e orientação	3	Problemas elétricos Falha na comunicação com o controlador	2	3	Inspeções de conexões e cabos Inspeções de conexões e cabos	Elétrica Elétrica	18 36
		Não realização da tarefa	4		3	3			
	Coleta de dados inconsistentes	Comprometimento dos dados obtidos	3	Problemas de visibilidade da tag Calibração incorreta Danificação da lente da câmera	2	3	Certificar luminosidade e ausência de obstáculos Executar varias verificações Inspeção e instalação correta	Elétrica Elétrica Mecânica	18 27 6
		Erro no posicionamento e orientação	3		3	3			
		Perda de dados	3		2	1			
Processamento dos subsistemas do manipulador (NUC).	Falha no processamento	Não funcionamento do manipulador	4	Falha no sistema operacional	2	3	Executar testes de verificação Inspeções de conexões e cabos	Elétrica Elétrica	24 36
			4	Falha na comunicação com o controlador	3	3			
		Perda de dados	3	Falha da memória interna	3	3	Rotina de teste	Elétrica	27
	Não converter níveis de tensão	Não alimentação do sistema	4	Falha nos componentes do conversor	4	2	Executar testes de verificação	Elétrica	32
Converter níveis de tensão DC (Conversor DC/DC)	Conversão de níveis de tensão incompatíveis com os atuadores	Não funcionamento dos sistemas dependentes do conversor	4	Falha nos componentes do conversor	4	2	Executar testes de verificação Executar testes de verificação	Elétrica Elétrica	32 32
		Danificação dos sistemas	4	Uso de tensão de entrada não especificada	2	4			
	Superaquecimento	Danificação do conversor	5	Mau contato nos terminais	3	3	Executar testes de verificação	Elétrica	45
Estabilizar níveis de tensão do sistema (Conversor DC/DC)	Não estabilizar níveis de tensão do sistema	Geração de ruído no sistema	4	Falha nos componentes do conversor	4	2	Executar testes de verificação	Elétrica	32

Fonte:Os autores.



## 5 GESTÃO DO CONHECIMENTO

### 5.1 Lições aprendidas

Durante o processo de desenvolvimento de um projeto diversos são os eventos imprevistos, podendo ser causados por aleatoriedades ou inexperiências. Assim sendo, torna-se válido ter um memorial coerente destes eventos, visando o crescimento da equipe envolvida num dado projeto e nos que sucederão.

Tabela 9: Condensação de eventos marcantes do projeto.

Lições aprendidas					
Tema	Fase	Impacto	O que ocorreu?	Como resolveu?	Resultados
Aquisição	Execução	Negativo	Erro na execução de furos nas chapas devido a problemas de medição	Foi realizados vários testes de medição antes da confecção das peças	Confeção correta das peças
Gestão	Execução	Negativo	Erro na configuração da comunicação dos motores, gerando perda de tempo de serviço.	Realização de uma nova configuração dos motores	Motores foram reconfigurados
Gestão	Planejamento	Negativo	Peça dimensionada de forma não compatível com a estrutura	Nova modelagem da peça levando em consideração os erros cometidos	Modelagem correta da peça
Gestão	Planejamento	Negativo	Perda de tempo por mudanças no tipo de AR Tag utilizada no projeto	Foi dedicado tempo extra para minimizar atrasos	Prosseguimento no projeto
Gestão	Planejamento	Positivo	Reuniões diárias para atualização do andamento do projeto e compartilhamento de experiências	Foram realizadas as reuniões diárias	Forte impacto positivo no andamento do projeto

### 5.2 Guia de uso

A configuração recomendada é possuir um computador com sistema operacional Ubuntu 18.04, *framework ROS* ([ROBOTICS, s.d.](#)) instalado e o pacote *OpenCV 3.3.1* ([CHANDEL, 2017](#)).

Siga os seguintes passos para a instalação das dependências do projeto:

Abra seu terminal para instalar os pacotes:

Instalar o pacote *ROS Controller*:

```
1 $ sudo apt-get install ros-melodic-controller
```

Instalar o pacote RQT:

```
1 $ sudo apt-get install ros-melodic-rqt*
```

Instalar o pacote de juntas:

```
1 $ sudo apt-get install ros-melodic-joint*
```

Instalar os pacotes de controle do Gazebo:

```
1 $ sudo apt-get install ros-melodic-gazebo-ros-control
```

Instalar o pacote de cinemática Trac-IK:

```
1 $ sudo apt-get install ros-melodic-trac-ik
```

Instalar o pacote *MoveIt*:

```
1 $ sudo apt-get install ros-melodic-moveit*
```

Crie um ambiente de trabalho no seu computador digitando em seu terminal:

```
1 $ mkdir -p catkin_ws/src  
2 $ cd catkin_ws  
3 $ catkin_make
```

Entre na pasta "src" e clone o pacote externo BIR-Marker-Localization:

```
1 $ cd src  
2 $ git clone -b final_settings https://github.com/Brazilian-  
Institute-of-Robotics/bir_marker_localization/tree/  
final_settings.git
```

Clone o pacote do manipulador RAJA em seguida:

```
1 $ git clone https://github.com/Brazilian-Institute-of-Robotics/  
raja_manipulator.git
```

Retorne para a pasta raiz do ambiente de trabalho e digite o comando para compilar os pacotes:

```
1 $ cd ..  
2 $ catkin_make
```

Utilização:

A aplicação é dividida em duas partes, então é necessário abrir dois terminais para que ela funcione.

Em ambos os terminais, navegue até a pasta do ambiente de trabalho e ative o ambiente utilizando o comando:

```
1 $ source devel/setup.bash
```

No primeiro terminal, digite:

```
1 $ roslaunch raja_gazebo gazebo.launch
```

Após a aplicação do Gazebo ser executada, despause o sistema apertando o botão *play* no canto inferior da interface gráfica do Gazebo. No segundo terminal digite:

```
1 $ rosrun raja_move_interface move_interface.py
```

A aplicação fará com o que o manipulador saia da posição *Home* e procure uma *tag* ligada à uma caixa e um botão de emergência. Ao achá-la, o manipulador posiciona sua ferramenta no em cima do botão de emergência. Caso não seja encontrada até o fim do ciclo de escaneamento, o robô volta à sua posição inicial.



## 6 CONCLUSÃO

O objetivo deste manipulador é levar seu *end effector* até um ponto no espaço que representa a posição do botão de emergência da caixa que está no mesmo ambiente de trabalho. Este objetivo foi concluído com sucesso. Esta etapa ambientada no sistema de simulação Gazebo, e apresentou resultados esperados nas atividades de inspeção visual, busca, identificação e planejamento e execução de trajetória. A atividade de orientação da ferramenta em relação ao objetivo final requer ajustes.

O uso de **ARUCO** *tags* provou-se uma eficiente ferramenta para identificação e localização de objetos externos ao robô utilizando câmera, trazendo a caixa, mesa e manipulador para o mesmo referencial de trabalho.

Ainda se está testando o comportamento do manipulador para diferentes ferramentas e atividades mais complexas, que possam requerer cálculos mais complexos de trajetória ou diferentes pontos no espaço.

A metodologia deste relatório será utilizada para a construção física do manipulador RAJA. Este será aplicado em um ambiente de trabalho similar para realizar as mesmas atividades e ter seus parâmetros atualizados. A próxima etapa será implementar o manipulador em uma base móvel para que ele possa interagir com um objeto em uma área externa.



## REFERÊNCIAS

- ALL On Robots - Cylindrical robot type. s.d. <<http://www.allonrobots.com/cylindrical-robot.html>>. Acessado: 14-04-2020. Citado na página 16.
- ANDERSON, M. *Introduction to the Robot Operating System (ROS) Middleware*. s.d. <[https://elinux.org/images/1/11/IntroductionToROS\\_Anderson.pdf](https://elinux.org/images/1/11/IntroductionToROS_Anderson.pdf)>. Acesso em: 28-04-2020. Citado na página 21.
- ATHANI, V. V. *Stepper Motors: Fundamentals, applications and design*. New Delhi: New Age International (P) Limited, 2005. Citado na página 18.
- AUTOMATION, O. I. *eCobra 800 Lite / Standard / Pro*. 2016. <<http://www.ia.omron.com/products/family/3517/dimension.html>>. Acessado: 14-04-2020. Citado na página 17.
- BARINKA, I. R. B. L. *Inverse Kinematics - Basic Methods*. Prague, Czech Republic: [s.n.], s.d. Citado na página 21.
- BRASIL perde a corrida da automação industrial. 2019. <<https://valor.globo.com/brasil/noticia/2019/07/29/brasil-perde-a-corrida-da-automacao-industrial.ghtml>>. Accessed: 2020-04-23. Citado na página 12.
- C3, C. d. C. C. *Introdução Robôs Manipuladores*. 2015. <<https://pt.slideshare.net/giselemoraessimas/sr-aula1-robosindustriais>>. Acessado: 14-04-2020. Citado na página 17.
- CARRARA, V. Introdução à robótica industrial. *INPE-Instituto Nacional de Pesquisas Espaciais, São José dos Campos*, 2015. Citado na página 15.
- CHANDEL, V. S. *Install OpenCV3 on Ubuntu*. 2017. <<https://www.learnopencv.com/install-opencv3-on-ubuntu/>>. Acesso em: 28-04-2020. Citado na página 47.
- COLLINS, D. *What is a Cartesian robot?* 2018. <<https://www.linearmotiontips.com/what-is-a-cartesian-robot/>>. Acesso em: 27-04-2020. Citado na página 16.
- ERTHAL, J. L. Estudo de métodos para a solução da cinemática inversa de robôs industriais para implementação computacional. *UFSC, Programa de Pós-Graduação*, 10 1992. Citado na página 11.
- ERTHAL, J. L. *Estudo de métodos para a solução da cinemática inversa de robôs industriais para implementação computacional*. Florianópolis: Departamento de Engenharia Mecânica, 1992. Citado 3 vezes nas páginas 19, 20 e 21.
- EXECUTIVE Summary World Robotics 2018 Industrial Robots. 2019. <[https://ifr.org/downloads/press2018/Executive\\_Summary\\_WR\\_2018\\_Industrial\\_Robots.pdf](https://ifr.org/downloads/press2018/Executive_Summary_WR_2018_Industrial_Robots.pdf)>. Accessed: 2020-04-23. Citado na página 12.
- FEDERICA Web Learning. s.d. <[https://mooc.federica.eu/l/introduction\\_to\\_robotics#22](https://mooc.federica.eu/l/introduction_to_robotics#22)>. Acessado: 14-04-2020. Citado na página 18.
- GAZEBO. *History*. 2014. <<http://gazebosim.org/>>. Acesso em: 28-04-2020. Citado na página 22.

GAZEBO Components. s.d. [Http://gazebosim.org/tutorials?tut=quick\\_start](Http://gazebosim.org/tutorials?tut=quick_start). Acessado: 24-04-2020. Citado na página 35.

HERNANDES, A. G. *Estudo da Modelagem Robótica*. Londrina, Paraná: [s.n.], 2014. Citado na página 21.

IEEE Guide to the world of robots. s.d. <<https://robots.ieee.org/robots/unimate/>>. Acesso em: 15-04-2020. Citado na página 15.

INDUSTRIAL Robots: Robot Investment Reaches Record 16.5 billion USD. 2019. <<https://ifr.org/ifr-press-releases/news/robot-investment-reaches-record-16.5-billion-usd/>>. Acesso em: 15-04-2020. Citado na página 11.

INTERNATION Federation of Robotics. 2018. <<https://ifr.org/ifr-press-releases/news/robot-investment-reaches-record-16.5-billion-usd>>. Accessed: 2020-04-23. Citado na página 12.

ISO-8373. *Robots and robotic devices — Vocabulary*. Geneva, CH, 2012. Citado na página 13.

LENTH, R. V. Some practical guidelines for effective sample size determination. 2001. Citado na página 36.

MONTGOMERY, D. C. *Estatística Aplicada e Probabilidade para Engenheiros*. 5th. ed. [S.l.]: LTC, 2012. Citado na página 36.

MOVEIT. *Concepts*. s.d. <<https://moveit.ros.org/documentation/concepts/>>. Acesso em: 28-04-2020. Citado na página 23.

MOVEIT. *Frequently Asked Questions*. s.d. <<https://moveit.ros.org/documentation/faqs/>>. Acesso em: 28-04-2020. Citado na página 22.

NAKAMURA, H. H. Y. *Inverse Kinematic Solutions With Singularity Robustness for Robot Manipulator Control*. Universidade de Aveiro: Journal of Dynamic Systems, Measurement and Control, 1986. Citado na página 19.

OBR Equipamentos industriais. s.d. <<http://www.obr.com.br/produtos/>>. Acessado: 14-04-2020. Citado 2 vezes nas páginas 26 e 27.

OPENCV. *About OpenCV*. 2020. <<https://opencv.org/about/>>. Acesso em: 28-04-2020. Citado na página 22.

PIMENTA, T. T. Controle de manipuladores robóticos. *PUC-Rio de Janeiro, Departamento de Engenharia de Controle e automação*, 12 2009. Citado na página 11.

RAO, P. K. B. S. S. *Optimization in the Design and Control of Robotic Manipulators: A Survey*. [S.l.: s.n.], 1989. Citado na página 21.

ROBOTICS, O. *Install ROS Melodic*. s.d. <<http://wiki.ros.org/melodic/Installation/Ubuntu>>. Acesso em: 28-04-2020. Citado na página 47.

ROBOTIS. s.d. <<http://www.robotis.us/>>. Acessado: 12-04-2020. Citado 2 vezes nas páginas 28 e 29.

ROBOTIS MX-106-R. s.d. <<http://www.robotis.us/dynamixel-mx-106r/>>. Acessado: 28-04-2020. Citado na página 29.

ROBOTIS PH42-020-S300-R. s.d. <<http://emanual.robotis.com/docs/en/dxl/p/ph42-020-s300-r/>>. Acessado: 28-04-2020. Citado na página 29.

ROBOTIS PH54-200-S500-R. s.d. <<http://www.robotis.us/dynamixel-ph54-200-s500-r/>>. Acessado: 28-04-2020. Citado na página 29.

ROMANO, V. F. *Robótica Industrial: Aplicação na indústria de manufatura e processos*. São Paulo: Edgar Blucher, 2002. v. 256. Citado 2 vezes nas páginas 11 e 13.

ROS Introduction. s.d. <<http://wiki.ros.org/ROS/Introduction>>. Acessado: 24-04-2020. Citado na página 34.

RUOCCHI, S. R. *Robot sensors and transducers*. [S.l.]: Springer Science & Business Media, 2013. Citado na página 14.

SANTOS, F. L. R. Matheus Araújo dos. *Modelagem Matemática e Simulação da Cinemática Direta do Robô Manipulador com Seis Juntas Rotativas – 6 GL*. Campo Mourão, PR: III Simpósio de Tecnologia e Engenharia Eletrônica - III SIMTEEL, 2016. Citado na página 19.

SANTOS, V. M. F. *Robótica Industrial: Apontamentos teóricos, Exercícios para aula pratica e Problemas de exames resolvidos*. Universidade de Aveiro: Departamento de Engenharia Mecânica, 2004. Citado 3 vezes nas páginas 19, 20 e 21.

SLOTINE HARUHIKO ASADA, H. A. J. J. E. *Robot Analysis and Control*. New York: [s.n.], 1986. Citado na página 20.

SPONG, M. W.; HUTCHINSON, S.; VIDYASAGAR, M. Robot modeling and control. Inc., ISBN-100-471-649, Jon Wiley & Sons, 2005. Citado na página 11.

TELEDYNE Dalsa. s.d. <<https://www.teledynedalsa.com/en/products/imaging/cameras/genie-nano-1gige/>>. Acessado: 12-04-2020. Citado na página 30.

TF - ROS. s.d. <<http://wiki.ros.org/tf>>. Acessado: 24-04-2020. Citado na página 36.

WEBER, F. et al. State of the art on manipulators. *Mobile Robotics in Health Care Services*, v. 7, p. 227, 01 2001. Citado 3 vezes nas páginas 14, 18 e 19.

WIKIPÉDIA. *Robot Operating System*. 2019. <[https://pt.wikipedia.org/wiki/Robot\\_Operating\\_System](https://pt.wikipedia.org/wiki/Robot_Operating_System)>. Acesso em: 28-04-2020. Citado na página 21.

WIKIPÉDIA. *Motor*. 2020. <<https://pt.wikipedia.org/wiki/Motor>>. Acessado: 27-04-2020. Citado na página 18.



## APÊNDICE A

### Tabelas

## Apêndice A

---

Lições aprendidas						
Tema	Fase	Impacto	O que ocorreu?	Como resolveu?	Resultados	Recomendações para os próximos projetos
Tecnológico	Execução	Negativo	Erro na execução de furos nas chapas devido a problemas de medição	Foi realizados vários testes de medição antes da confecção das peças	A peça foi corrigida, porém houve atraso na tarefa	Recomenda-se a verificação das dimensões mais de uma vez
Tecnológico	Planejamento	Negativo	Ausência de uma metodologia de trabalho	Reunião com objetivo de definir uma metodologia	A metodologia foi definida e atrasos nas atividades foram minimizados	Recomenda-se definir antes da execução uma metodologia de trabalho
Tecnológico	Planejamento	Negativo	Peca dimensionada de forma não compatível com a estrutura	Remodelagem da peça	Modelagem correta; atraso no cronograma	Recomenda-se a verificação das dimensões necessárias para a peça
Tecnológico	Planejamento	Negativo	Necessidade de mudança no planejamento de detecção, modificando a AR Tag utilizada	Foi realizada a mudança no planejamento	Foi dado continuidade no projeto, porém com atraso significativo na tarefa	Analizar os métodos disponíveis antes do planejamento
Gestão	Planejamento	Positiva	Reuniões diárias para atualização do andamento do projeto e compartilhamento de experiências	Foram realizadas as reuniões diariamente	Forte impacto no andamento do projeto	Realização de reuniões diárias com toda equipe do projeto

## APÊNDICE B

**Script em Octave para o cálculo do *workspace***

```
1 clc
2 clear
3 pkg load syms
4
5 syms theta1 d1 a1 alpha1 ...
6     theta2 d2 a2 alpha2 ...
7     theta3 d3 a3 alpha3
8
9 d2 = 0;
10 d3 = 0;
11 a1 = 0;
12 alpha1 = pi/2;
13 alpha2 = 0;
14 alpha3 = 0;
15
16 A_01 = [
17     cos(theta1) -sin(theta1)*cos(alpha1) sin(theta1)*sin(alpha1)
18         a1*cos(theta1);
19     sin(theta1) cos(theta1)*cos(alpha1) -cos(theta1)*sin(alpha1)
20         a1*sin(theta1);
21     0           sin(alpha1)           cos(alpha1)      d1;
22     0           0                   0               1];
23
24 A_12 = [
25     cos(theta2) -sin(theta2)*cos(alpha2) sin(theta2)*sin(alpha2)
26         a2*cos(theta2);
27     sin(theta2) cos(theta2)*cos(alpha2) -cos(theta2)*sin(alpha2)
28         a2*sin(theta2);
29     0           sin(alpha2)           cos(alpha2)      d2;
30     0           0                   0               1];
31
32 A_23 = [
33     cos(theta3) -sin(theta3)*cos(alpha3) sin(theta3)*sin(alpha3)
34         a3*cos(theta3);
35     sin(theta3) cos(theta3)*cos(alpha3) -cos(theta3)*sin(alpha3)
36         a3*sin(theta3);
37     0           sin(alpha3)           cos(alpha3)      d3;
38     0           0                   0               1];
39
40 A_03 = simplify(A_01 * A_12 * A_23)
41
42 pose_x = A_03[13]
```

```
37 pose_y = A_03[14]
38 pose_z = A_03[15]
39
40 d1 = 0.14;
41 a2 = 0.56;
42 a3 = 0.46;
43
44 i = 1;
45 k = 1;
46 theta1 = 0;
47 for theta1 = -pi:0.1:pi
    for theta2 = -pi:0.3:pi;
        for theta3 = -pi:0.3:pi
            aux_x = pose_x;
            aux_y = pose_y;
            aux_z = pose_z;
            if aux_z > 0 && aux_y > 0
                x(i) = aux_x;
                y(i) = aux_y;
                z(i) = aux_z;
                i = i + 1;
            end
            k = k + 1;
            disp(k)
        end
    end
end
64
65 scatter3(x, y, z)
66 xlabel("y")
67 ylabel("z")
68 zlabel("z")
```



---

## **MANIPULADOR ROBÓTICO JEROTIMON**

---

## **MANIPULADOR ROBÓTICO JEROTIMON**

Relatório do Projeto Manipuladores Inteligentes

### **Autores:**

Jean Paulo Silva  
Jéssica Lima Motta  
Leonardo Mendes de Souza Lima  
Miguel Felipe Nery Vieira  
Rodrigo Formiga Farias  
Vinicius José Gomes de Araújo Felismino

### **Facilitadores:**

Lucas Cruz da Silva  
Rebeca Tourinho Lima  
Tiago Pereira de Souza  
Marco Antonio dos Reis

**Salvador  
Bahia, Brasil**

**Novembro de 2020**



Título: Manipulador Robótico JeRoTIMON	
<b>PROD. TEC. CCRoSA - 001 / 2020</b>	Versão
<b>Classificação:</b> ( ) Confidencial (X) Restrito ( ) Uso Interno ( ) Público	01

**Informações Confidenciais** - Informações estratégicas para o CCRoSA e Senai Cimatec.

Seu manuseio é restrito a usuários previamente autorizados pelo Gestor da área.

**Informações Restritas** - Informação cujo conhecimento, manuseio e controle de acesso devem estar limitados a um grupo restrito de pesquisadores que necessitam utilizá-la para exercer suas atividades profissionais.

**Informações de Uso Interno** - São informações destinadas à utilização interna por pesquisadores e parceiros.

**Informações Públicas** - Informações que podem ser distribuídas ao público externo, o que, usualmente, é feito através dos canais apropriados.

### Dados Internacionais de Catalogação na Publicação (CIP)

Jean Paulo Silva Jéssica Lima Motta Leonardo Mendes de Souza Lima Miguel Felipe Nery Vieira Rodrigo Formiga Farias 000 Vinicius José Gomes de Araújo Felismino	Lucas Cruz da Silva Rebeca Tourinho Lima Tiago Pereira de Souza Marco Antonio dos Reis	Manipulador Robótico JeRoTIMON Salvador Bahia, Brasil Novembro de 2020	Keywords: 1. Manipulator. 2. Simulation. 3. Computer vision. 000
---	---	---	--



## SUMÁRIO EXECUTIVO

O projeto de Manipuladores - Desafio 2.2, também conhecido como **JeRoTIMON Manipulator** configura-se: sob o Programa de Formação de Novos Talentos do [Centro de Competência de Robótica e Sistemas Autônomos \(CCRoSA\)](#) do Serviço Nacional de Aprendizagem Industrial, Departamento Regional da Bahia - Senai/DR/BA, sendo este o principal fomentador do programa. O presente trabalho tem como impulsionador principal a capacitação de novos pesquisadores preparados para solucionar os mais diversos problemas relacionados a robótica e sistemas autônomos.

O projeto foi considerado como início técnico no dia 27 de Fevereiro de 2020, ocorrendo um hiato de desenvolvimento entre o período de 17 de março de 2020 a 07 de setembro de 2020 devido às medidas de isolamento social decorrentes da pandemia do novo coronavírus, havendo seu retorno no dia 08 de setembro de 2020.

O prazo de execução planejado foi de 50 dias.

## RESUMO

JeRoTIMON é um manipulador projetado, simulado e construído com o intuito de atender às demandas relacionadas ao reconhecimento de marcadores visuais e acionamento de interruptores, chaves ou botões. Posteriormente, este mesmo manipulador robótico será integrado ao robô *Warthog*, desenvolvido pela *Clearpath Robotics*, com o propósito de realizar a atividade de investigação em ambiente externo, desta vez participando de uma busca e desarme de bombas sem potencial destrutivo. O pacote de simulação do manipulador foi construído através do software *Gazebo* aliado ao *MoveIt* e a ferramenta de visualização *Rviz*, possibilitando assim que as atividades realizadas no mundo real tenham sido previamente testadas em ambiente simulado onde é possível analisar os movimentos e limitações do manipulador.

## ABSTRACT

JeRoTIMON is a manipulator designed, simulated and built in order to meet demands related to the recognition of visual markers and activation of interrupters, switches or buttons. Posteriorly, this manipulator robot arm will be attached in *Warthog*, robot designed by *Clearpath Robotics*, with the purpose of carrying out the research activity in an external environment, this time, integrating a searching and defusing non-destructive bombs. The manipulator simulation package was built using software *Gazebo*, allied to *Moveit* and *Rviz* visual tool, allowing that activities practiced in the real world have been previously tested in a computer environment where it is possible to analyze manipulator's movements and limitations.

## LISTA DE FIGURAS

Figura 1:	Elos e junta de um manipulador robótico.	18
Figura 2:	Arquitetura geral do sistema.	21
Figura 3:	Configuração D-H do manipulador JeRoTIMON.	24
Figura 4:	<i>Workspace</i> do manipulador no plano X-Y.	25
Figura 5:	<i>Workspace</i> do manipulador no plano X-Z.	25
Figura 6:	<i>Workspace</i> do manipulador no plano Y-Z.	26
Figura 7:	Estrutura analítica do protótipo.	27
Figura 8:	Fluxograma do sistema de escanamento.	28
Figura 9:	Fluxograma do sistema de planejamento e execução de trajetória.	30
Figura 10:	Modelos simulados do manipulador e do painel elétrico.	32
Figura 11:	Imagen capturada pela câmera RGB.	32
Figura 12:	Identificação dos motores.	33
Figura 13:	Base do manipulador.	34
Figura 14:	Vista lateral do <i>link</i> 2.	34
Figura 15:	Vista superior do <i>link</i> 2.	35
Figura 16:	Teledyne Genie Nano C2590.	36
Figura 17:	Lente 16mm C Series VIS-NIR.	36
Figura 18:	Suporte para fixação da câmera.	37
Figura 19:	Integração suporte-câmera-lente.	37
Figura 20:	UWE-12/10-Q12PB-C.	38
Figura 21:	U2D2.	39
Figura 22:	Caixa objetivo.	40
Figura 23:	Manipulador na posição home.	41
Figura 24:	Histograma do conjunto de dados sem detecção da <i>tag</i> .	47
Figura 25:	Boxplot do conjunto de dados sem detecção da <i>tag</i> .	47
Figura 26:	<i>Boxplot</i> da variância do tempo de busca entre os pontos A e B.	50
Figura 27:	<i>Boxplot</i> da variância do tempo total da missão entre os pontos A e B.	51
Figura 28:	Modelo completo teste R&R para tempo de busca.	52
Figura 29:	Modelo completo teste R&R para tempo total da missão.	52
Figura 30:	Gráficos do estudo R&R para tempo de busca.	53
Figura 31:	Árvore de falhas do sistema.	57
Figura 32:	Coordenadas.	75
Figura 33:	Link 0.	75
Figura 34:	Link 1.	76
Figura 35:	Link 2.	77

Figura 36: Link 3 . . . . .	77
Figura 37: Link 4 . . . . .	78
Figura 38: Link 5 . . . . .	79
Figura 39: Desenho técnico do JeRoTIMON, vistas: frontal, lateral esquerda e superior . . . . .	84
Figura 40: Desenho técnico do JeRoTIMON, vista isométrica. . . . .	85
Figura 41: Analise estática - condição de maior esforço exigido. . . . .	86
Figura 42: Analise estática - condição de maior esforço exigido (cargas no centróide). . . . .	86



## LISTA DE TABELAS

Tabela 1:	Especificações técnicas do manipulador JeRoTIMON.	23
Tabela 2:	Parâmetros D-H para o manipulador JeRoTIMON.	24
Tabela 3:	Parâmetros dos motores.	33
Tabela 4:	Parâmetros dos motores.	38
Tabela 5:	Torque das juntas.	39
Tabela 6:	Ângulos dos motores na posição home.	41
Tabela 7:	Modelo de experimentos para análise de 3 variáveis.	43
Tabela 8:	Modelo de experimentos ANOVA sem detecção da <i>tag</i> .	44
Tabela 9:	Modelo de experimentos ANOVA com detecção da <i>tag</i>	44
Tabela 10:	Resultados das amostras coletadas.	44
Tabela 11:	Resultados da análise de regressão linear.	45
Tabela 12:	Resultados da análise de regressão linear (variável algoritmo).	45
Tabela 13:	Resultados dos experimentos sem detecção da <i>tag</i> .	46
Tabela 14:	Resultados dos experimentos sem detecção da <i>tag</i> .	48
Tabela 15:	Resumo dos conjuntos de dados das variáveis de saída.	50
Tabela 16:	Tabela dos valores de $\rho$ -valor do teste ANOVA.	50
Tabela 17:	<i>FMECA</i> do sub-sistema de potência	55
Tabela 18:	<i>FMECA</i> do sub-sistema aquisição	55
Tabela 19:	<i>FMECA</i> do sub-sistema estrutural	56
Tabela 20:	<i>FMECA</i> do sub-sistema de atuação	56
Tabela 21:	<i>FMECA</i> do sub-sistema de processamento	56
Tabela 22:	Lições aprendidas	59
Tabela 23:	Tabela de suportes	81
Tabela 24:	Especificações do motor <i>Dynamixel HP42-020-S300-R</i>	119
Tabela 25:	Especificações do motor <i>Dynamixel PH54-100-S500-R</i>	123
Tabela 26:	Especificações do motor <i>Dynamixel PH54-200-S500-R</i>	127



## LISTA DE SÍMBOLOS E ABREVIATURAS

**AGV** Automated Guided Vehicle

**CAD** Computer Aided Design

**CCRoSA** Centro de Competência de Robótica e Sistemas Autônomos

**DoF** Degrees of Freedom

**FMECA** Failure Modes, Effects and Critically Analysis

**OMPL** Open Motion Planning Library

**OpenCV** Open Source Computer Vision

**ROS** Robot Operating System

**SOTA** Study Of The Art

**URDF** Unified Robot Description Format

# SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>13</b>
1.1 Objetivos . . . . .	13
1.2 Justificativa . . . . .	14
1.3 Organização do relatório . . . . .	14
<b>2 CONCEITO DO SISTEMA</b>	<b>17</b>
2.1 Parâmetros básicos . . . . .	17
2.1.1 Requisitos do cliente . . . . .	17
2.1.2 Requisitos técnicos . . . . .	17
2.1.3 Estudo do estado da arte . . . . .	18
<b>3 DESENVOLVIMENTO DO SISTEMA</b>	<b>21</b>
3.1 Descrição do sistema . . . . .	21
3.1.1 Arquitetura geral . . . . .	21
3.1.2 Especificação técnica . . . . .	22
3.1.3 Ambiente de operação . . . . .	24
3.1.4 Estrutura analítica do protótipo . . . . .	26
3.2 Especificação funcional . . . . .	27
3.2.1 Escaneamento . . . . .	27
3.2.1.1 Descrição . . . . .	27
3.2.1.2 Premissas necessárias . . . . .	28
3.2.1.3 Dependências . . . . .	29
3.2.1.4 Saídas . . . . .	29
3.2.2 Planejamento e Execução de Trajetória . . . . .	29
3.2.2.1 Descrição . . . . .	29
3.2.2.2 Premissas necessárias . . . . .	30
3.2.2.3 Dependências . . . . .	30
3.2.2.4 Saídas . . . . .	31
3.3 Arquitetura de software . . . . .	31
3.4 Simulação do sistema . . . . .	31

<b>4 IMPLEMENTAÇÃO</b>	<b>33</b>
4.1 Parametrização dos motores . . . . .	33
4.2 Estrutura física . . . . .	33
4.2.1 Base . . . . .	34
4.2.2 Elos ou <i>links</i> . . . . .	34
4.2.3 Suportes . . . . .	35
4.2.4 Câmera . . . . .	35
4.2.5 Atuadores . . . . .	37
4.3 Sistema de Potência . . . . .	38
4.4 Comunicação . . . . .	38
4.5 Análise de esforços . . . . .	39
4.6 Configurações . . . . .	40
<b>5 RESULTADOS E ANÁLISES</b>	<b>43</b>
5.1 Caracterização do problema e determinação do modelo . . . . .	43
5.2 Planejamento dos experimentos . . . . .	43
5.3 Resultados alcançados . . . . .	44
5.3.1 Análise de regressão linear . . . . .	44
5.3.2 ANOVA . . . . .	46
5.3.2.1 Análise de variância sem detecção da <i>tag</i> . . . . .	46
5.3.2.2 Análise de variância com detecção da <i>tag</i> . . . . .	48
5.3.3 Análise de repetibilidade e reproduzibilidade . . . . .	51
5.4 Conclusões dos testes estatísticos . . . . .	54
<b>6 CONFIABILIDADE DO SISTEMA</b>	<b>55</b>
6.1 Análise dos modos e efeitos de falhas . . . . .	55
6.2 Análise da árvore de falhas . . . . .	56
<b>7 GESTÃO DO CONHECIMENTO</b>	<b>59</b>
7.1 Lições aprendidas . . . . .	59
7.2 Guia de uso para simulação . . . . .	59
7.3 Guia de uso para o modelo real . . . . .	62
<b>8 CONCLUSÃO</b>	<b>65</b>

<b>REFERÊNCIAS</b>	<b>67</b>
<b>APÊNDICE A Árvores de TF desconectadas</b>	<b>70</b>
<b>APÊNDICE B Árvores de TF conectadas</b>	<b>71</b>
<b>APÊNDICE C Diagrama de nós do sistema</b>	<b>74</b>
<b>APÊNDICE D Propriedades de Massa do JeRoTIMON</b>	<b>75</b>
<b>APÊNDICE E Lista de suportes.</b>	<b>81</b>
<b>APÊNDICE F Projeto mecânico</b>	<b>83</b>
F.1 Desenho mecânico . . . . .	83
F.2 Analise estática . . . . .	86
<b>APÊNDICE G Algoritmo de busca e acionamento do painel</b>	<b>89</b>
<b>APÊNDICE H Diagrama elétrico do JeRoTIMON</b>	<b>101</b>
<b>APÊNDICE I Diagrama de conexão do JeRoTIMON</b>	<b>109</b>
<b>ANEXO A Especificações da câmera Dalsa Genie Nano</b>	<b>115</b>
<b>ANEXO B Especificações do motor <i>Dynamixel PH42-020-S300-R</i></b>	<b>119</b>
<b>ANEXO C Especificações do motor <i>Dynamixel PH54-100-S500-R</i></b>	<b>123</b>
<b>ANEXO D Especificações do motor <i>Dynamixel PH54-200-S500-R</i></b>	<b>127</b>

# 1 INTRODUÇÃO

A robótica é um campo relativamente jovem da tecnologia moderna que atravessa os limites da engenharia tradicional ([SPONG; HUTCHINSON; VIDYASAGAR, 2005](#)). O estudo da robótica é um ramo da tecnologia que engloba área de mecânica, eletrônica e computação, com graus de teoria de controle, microeletrônica, inteligência artificial, fatores humanos e de produção ([PIMENTA, 2009](#)). Segundo ([ERTHAL, 1992](#)), o crescimento da robótica na indústria é justificado em face das exigências de maior qualidade, produtividade e flexibilidade nos processos fabris. Na área industrial, a robótica evoluiu devido ao aumento de uso de robôs e manipuladores industriais.

O estudo do desenvolvimento de manipuladores robóticos foi iniciado por volta de 1954 com George Devol, quando foi desenvolvido o primeiro robô programável e desde então grandes desenvolvimentos nessa área foram atingidos. Como resultado desse avanço, o investimento de empresas foi em torno de 16,5 bilhões de dólares em 2018, chegando a marca de 420 mil unidades enviadas mundialmente, com perspectiva de crescimento médio de 12% ao ano entre 2020 e 2022 ([INDUSTRIAL..., 2019](#)).

Um manipulador robótico é um dispositivo mecânico composto de elementos rígidos (elos) que proporcionam a sustentação e alcance do braço. A inevitabilidade de apresentar algum grau de flexibilidade faz com que esses elos necessitem ser projetados para apresentar elevada rigidez aos esforços que o manipulador será submetido. Esses elos são conectados entre si através de articulações (juntas), que oferecem graus de liberdade ao manipulador e controle de movimento relativo entre os elos. Essas juntas podem ser basicamente divididas em dois grupos: juntas prismáticas e juntas de rotação. Neste projeto foram utilizadas juntas de rotação. A disposição dessas juntas determina a classificação dos manipuladores como Cartesianos, Esféricos, Cilíndricos, entre outros ([ROMANO, 2002](#)).

Este relatório descreve o processo de construção de um manipulador robótico desenvolvido no Centro de Competência em Robótica e Sistemas Autônomos do SENAI CIMATEC e é destinado ao programa de formação Novos Talentos. São descritas as etapas de concepção, simulação, testes e implementação física.

## 1.1 Objetivos

O propósito deste projeto é construir um manipulador robótico capaz de identificar um marcador visual por meio de uma câmera RGB e proceder com a função de acionar um painel elétrico. Para isso, os objetivos específicos são:

- Realizar estudo do Estado da Arte([SOTA](#)) sobre manipuladores.
- Realizar testes e parametrização dos servomotores.

- Propor um modelo de manipulador robótico.
- Parametrizar o pacote de reconhecimento de marcadores visuais.
- Desenvolver um pacote de configuração do *MoveIt*.
- Desenvolver o código para realizar a missão.
- Realizar a simulação do protótipo do manipulador em software.
- Implementar a versão física do protótipo.
- Realizar testes e estudos estatísticos.

## 1.2 Justificativa

Apesar da crescente demanda, há falta de profissionais habilitados à desenvolver pesquisas e aplicações na área da robótica. O presente trabalho tem como impulsionador principal a capacitação de novos pesquisadores preparados para solucionar os mais diversos problemas relacionados a robótica e sistemas autônomos.

A busca por melhor eficiência e precisão na realização de atividades em locais que a presença do ser humano torna-se difícil, arriscado e até mesmo impossível, vem se tornando cada vez maior no cotidiano dos ambiente industriais. Além disso, é importante a capacidade do robô de interagir com o ambiente a partir da captura e análise de estímulos visuais ([LEITE, 2005](#)).

Este projeto traz, dentre os benefícios, a utilização de manipuladores robóticos autônomos que sejam capazes de identificar marcadores visuais e realizar tarefas que possam ser perigosas e/ou repetitivas para o ser humano. Espera-se que este projeto seja continuado e que seus resultados sejam compartilhados na comunidade científica, contribuindo para a construção de outros manipuladores com características e/ou objetivos semelhantes.

## 1.3 Organização do relatório

O presente relatório está organizado em oito capítulos, sendo este de introdução e descrição da justificativa/motivação dos objetivos e da organização do documento.

No capítulo [2](#), Conceito do Sistema, são descritos parâmetros básicos do projeto, dentre eles os requisitos do cliente, requisitos técnicos e o estudo do estado da arte.

O capítulo [3](#), Desenvolvimento do Sistema, apresenta a descrição do sistema onde serão apresentados a arquitetura geral, especificações técnicas, o ambiente de operação do manipulador e a estrutura analítica do protótipo. Além disso, trará as especificações funcionais que compõem o sistema, sua arquitetura de software e o que foi desenvolvido para simulação.

O capítulo 4, Implementação, explica a construção física do manipulador. São expostos seus parâmetros de configuração e sua estrutura.

O capítulo 5, Resultados e Análises, são apresentados os resultados alcançados e a análise dos dados amostrados através de estudos estatísticos.

O capítulo 6, Confiabilidade do Sistema, detalha a análise dos modos e efeitos de falhas.

No capítulo 7, Gestão do Conhecimento, é feito um estudo sobre as lições aprendidas além de apresentar o guia de uso.

Por fim, o capítulo 8 apresenta a conclusão do relatório.



## 2 CONCEITO DO SISTEMA

A norma técnica (ISO-8373, 2012) criada para padronizar o vocabulário referente aos robôs e dispositivos robóticos operando em ambientes industriais e não industriais, define o manipulador como uma máquina na qual o mecanismo, geralmente, consiste em uma série de segmentos, articulados ou deslizantes entre si, com o objetivo de empunhar e/ou mover objetos (peças ou ferramentas) em vários graus de liberdade. Em outras palavras, um manipulador é um equipamento programável baseado em atuadores, com um certo grau de liberdade, projetado para realizar uma variedade de atividades, assim como realização de diversos processos industriais (ISO-8373, 2012).

Neste capítulo serão tratados os requisitos solicitados pelo cliente, os requisitos técnicos do projeto, o estudo do estado da arte sobre manipuladores e o ambiente de operação em que este manipulador realizará a atividade.

### 2.1 Parâmetros básicos

Nesta seção encontram-se os requisitos solicitados pelo cliente, ou seja, a tarefa que precisa ser realizada e em qual ambiente o manipulador será simulado e concebido. Além disso, são exibidos os requisitos técnicos que tratam das especificações do sistema e uma breve revisão teórica de conceitos relacionados ao manipulador.

#### 2.1.1 Requisitos do cliente

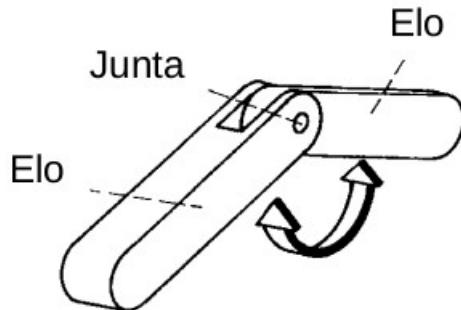
Requisitos predeterminados pelo cliente consistem em exigências de funcionamento que devem ser observadas ao final do projeto, para que se considere um sucesso a concepção deste. Para tal, foram determinadas algumas características desejáveis no projeto:

1. Desenvolver um manipulador robótico.
2. Realizar a tarefa de detecção de um marcador visual e acionamento do painel elétrico na orientação vertical e horizontal.
3. Realizar a simulação do manipulador no ambiente *ROS* utilizando o software *Gazebo*.
4. Utilizar o *framework MoveIt* para o controle do manipulador.
5. Realizar estudo estatístico para verificação da performance do manipulador.

#### 2.1.2 Requisitos técnicos

Os requisitos técnicos de um projeto são especificações necessárias para o funcionamento esperado do projeto. Podem ser sobrepostos aos requisitos do cliente em caso de conflito

Figura 1: Elos e junta de um manipulador robótico.



Fonte: ([SANTOS, 2004](#)).

entre o esperado pelo cliente e o necessário para que o projeto seja bem sucedido, objetivando manter o projeto o mais eficiente dentro do escopo planejado. Os requisitos foram:

1. O manipulador deve estar acoplado em uma base fixa situada a 0,28 m de uma das extremidades da bancada e centralizada com a mesma.
2. Utilizar servo motores Dynamixel PH54-200-S500-R, PH54-100-S500-R e PH42-020-S300-R da Robotis.
3. Conter uma câmera RGB Teledyne Dalsa Genie Nano C2590.
4. Deverá possuir 5 graus de liberdade.
5. Suportar uma carga máxima de 2 kg.
6. Ser capaz de acionar um painel elétrico.

### 2.1.3 Estudo do estado da arte

Com o advento do exponencial crescimento da tecnologia há um foco crescente na pesquisa e comercialização de robôs ([HERNÁNDEZ-ORDOÑEZ et al., 2018](#)). Estes, por sua vez, são classificados em três grupos: manipuladores, veículos auto-guiados ([AGV](#)) e robôs móveis. Neste projeto, o objeto de interesse são os manipuladores robóticos, sistemas que possuem estrutura física similar a um braço humano. Estes robôs são compostos por partes rígidas, denominado de elos, conectados entre si por juntas ([SANTOS, 2004](#)). Esta estrutura encontra-se descrita na Figura 1.

Muitas pesquisas têm sido realizadas na área de manipuladores robóticos. Em ([HERNANDEZ-MENDEZ et al., 2017](#)) é descrito o desenvolvimento de um manipulador robótico com 3 *DoF* e dois dedos independentes. Este robô foi desenvolvido com o propósito de manipular objetos, cujas localizações são conhecidas, e transportá-los de uma localidade para outra

utilizando *ROS*. Os autores utilizaram o *MoveIt* para tratar do planejamento de trajetória e o *Rviz* como ferramenta de visualização. Além disso, foi desenvolvido um controlador de posição e força para o *endeffector*<sup>1</sup> durante o processo de *pick and place*<sup>2</sup>. Os experimentos realizados trouxeram bons resultados para o que foi proposto, sendo ressaltada a necessidade de adicionar um sistema de visão que permita identificar e localizar o objeto alvo.

O planejamento de trajetória para um manipulador de 5 *Dof* a partir do *MoveIt* é exibido em (ZHANG; LIN; WU, 2019). A partir de um modelo *CAD* já existente foi construído o modelo *URDF* utilizado para simulação no *ROS*. O processo de planejamento foi visualizado a partir do *Rviz* e o caminho planejado foi transmitido para os servo-motores possibilitando ao manipulador executar a sua rotina. O robô tem como tarefa manipular um objeto alvo de um local para outro, identificado a partir de técnicas de visão computacional que combinam os algoritmos *SIFT* e *RANSAC*. Os resultados experimentais mostraram que o uso do *MoveIt* reduz as dificuldades de operação para manipuladores e oferece vantagens em termos de validação de algoritmos e exploração de funções, sendo possível utilizar o método proposto para o controle em tempo real do robô.

A aplicação de técnicas de visão computacional em um manipulador do tipo *Robai Cyton Gamma 3000* é exibida em (KHAN; KONDA; RYU, 2018). O robô é conectado com uma câmera externa via *ROS*, possui um *endeffector* em forma de garra que segura uma estrutura similar a um prato, e tem como tarefa equilibrar uma bola localizada no centro do prato. Para realizar o controle da tarefa de equilíbrio, a bola é identificada a partir de um algoritmo escrito em C++ e que utiliza bibliotecas do *OpenCV*. As juntas do manipulador são atuadas a partir de servo-motores *Dynamixel* que são diretamente controlados por um algoritmo. O sistema foi desenvolvido de forma que os componentes se comunicassem entre si para receber respostas do sistema de visão e dos motores, computá-las e enviar comandos de controle que permitissem executar a tarefa. Foram realizados testes e os resultados experimentais foram satisfatórios, sendo o manipulador capaz de equilibrar a bola em uma pequena vizinhança do centro do prato.

O uso de marcadores visuais do tipo *ArUco* associados ao planejamento de trajetória para um manipulador robótico é abordado em (JAVEED; PRAKASH; KULKARNI, 2019). O manipulador encontra-se acoplado a uma plataforma móvel e tem como objetivo o transporte de objetos de forma autônoma em um determinado espaço. Os autores também descrevem a integração dos mecanismos de detecção do marcador visual, navegação do robô e planejamento de trajetória, realizados no *ROS*. O robô é capaz de estimar a pose do marcador, aproximar-se e realizar sua tarefa. Os testes realizados mostraram a

---

<sup>1</sup> Na robótica, um *endeffector* é o dispositivo no final de um braço robótico, projetado para interagir com o meio ambiente.

<sup>2</sup> Sequência de movimentos na qual o manipulador robótico pega determinado objeto e o transfere a uma pose alvo.

eficiência do sistema, sendo apontada a necessidade de um estudo futuro para possibilitar o planejamento de trajetória em um ambiente com obstáculos.

### 3 DESENVOLVIMENTO DO SISTEMA

Nesta seção serão explicitadas as características do manipulador JeRoTIMON, abordando os sistemas que o compõem em *software* e em *hardware*. Suas funcionalidades principais são abordadas e a conexão entre as mesmas é exibida.

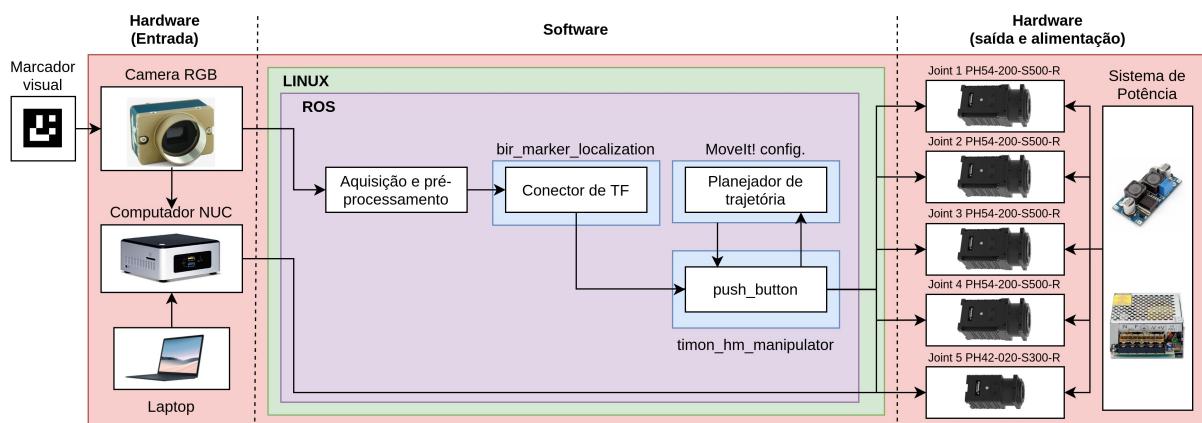
#### 3.1 Descrição do sistema

JeRoTIMON é um manipulador desenvolvido para atender às demandas relacionadas ao reconhecimento de marcadores visuais e, a partir desta identificação, realizar o acionamento de um painel elétrico. Os pacotes que constituem este robô foram concebidos através do *software* de simulação *Gazebo*, da ferramenta de visualização *Rviz* e do *framework*<sup>1</sup> para planejamento de trajetória *MoveIt*. O uso dessas ferramentas possibilita que uma grande variedade de atividades que venham a ser realizadas no mundo real tenham sido previamente testadas em ambiente computacional.

##### 3.1.1 Arquitetura geral

A Figura 2 ilustra a estruturação do sistema e a relação entre *software* e *hardware*. As cores representam o sistema geral(salmão), sistema operacional(verde), *framework*(roxo) e pacotes(azul).

Figura 2: Arquitetura geral do sistema.



Fonte: Autoria própria.

Um laptop, conectado via acesso remoto, dá início a aplicação no computador *NUC*<sup>2</sup> que possui instalado o software do protótipo. Com o sistema iniciado, a câmera RGB é

<sup>1</sup> São conjuntos de aplicações dentro de um projeto que interagem entre si e com isso se alcança resultados como uma determinada função de um programa.

<sup>2</sup> Computador pequeno, completo e altamente eficiente energeticamente.

capaz de obter dados visuais do ambiente e enviá-los para o *ROS*. Ao encontrar o marcador visual, o pacote *bir\_marker\_localization* é capaz de unir as árvores de *TF*<sup>3</sup> do painel elétrico e do manipulador, que antes encontravam-se desconectadas. Esta conexão, garante que sejam conhecidos os dados de posição ao nó *push\_button* possibilitando o planejamento de trajetória para o ponto desejado. Com o caminho planejado, *push\_button* pode enviar os comandos para cada junta do manipulador, onde encontram-se os motores *Dynamixel* que são alimentadas pelo sistema de potência.

### 3.1.2 Especificação técnica

Na Tabela 1 estão elencadas as especificações técnicas do manipulador robótico JeRoTIMON. O numero de Graus de Liberdade foi definido baseado na capacidade de movimentação necessária para a realização dos desafios propostos. A carga útil, peso e o alcance máximo foram calculados com o auxilio do software *Onshape*, uma alternativa ao cálculo manual. A faixa de operação dos motores foi obtida segundo os limites de segurança observados, o que acrescenta proteção principalmente ao cabeamento do sistema. Informações a respeito de componentes como câmeras e motores seguem o seu padrão original de fabricação.

---

<sup>3</sup> Pacote do *ROS* que permite verificar as relações entre os *frames* na estrutura de árvore.

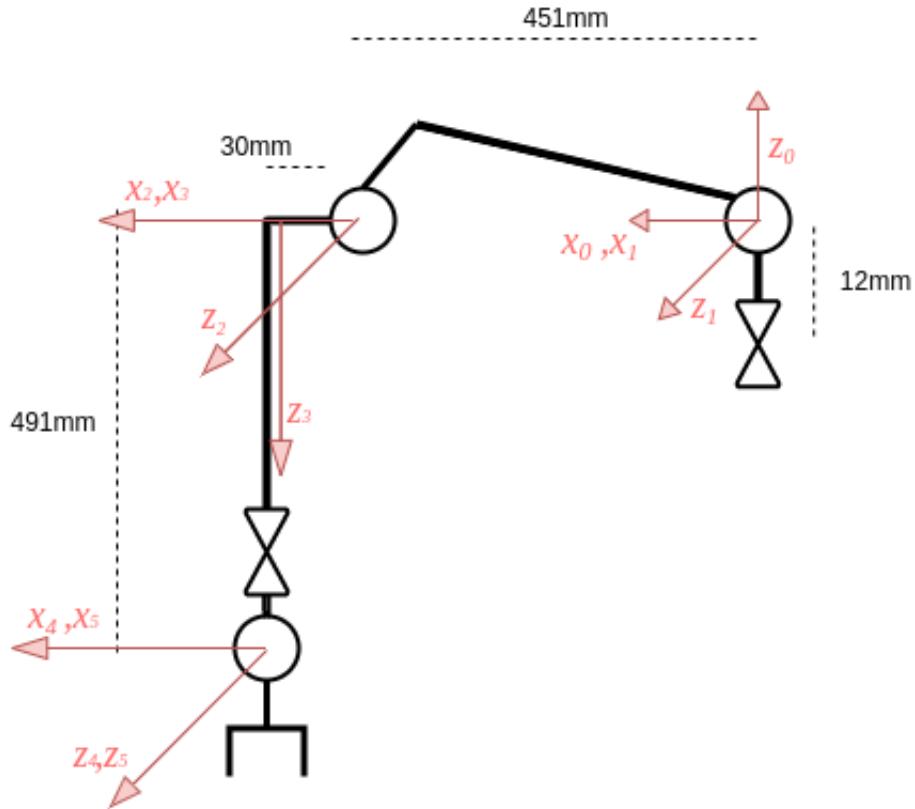
Tabela 1: Especificações técnicas do manipulador JeRoTIMON.

Item	JeRoTIMON
Graus de liberdade	5
Carga útil	2 [kg]
Alcance	981 [mm]
Peso (sem base)	6,4 [kg]
Peso (com base)	10 [kg]
Tensão de operação	24 [V]
Resolução	Junta 1, Junta 2, Junta 3, Junta 4: 1,003,846 [pulsos/rev]
	Junta 5: 607,500 [pulsos/rev]
Motores	Junta 1, Junta 2, Junta 3: PH54-200-S500-R (200 W)
	Junta 4: PH54-100-S500-R (100W)
	Junta 5: PH42-020-S300-R (20 W)
Faixa de operação	Junta 1: $-45^\circ \sim 45^\circ$
	Junta 2: $-90^\circ \sim 90^\circ$
	Junta 3: $-43^\circ \sim 173^\circ$
	Junta 4: $-90^\circ \sim 90^\circ$
	Junta 5: $-90^\circ \sim 90^\circ$
Câmera	Teledyne Genie Nano C2590
Tipo de sensor de posição	Posse inicial: Codificador Absoluto
	Controle: Codificador Incremental
Comunicação	USB
Padrão elétrico	RS485
Taxa de transmissão	57,600 [bps]

Fonte: Autoria própria.

Para estabelecer uma relação entre o *endeffector* e a base do manipulador é necessário descrever o seu sistema de coordenadas em relação ao sistema de coordenadas de origem. Para isto, é utilizada a notação de *Denavit-Hartenger*(D-H). A partir da configuração D-H exibida na Figura 3 foram retirados os parâmetros exibidos na tabela 2.

Figura 3: Configuração D-H do manipulador JeRoTIMON.



Fonte: Autoria própria.

Tabela 2: Parâmetros D-H para o manipulador JeRoTIMON.

Link	a (mm)	$\alpha(^{\circ})$	d (mm)	$\theta(^{\circ})$
1	0	90	12	0
2	452	0	0	$90 - \arctan(30/451)$
3	30	-90	0	$45 + \arctan(30/451)$
4	0	90	491	0
5	0	0	0	0

Fonte: Autoria própria.

### 3.1.3 Ambiente de operação

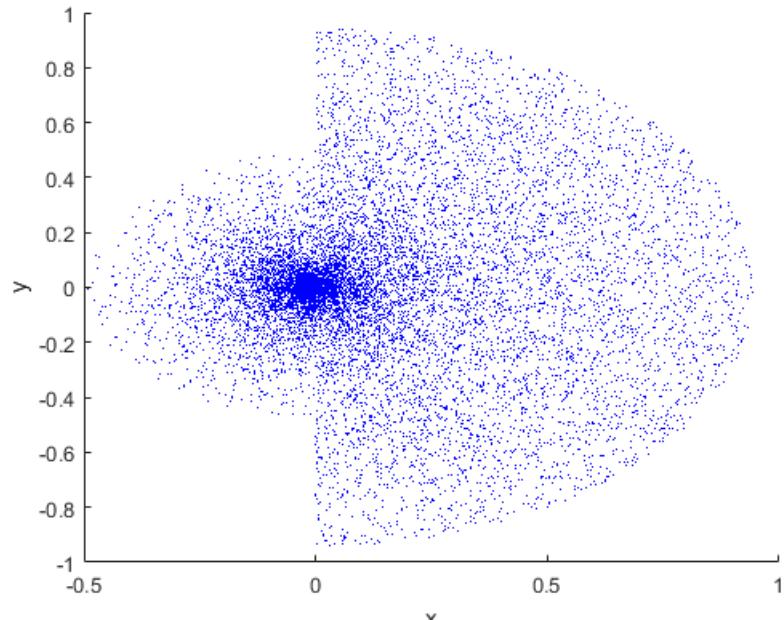
O ambiente físico para a realização do desafio, onde serão incluídos o manipulador e um painel elétrico, é uma mesa com 1.7 m de comprimento por 0.8 m de largura. É necessário então que verifique-se quais os pontos deste ambiente de trabalho que estão dentro da área de alcance (*workspace*)<sup>4</sup> do manipulador.

A partir da tabela 2 foi desenvolvido um código utilizando o software *Matlab R2020a* capaz de gerar pontos que populem o *workspace* do manipulador nas projeções dos planos

<sup>4</sup> É a área de alcance do manipulador, região na qual este consegue operar.

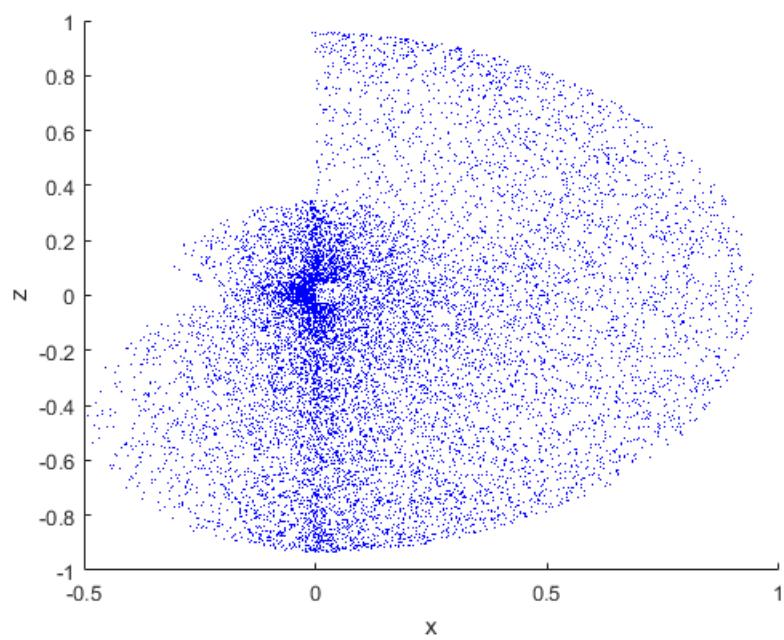
X-Y, X-Z e Y-Z, conforme exibido nas figuras 4, 5 e 6. A região em azul indica o alcance do robô e, a partir destas imagens, é possível observar que o manipulador possui restrições de operação devidas às limitações existentes em cada junta.

Figura 4: *Workspace* do manipulador no plano X-Y.



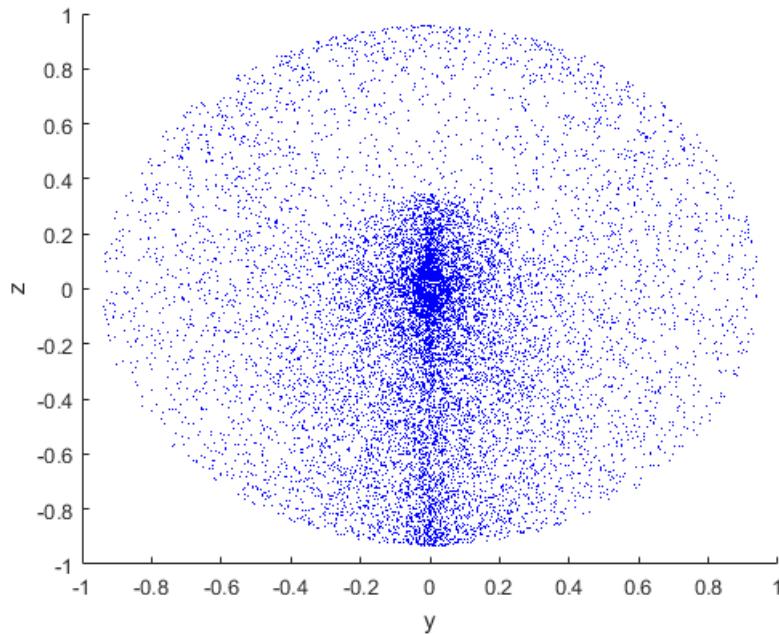
Fonte: Autoria própria.

Figura 5: *Workspace* do manipulador no plano X-Z.



Fonte: Autoria própria.

Figura 6: *Workspace* do manipulador no plano Y-Z.

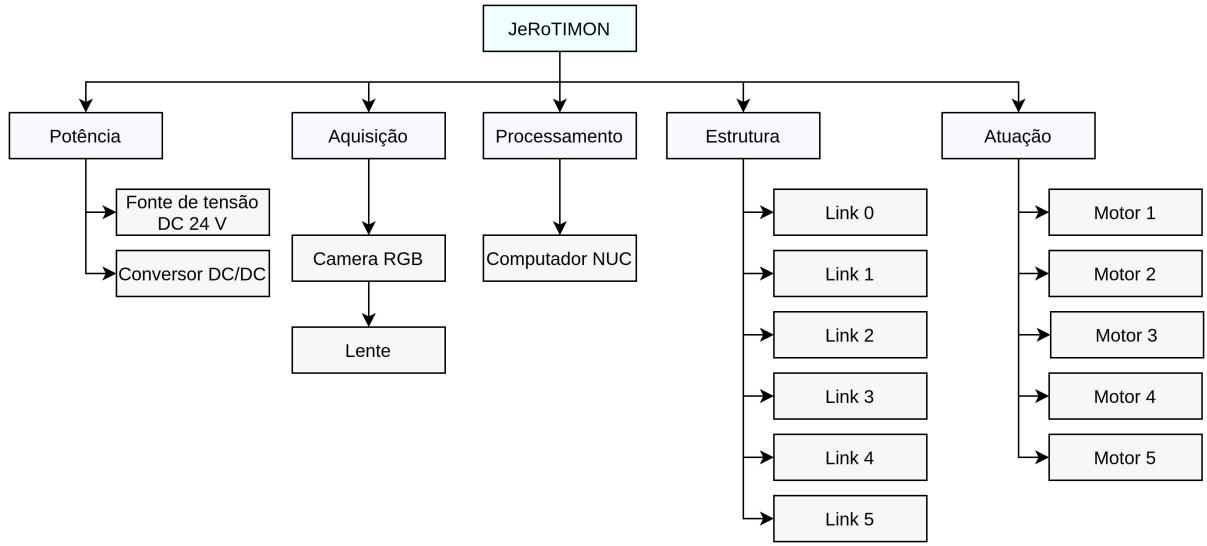


Fonte: Autoria própria.

### 3.1.4 Estrutura analítica do protótipo

A estrutura analítica do protótipo mostrada na Figura 7 exibe as relações sistemáticas entre as partes que compõem o manipulador. A estrutura hierárquica possui três níveis: o primeiro, referente ao sistema principal JeRoTIMON; o segundo, que é composto pelos sub-sistemas de potência, aquisição, processamento, estrutura e atuação; e o terceiro, composto pelos itens que fazem parte de cada um destes sub-sistemas.

Figura 7: Estrutura analítica do protótipo.



Fonte: Autoria própria.

## 3.2 Especificação funcional

O manipulador descrito trabalha acionando os botões encontrados pelo sistema de aquisição. Seu software funciona baseado na troca de mensagens entre duas funcionalidades principais: Escaneamento e Planejamento/Execução de trajetória. O Escaneamento corresponde à detecção de um marcador visual, que uma vez detectado, informa a posição no espaço de um painel elétrico que precisa ser acionado. O planejamento e execução de trajetória utiliza cálculos de cinemática direta e inversa para definir a trajetória de movimentação que permitirá ao manipulador realizar sua tarefa.

### 3.2.1 Escaneamento

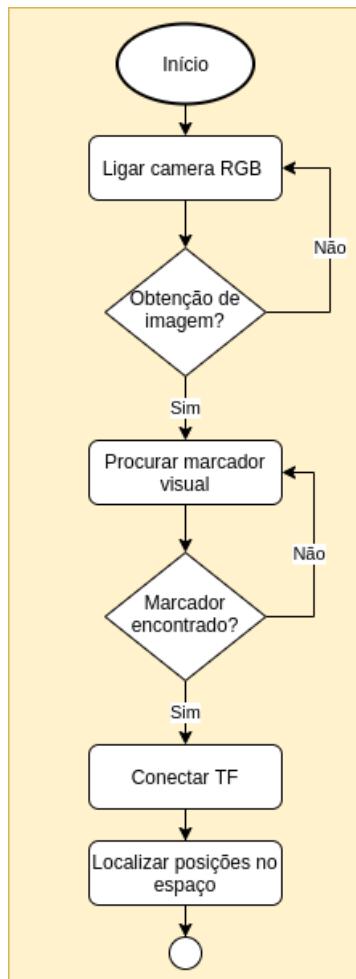
Uma câmera RGB *Teledyne Genie Nano C2590* equipada com lente *kowa LM8FC* foi acoplada ao manipulador JeRoTIMON. Através da mesma é realizada a detecção do marcador visual, utilizando a biblioteca *ArUco* e o pacote *Bir Marker Localization*, hospedado no site do Github no perfil do BIR - Brazilian Institute of Robotics ([BIR... , 2020](#)).

#### 3.2.1.1 Descrição

A Figura 8 exibe o fluxograma que descreve o funcionamento do sistema de escaneamento integrado ao manipulador. Após a captura da imagem, a partir da câmera RGB, é feito um processamento dos dados obtidos afim de localização da *tag ArUco*, cujos tamanho e ID foram previamente estabelecidos. Caso o marcador seja encontrado, as árvores de *TFs* do painel elétrico onde encontra-se o marcador, e do manipulador robótico são conectadas,

possibilitando a localização no espaço da pose alvo. Os apêndices A e B exibem as árvores de *TFs* antes e após a conexão realizada.

Figura 8: Fluxograma do sistema de escanamento.



Fonte: Autoria própria.

### 3.2.1.2 Premissas necessárias

- Conter um marcador visual anexado ao painel elétrico.
- A orientação dos elementos envolvidos no escaneamento, câmera RGB e marcador visual, devem ser definidos conforme estabelecido pelo pacote *bir\_marker\_localization*.
- Não haver oclusão do marcador visual.
- Nível de iluminação do ambiente suficiente para a execução da detecção.
- O marcador visual deverá possuir tamanho adequado para detecção.

### 3.2.1.3 Dependências

Para realizar a etapa de detecção é necessária a instalação do *OpenCV* versão 3.3.1, *driver GigE-V Framework* e a inserção dos pacotes *bir\_marker\_localization* e *def\_cam\_tedelyne\_nano* no *workspace* do manipulador.

### 3.2.1.4 Saídas

É fornecida ao sistema resposta por meio de uma sequência de mensagens publicadas no tópico */timon/camera/image\_raw*. Estes dados são analisados pelo detector *ArUco*, possibilitando seu uso em um pacote desenvolvido na linguagem C++ que determina a posição do painel elétrico associado ao marcador visual.

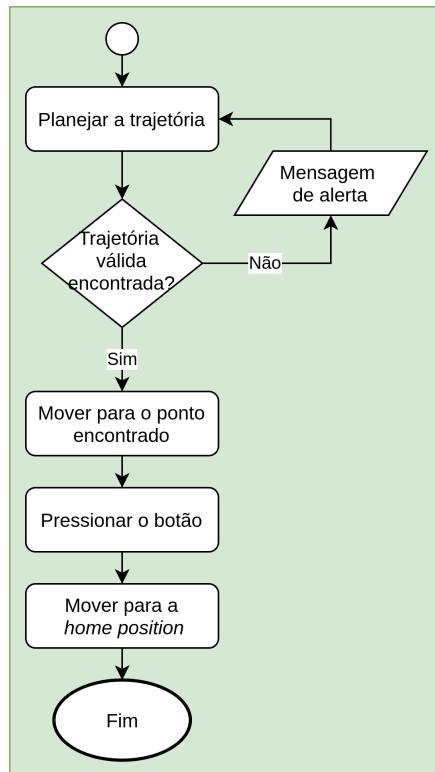
## 3.2.2 Planejamento e Execução de Trajetória

As equações cinemáticas são a base que possibilitam a pesquisa do movimento dos manipuladores. A cinemática inversa provê um conjunto de valores para as juntas do manipulador com o intuito de alcançar uma determinada pose pré-estabelecida do seu *endeffector*. Para resolver as equações da cinemática inversa do JeRoTIMON, optou-se por utilizar o plugin TRAC-IK, um método alternativo ao padrão da inversa Jacobiana utilizado pelo *MoveIt*. Este método se adequa bem a manipuladores que possuem limitações em suas juntas, ao contrário de algoritmos baseados no teorema de Newton ([BEESON; AMES, 2015](#)). Para o planejamento de trajetória foi utilizada a biblioteca *OMPL*, uma coleção de algoritmos de planejamento utilizada por padrão no *MoveIt* ([SUCAN; MOLL; KAVRAKI, 2012](#)).

### 3.2.2.1 Descrição

A Figura 9 exibe o funcionamento do sistema de planejamento e execução de trajetória aplicado ao manipulador. A pose do painel elétrico, determinada a partir do que foi mostrado em 3.2.1, é enviada como entrada para o *MoveIt*. Caso seja possível, é realizado o planejamento de uma trajetória para cada uma das juntas do manipulador, a fim de movê-lo para a pose desejada. Esta trajetória é então enviada para os atuadores das juntas, que passam a executá-la. Após a realização da rotina para o pressionamento do painel elétrico, o manipulador retorna para sua posição inicial. Caso alguma condição impeça o planejamento de trajetória, como por exemplo o posicionamento do painel elétrico fora da área de trabalho do manipulador ou falhas nas soluções para as equações da cinemática inversa, uma mensagem de alerta é exibida e o robô realiza uma nova tentativa de planejamento.

Figura 9: Fluxograma do sistema de planejamento e execução de trajetória.



Fonte: Autoria própria.

### 3.2.2.2 Premissas necessárias

- Viabilidade das soluções para cinemática inversa.
- Painel elétrico estar posicionado na área de trabalho do manipulador.

### 3.2.2.3 Dependências

O sistema de movimentação é dependente da versão Melodic Morenia do *framework ROS* e da plataforma *MoveIt*. Além destes, uma lista de pacotes deve ser instalada previamente para o funcionamento correto do sistema:

- ros-melodic-ros-control
- ros-melodic-gazebo-ros-control
- ros-melodic-controller-manager
- ros-melodic-joint-trajectory-controller
- ros-melodic-joint-state-controller
- ros-melodic-position-controllers

- ros-melodic-trac-ik-kinematics-plugin

### 3.2.2.4 Saídas

As respostas fornecidas pelo sistema de planejamento e execução são publicadas nos motores *Dynamixel* integrados às juntas do manipulador. A trajetória gerada pelo *MoveIt* pôde ser visualizada a partir do tópico */move\_group/display\_planned\_path* e é publicada nos motores a partir do tópico */timon\_arm\_controller/dynamixel\_state*.

## 3.3 Arquitetura de software

O robô JeRoTIMON foi desenvolvido para atuar em conjunto com o *ROS*, isto é, segue o propósito de conectar diferentes módulos, como câmeras, motores, sensores e códigos. A proposta é conectar um programa de visão capaz de conectar árvores de TF, através da identificação de marcadores visuais, com um sistema de movimentação que posiciona o manipulador para acionar o painel elétrico.

O apêndice C exibe o diagrama do *software* do sistema, obtido via *rqt\_graph*. Observa-se a comunicação entre o manipulador (*/timon/robot\_state\_publisher*) e o painel elétrico (*/box/box\_state\_publisher*) a partir da conexão das árvores de *TFs* realizadas pelo */marker\_localization* que recebe e analisa os dados da câmera (*/timon/camera\_image\_raw*). Pode-se verificar também que o nó */arm* comunica-se com o nó */move\_group* e com o nó */timon\_arm\_controller* a fim de enviar a trajetória planejada pelo *MoveIt* para os motores *dynamixel*.

## 3.4 Simulação do sistema

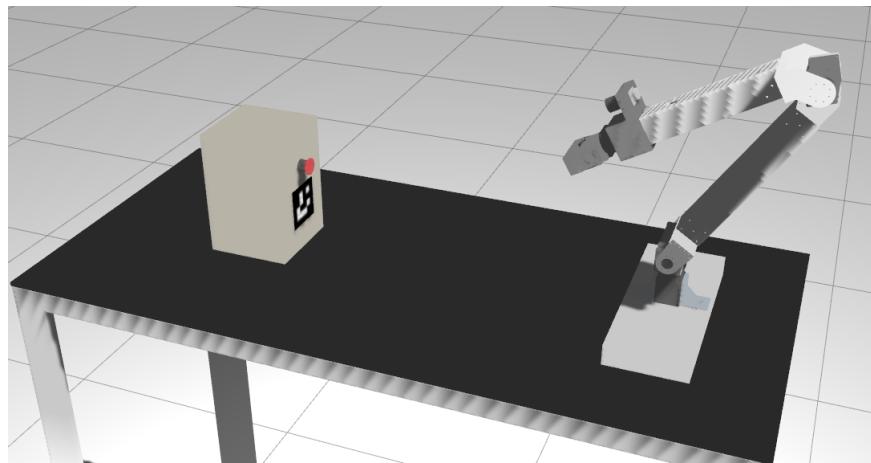
A simulação do robô JeRoTIMON inserido em seu ambiente de trabalho foi realizada na plataforma *Gazebo*. Para isto, realizou-se o desenvolvimento dos arquivos *timon\_arm.urdf* e *box.urdf*, modelos *URDF* que descrevem o robô e o painel elétrico a ser acionado. Foram levados em consideração características de massa, inércia e dimensão para cada componente destes modelos, para que houvesse maior fidelidade possível com o que representam fisicamente, de forma a garantir que os testes realizados possam ser validados em ambiente real.

O pacote *ros\_control* dispõe de uma lista de controladores disponíveis. Para JeRoTIMON, foram utilizados controladores do tipo *position\_controllers/JointTrajectoryController*, e as transmissões para cada junta do manipulador foram definidas em seu modelo *URDF*.

Para simulação dâ câmera RGB, foi desenvolvido o arquivo *camera.xacro*. O plugin padrão do *Gazebo* foi utilizado, *ligazebo\_ros\_camera.so* e as especificações da câmera RGB *Teledyne Genie Nano C2590* foram levados em consideração, garantindo maior realismo à simulação. A figura 10 exibe os modelos simulados do manipulador e do painel elétrico,

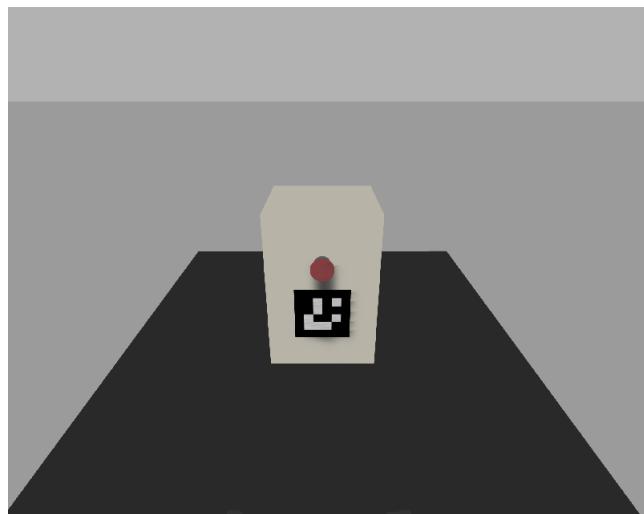
enquanto a figura 11 exibe imagem capturada pela câmera instalada no manipulador.

Figura 10: Modelos simulados do manipulador e do painel elétrico.



Fonte: Autoria própria.

Figura 11: Imagem capturada pela câmera RGB.



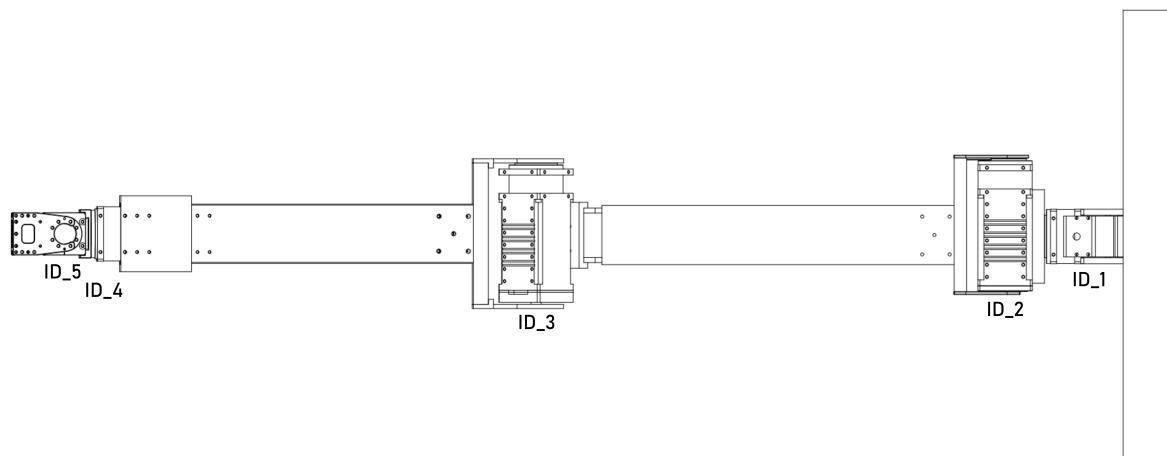
Fonte: Autoria própria.

## 4 IMPLEMENTAÇÃO

### 4.1 Parametrização dos motores

Na Tabela 3 encontram-se as configurações para os motores, que estão identificados na Figura 12. Nesta Tabela estão especificados os ângulos e as posições mínimas e máximas definidas para cada motor. Estes parâmetros foram escolhidos após testes e verificações de acordo com a capacidade do manipulador interagir com o ambiente de trabalho, pensando nas possíveis localizações que a caixa poderá estar situada.

Figura 12: Identificação dos motores.



Fonte: Autoria própria.

Tabela 3: Parâmetros dos motores.

Motor	Ângulo (min)	Posição (min)	Ângulo (máx)	Posição (máx)
ID_1	-45°	-125000	45°	125000
ID_2	-90°	-250962	90°	250962
ID_3	-43°	-119095	173°	483855
ID_4	-170°	-475464	170°	475464
ID_5	-90°	-151875	90°	151875

Fonte: Autoria própria.

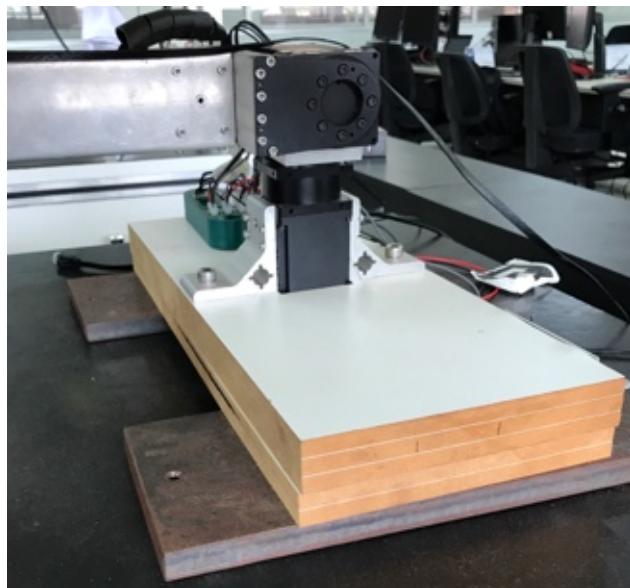
### 4.2 Estrutura física

Após montagem física, o manipulador obteve um alcance de aproximadamente 980 mm. Levou-se em consideração o mínimo alcance necessário para que a missão possa ser realizada conforme as especificações do cliente. Então foram projetadas cinco juntas rotacionais como mostra a Figura 12.

#### 4.2.1 Base

Foi decidido utilizar uma base de madeira com dimensões de aproximadamente  $450 \times 180$  mm e espessura de 50 mm (Figura 13). Esta base será utilizada para fixar o manipulador, garantindo estabilidade durante a execução da tarefa.

Figura 13: Base do manipulador.



Fonte: Autoria própria.

#### 4.2.2 Elos ou *links*

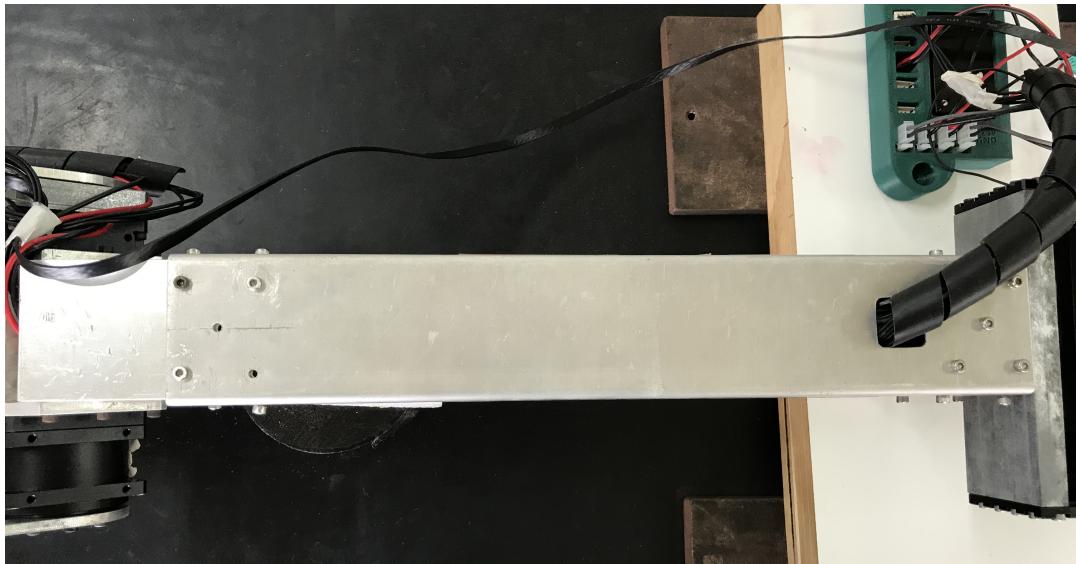
Para estruturação dos elos do robô são utilizados dois perfis de alumínio vazados com comprimento, largura e altura de  $350 \times 58.7 \times 58.7$  mm, respectivamente (Figuras 14 e 15). Estas dimensões foram escolhidas levando em consideração a disponibilidade dos materiais, capacidade de alcance do braço para realizar a tarefa e capacidade estrutural do manipulador, para assim suportar esforços de natureza estática e dinâmica.

Figura 14: Vista lateral do *link* 2.



Fonte: Autoria própria

Figura 15: Vista superior do *link 2*.



Fonte: Autoria própria

#### 4.2.3 Suportes

Foram utilizados suportes com finalidade de unir os elos do manipulador e distribuir movimentos rotativos para as juntas seguintes, feitos em alumínio. Alguns suportes foram adquiridos através da ROBOTIS, enquanto outros foram confeccionados pelo laboratório CCRoSA. A Tabela com os suportes utilizados e suas imagens e medidas pode ser vista no apêndice E.

#### 4.2.4 Câmera

Para prover o sistema com capacidade de detecção da *tag* e assim obter dados necessários para aquisição de pose e orientação relativa, utilizou-se uma câmera de vídeo modelo Teledyne Genie Nano C2590 (Figura 16) e foi acoplada a lente 16mm C Series VIS-NIR (Figura 17). A comunicação entre a câmera e o sistema é feita através de um cabo categoria 6 RJ45, e sua ligação pode ser vista no apêndice I.

Figura 16: Teledyne Genie Nano C2590.



Fonte: ([TELEDYNE...](#), s.d.)

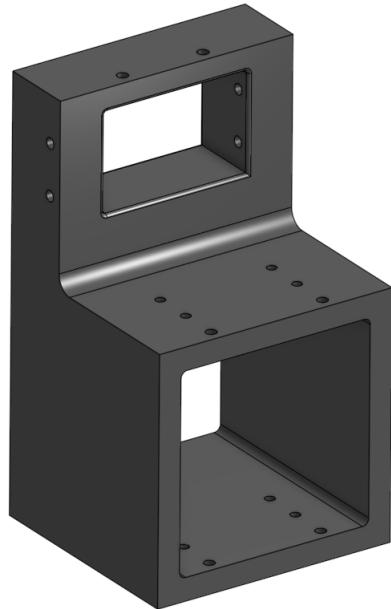
Figura 17: Lente 16mm C Series VIS-NIR.



Fonte: ([16MM...](#), s.d.)

Este conjunto pode ser fixado próxima a extremidade do manipulador robótico utilizando um suporte impresso em ABS (Figura 18), facilitando a detecção da *tag* e não comprometendo a estrutura do manipulador. O suporte é conectado ao final do *link 3*, como mostra o apêndice D. A integração pode ser visualizada na Figura 19.

Figura 18: Suporte para fixação da câmera.



Fonte: Autoria própria

Figura 19: Integração suporte-câmera-lente.



Fonte: Autoria própria

#### 4.2.5 Atuadores

Os atuadores do manipulador são motores de corrente contínua, integrados com redutor de velocidade, controlador e driver. Foram utilizados atuadores *Dynamixel* da fabricante

ROBOTIS. Entre os modelos figuram o PH54-200-S500-R, presente nas juntas 0, 1 e 2, o motor PH54-100-S500-R para a junta 3 e o motor PH42-020-S300-R foi utilizado para a junta 4. As especificações do fabricante mais relevantes no projeto estão apresentadas nas tabelas 4 e 5. As folhas de dados estão disponíveis nos anexos B, C e D para os atuadores PH42-020-S300-R, PH54-100-S500-R, PH54-200-S500-R, respectivamente.

Tabela 4: Parâmetros dos motores.

Tipo	Torque (N.m)	Tensão (V)	Corrente (A)	Velocidade de Rotação (rpm)	Dimensões (mm)
PH54-200-S500-R	44.7	24.0	9.3	29.0	54.0 X 126.0 X 54.0
PH54-100-S500-R	25.3	24.0	5.5	29.2	54.0 X 108.0 X 54.0
PH42-020-S300-R	5.1	24.0	1.5	29.2	42.0 X 84.0 X 42.0

Fonte: ([ROBOTIS, s.d.](#))

### 4.3 Sistema de Potência

O sistema necessitará ser energizado em dois níveis de tensão, 12 V e 24 V. Durante a realização dos testes, utilizou-se uma fonte com níveis de tensão de 0-30 V e fornecimento de corrente de 0-10 A, para atender os requisitos necessários de potência dos motores. Foi utilizado um conversor DC-DC modelo UWE-12/10-Q12PB-C (Figura 20) na saída da fonte para atingir o nível de tensão de 12 V e energizar a câmera. Um esquema elétrico é fornecido no apêndice H, indicando as conexões dos cabos entre os motores e a câmera.

Figura 20: UWE-12/10-Q12PB-C.



Fonte: ([UWE-12/10-Q12PB-C, s.d.](#))

### 4.4 Comunicação

O manipulador é conectado via USB através de um dispositivo denominado U2D2 ([U2D2, s.d.](#)). Ele consegue parametrizar e enviar comandos para os motores através dos softwares originais da ROBOTIS chamados Dynamixel Workbench e Dynamixel Wizard. O padrão elétrico utilizado é RS-485, ou seja, utiliza-se de 4 fios de conexão que transmitem níveis de tensão e dados. A conexão de entre o computador e os motores, utilizando o U2D2 de

intermédio, pode ser vista no apêndice I. A velocidade de transmissão (*baud rate*) definida para todos os motores no sistema é de 57600 bps.

Figura 21: U2D2.



Fonte: ([U2D2, s.d.](#))

## 4.5 Análise de esforços

A análise de esforços mecânicos aos quais o manipulador está exposto foi realizada para embasamento e confirmação da capacidade da estrutura e dos atuadores suportarem os esforços de momento torsor. Para isso foram analisadas individualmente as juntas em seus estados críticos. Na Tabela 5 é apresentada a análise de esforços nas juntas e comparados com os máximos esforços suportados pelos atuadores (conforme vistos nos apêndices B, C e D), para evitar falhas mecânicas.

Levando em consideração os valores obtidos nas análises de esforços, tem-se as juntas 0 e 1 como juntas críticas do sistema, pois elas sofrem a maior força do sistema e tem o maior torque exigido. Na configuração apresentada, as juntas 0 e 1 suportam a carga máxima de 45.22 N sem chegar a falha mecânica, logo o manipulador possui um limite (*payload*) de aproximadamente 2.1 kg na sua extremidade.

Tabela 5: Torque das juntas.

Junta	Torque máximo fornecido pelos motores (Nm)	Força resultante na junta (N)	Torque exigido pela junta (Nm)
0	44.7	45.22	25.8
1	44.7	45.22	25.8
2	44.7	21.44	6.9
3	25.3	4.8	0.12
4	5.1	0.87	0.05

Fonte: Autoria própria.

## 4.6 Configurações

O conjunto formado pelo manipulador, seu espaço de trabalho e os objetos com os quais ele deverá interagir compõem o sistema abordado neste trabalho. O espaço de trabalho foi representado como uma bancada de  $1.70\text{ m} \times 0.80\text{ m} \times 0.028\text{ m}$ . Há apenas um objeto com o qual o manipulador deverá interagir, uma caixa de dimensões  $0.30\text{ m} \times 0.20\text{ m} \times 0.20\text{ m}$ .

Para o desafio foi utilizado um marcador visual *ArUco* com dimensões  $58\times 58\text{ mm}$  de id 4, conforme mostrado na Figura 22. Este tem 8 cm do centro do *ArUco* para o botão e 7 cm do centro do *ArUco* para a lâmpada, esta identifica se o botão foi acionado ou não pelo manipulador.

Figura 22: Caixa objetivo.



Fonte: Autoria própria

O manipulador possui a posição home definida como da Figura 23 onde o ângulo de cada motor está descrito na Tabela 6.

Figura 23: Manipulador na posição home.



Fonte: Autoria própria.

Tabela 6: Ângulos dos motores na posição home.

<b>Motor</b>	<b>Ângulo</b>
ID_1	0°
ID_2	-24°
ID_3	141°
ID_4	0°
ID_5	0°

Fonte: Autoria própria.



## 5 RESULTADOS E ANÁLISES

Os experimentos apresentados nesta seção possuem como objetivo avaliar o desempenho do manipulador robótico desenvolvido.

### 5.1 Caracterização do problema e determinação do modelo

Para análise da performance do manipulador, deseja-se avaliar inicialmente a interação de algumas variáveis que podem influenciar no desempenho do manipulador. Em um segundo momento, após estabelecidas as melhores configurações de algoritmo e velocidade para o robô, os testes devem seguir uma análise de repetibilidade e de variância do sistema.

No planejamento dos experimentos, foram levantadas quais variáveis seriam de interesse para avaliar a performance do robô, com isso foram selecionadas como variáveis de saída a precisão do manipulador, ou seja, a capacidade do mesmo de chegar a uma posição determinada com o menor erro possível, tempo de busca do marcador visual, e o tempo necessário para o acionamento do botão (missão imposta ao manipulador).

Também foram selecionadas as variáveis independentes, ou seja, as variáveis de entrada do processo, que serão analisadas quanto às suas influências no sistema de forma isolada ou a interação entre as mesmas. As variáveis selecionadas para o estudo foram: velocidade de operação dos motores, algoritmo de planejamento de trajetória utilizado e a posição da caixa no ambiente de trabalho do manipulador.

### 5.2 Planejamento dos experimentos

Definido o problema e as variáveis a serem estudadas, foi realizado um planejamento dos experimentos, para assim gerar dados e resultados confiáveis acerca da melhor configuração para o manipulador. As tabelas 7, 8 e 9 apresentam os modelos dos experimentos.

Tabela 7: Modelo de experimentos para análise de 3 variáveis.

Variáveis Independentes	Níveis		Variáveis dependentes
Algoritmo de planejamento de trajetória	RRT-CONNECT	Kpiece	Erro de posição
Velocidade dos motores	300	500	Tempo total
Posição da caixa	A	B	

Fonte: Autoria própria.

Tabela 8: Modelo de experimentos ANOVA sem detecção da *tag*.

Ponto	Número de repetições	Variáveis dependentes
A	10	Erro de posição
B	10	Erro de posição

Fonte: Autoria própria.

Tabela 9: Modelo de experimentos ANOVA com detecção da *tag*

Operador	Pose da caixa	Número de regravações	Sucesso	Variáveis dependentes
1	A	10	-1,1	Erro de posição
2	B	10	-1,1	Tempo de busca
1	B	10	-1,1	Tempo total
2	A	10	-1,1	

Fonte: Autoria própria.

## 5.3 Resultados alcançados

A partir dos testes realizados sob orientação do planejamento, foram coletados os resultados do tempo de busca do marcador visual, tempo da missão do manipulador e o erro de posição em cada experimento. As amostras dos testes coletadas foram analisados da seguinte forma: Análise de regressão linear, análise de variância e teste R&R, para os testes com e sem a detecção do marcador visual .

### 5.3.1 Análise de regressão linear

A partir do planejamento de experimentos realizados, foram tomados três dados de cada condição estabelecida e utilizada a média entre os três resultados como valor do experimento. As amostras coletadas estão representadas na tabela 10. O algoritmo de planejamento RRT CONNECT é representado pelo sinal de subtração (-) enquanto o sinal de adição (+) representa o algoritmo Kpiece. Já para as velocidades, (+) representa o valor de 500 rpm e (-) representa 300 rpm.

Tabela 10: Resultados das amostras coletadas.

Ordem	Algoritmo	Velocidade	Erro	Tempo
1	-	-	0.0004759353	175.25667
2	+	-	0.0009316073	82.24667
3	-	+	0.0002889380	139.39333
4	+	+	0.0008311057	90.34333

Fonte: Autoria própria.

O resultado obtido a partir da análise linear aplicada aos dados é disposto na tabela 11. O primeiro modelo dessa análise pode ser visualizado na Equação 5.1. Esta equação

mostra um modelo que consegue explicar o valor do erro de posição a partir da relação entre os parâmetros.

Tabela 11: Resultados da análise de regressão linear.

	<b>Estimate</b>	<b>Std.</b>	<b>Error</b>	<b>t value</b>	<b>Pr(&gt; t )</b>
(Intercept)	4.543e-04	3.745e-05	12.130	0.0524	
algoritmo+	4.989e-04	4.325e-05	11.536	0.0550	
velocidade+	-1.437e-04	4.325e-05	-3.324	0.1860	

Fonte: Autoria própria.

$$\text{erro} = 4.543.(10^{-04}) + 4.989.(10^{-04}).\text{algoritmo}(+) - 1.437.(10^{-04}).\text{velocidade}(+) \quad (5.1)$$

Observando os dados apresentados na tabela 11, o  $\rho$ -valor da variável velocidade é superior a 0,05, aceitando a hipótese nula, ou seja, a variável velocidade possui pouca ou nenhuma influência na tomada de dados do sistema. Porém, percebe-se que o  $\rho$ -valor da variável algoritmo é próxima de 0.05, implicando que a mesma pode ser analisada de forma individual, rejeitando-se a hipótese nula para esta variável. Desta forma refazendo-se a análise linear do erro de posição com relação apenas a variável algoritmo obtém-se o novo modelo representado na equação 5.2. O resultado dos coeficientes pode ser visto na tabela 12.

Tabela 12: Resultados da análise de regressão linear (variável algoritmo).

	<b>Estimate</b>	<b>Std.</b>	<b>Error</b>	<b>t value</b>	<b>Pr(&gt; t )</b>
(Intercept)	3.824e-04	7.506e-05	5.095	0.0364	
algoritmo+	4.989e-04	1.062e-04	4.700	0.0424	

Fonte: Autoria própria.

$$\text{erro} = 3.824.(10^{-04}) + 4.989.(10^{-04}).\text{algoritmo}(+) \quad (5.2)$$

Como apresentado na tabela 12 o algoritmo apresenta  $\rho$ -valor menor que 0.05, ou seja, rejeita-se a hipótese nula para esta variável, mostrando que a mesma tem influência no erro de posição do manipulador.

Após esta análise serão realizados novos testes levando em consideração as variáveis examinadas neste estudo, sendo configurado o sistema com o algoritmo RRT CONNECT, uma vez que o algoritmo Kpiece teve influência positiva, ou seja, aumentou o erro de posição do manipulador, o que não é desejável para este sistema. A velocidade dos motores não apresentou influência considerável nos testes, portanto foi configurada com a menor velocidade por motivos de segurança.

### 5.3.2 ANOVA

Com o algoritmo de planejamento e velocidade estabelecidas para a continuidade dos testes, foram realizados dois tipos de experimentos: análise de erro de posição do *end-effector* sem a detecção do marcador visual, ou seja, o manipulador recebe comandos de posição em x, y e z dentro da sua área de trabalho e então é observado o erro ao executar o comando de posição. O segundo teste foi semelhante, porém, este contou com a detecção da *tag*, recebendo assim o comando de posição a partir da detecção do marcador visual. Com isso foram tomados dados em duas diferentes poses para assim analisar a variância do erro em cada pose e entre elas.

#### 5.3.2.1 Análise de variância sem detecção da *tag*

Para esta análise foram estabelecidas, na área de trabalho do manipulador, duas posições, onde Pose A equilava a  $x = 0.314445$   $y = 0.730867$   $z = 0.233471$  e Pose B equilava a  $x = -0.153158$   $y = 0.755479$   $z = 0.238906$ , ambas referentes ao *endeffecter* do manipulador, e então foram realizados 10 execuções de movimento para cada pose estabelecida. A tabela 13 mostra os erros de posição do *endeffecter* para cada uma das execuções.

Tabela 13: Resultados dos experimentos sem detecção da *tag*.

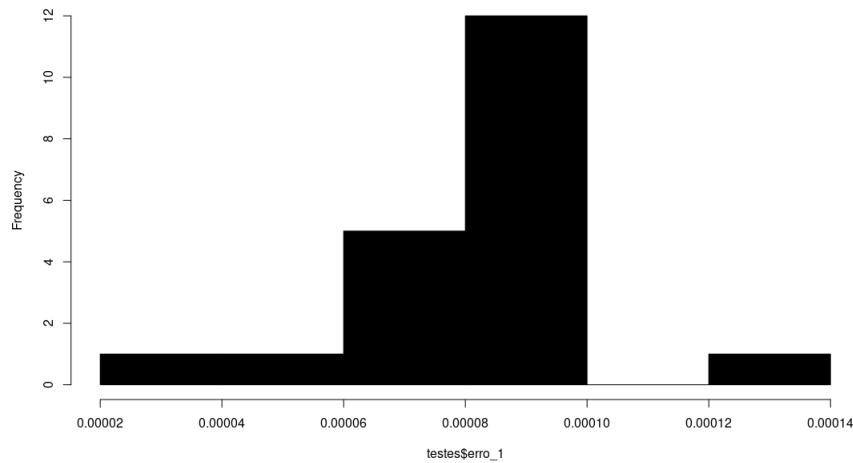
	Pose	Erro
1	A	0.000135296
2	A	0.000095000
3	A	0.000090747
4	A	0.000097944
5	A	0.000081093
6	A	0.000068184
7	A	0.000088978
8	A	0.000074532
9	A	0.000036373
10	A	0.000086499
11	B	0.000095900
12	B	0.000085300
13	B	0.000064900
14	B	0.000089500
15	B	0.000089000
16	B	0.000048300
17	B	0.000096700
18	B	0.000090200
19	B	0.000077800
20	B	0.000079800

Fonte: Autoria própria.

Para avaliar os dados, inicialmente foi realizado um teste de normalidade, como forma

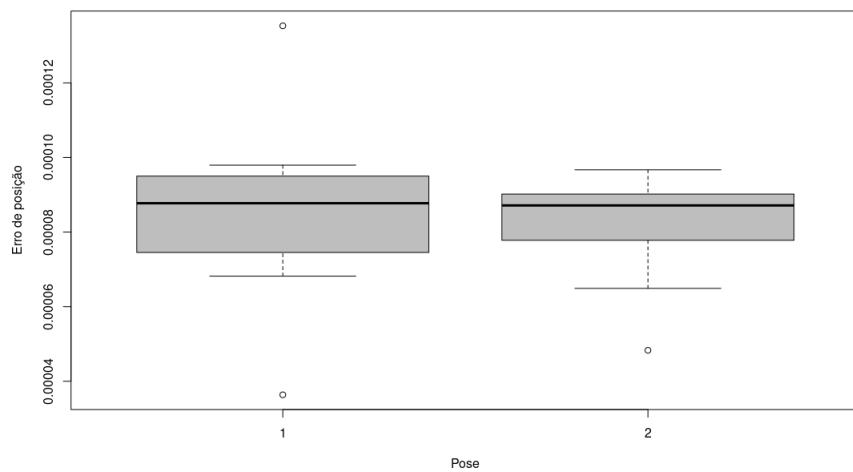
de verificar se os dados são bem modelados por uma distribuição normal ou não. O teste de Shapiro-Wilk foi utilizado para esse conjunto de dados, retornando um  $\rho$ -valor de 0.05569, o que indica que a hipótese nula não pode ser rejeitada, logo os dados seguem uma distribuição normal. A distribuição dos mesmos pode ser melhor visualizada através de um histograma, como mostra a figura 24.

Figura 24: Histograma do conjunto de dados sem detecção da *tag*.



Segundo a análise dos testes, foi realizado um teste ANOVA com a amostra de dados coletada. O objetivo foi avaliar a variância dos dados em cada ponto e entre eles, analisando se estes diferem estatisticamente entre si. Com esta análise, obteve-se um  $\rho$ -valor de 0.6912, com isso tem-se embasamento estatístico para afirmar que os dados no ponto "A" e no ponto "B" são estatisticamente iguais, os erros de pose apresentados nos dois pontos não diferem entre si. Esta análise pode ser visualizada na figura 25.

Figura 25: Boxplot do conjunto de dados sem detecção da *tag*.



Observando a figura 25 a pose 1 (A) possui maior variação dos seus dados em relação a pose 2 (B), porém as duas não diferem estatisticamente entre si.

### 5.3.2.2 Análise de variância com detecção da *tag*

Foi tomado um novo conjunto de dados, como mostra a tabela 14. Este experimento foi realizado com variação do operador e posição da caixa. Foram tomados dados de tempo de busca da *tag*, tempo total da missão e o erro de posição do *end-effector*. Também foi observado a porcentagem de sucesso na conclusão da missão, sendo o índice '1' correspondente ao sucesso na missão e o índice '-1' correspondente à falha.

Tabela 14: Resultados dos experimentos sem detecção da *tag*.

	Operador	Pose	Erro de pose	Tempo de busca	Tempo total	Sucesso da missão
1	1	A	0.010451819	27.1022	93.9316	1
2	1	A	0.009745817	29.1912	91.8192	1
3	1	A	0.010017589	27.3904	95.6437	1
4	1	A	0.010819201	28.4745	90.5214	1
5	1	A	0.011133634	27.7250	98.7845	1
6	1	A	0.010562436	28.8388	94.6226	1
7	1	A	0.010132297	29.1519	101.8420	1
8	1	A	0.010693537	30.2186	90.1885	1
9	2	A	0.010905047	30.2317	94.0104	1
10	2	A	0.010469991	28.9793	96.2312	1
11	2	A	0.010355747	29.0720	103.6160	1
12	2	A	0.010268870	28.6952	93.0651	1
13	2	A	0.010230915	28.6420	87.2120	1
14	2	A	0.004079350	26.8585	86.4156	1
15	2	A	0.004427968	26.1808	85.7109	1
16	2	A	0.009985857	30.0858	90.4296	1
17	1	A	0.009754491	28.2042	86.8758	1
18	1	A	0.010334966	28.6983	92.2869	1
19	1	A	0.010058060	29.4320	94.3545	1
20	1	A	0.010153228	28.1812	90.6696	1
21	1	A	0.010120659	29.2442	87.1852	1
22	1	A	0.010105795	28.9948	88.3569	1
23	1	A	0.010105795	28.9948	88.3569	-1
24	1	A	0.011557242	32.9496	102.3640	1
25	2	A	0.011245238	34.7259	98.7636	1
26	2	A	0.011190817	31.1780	113.0500	1
27	2	A	0.010174117	28.7923	88.9735	1
28	2	A	0.010007278	28.8178	91.8702	1
29	2	A	0.009963924	29.0347	86.7183	1
30	2	A	0.010475097	28.2422	99.1881	1
31	2	A	0.010741058	28.2118	91.7459	1
32	2	A	0.004469904	28.4826	88.1583	-1
33	1	A	0.010086673	28.7112	87.9372	1
34	1	A	0.010515376	28.8792	92.9002	1
35	1	A	0.010062717	29.1420	91.5910	1
36	1	A	0.010422717	29.0783	94.3296	1
37	1	A	0.010185682	27.9274	88.6971	1
38	1	A	0.010281946	29.7805	100.1800	1
39	1	A	0.013133281	20.0540	82.3651	-1
40	1	A	0.012344603	19.1706	81.7740	1

	<b>Operador</b>	<b>Pose</b>	<b>Erro de pose</b>	<b>Tempo de busca</b>	<b>Tempo total</b>	<b>Sucesso da missão</b>
<b>41</b>	2	B	0.005245180	20.4070	84.7338	1
<b>42</b>	2	B	0.007158873	20.4408	84.3043	1
<b>43</b>	2	B	0.011482759	20.7824	83.2411	1
<b>44</b>	2	B	0.087866430	23.4152	87.9248	-1
<b>45</b>	2	B	0.008514302	19.9797	86.4344	1
<b>46</b>	2	B	0.005144988	19.8502	84.5138	1
<b>47</b>	2	B	0.013200185	19.9392	85.7667	-1
<b>48</b>	2	B	0.004874412	20.2860	80.6176	1
<b>49</b>	1	B	0.012512453	20.7894	85.9667	1
<b>50</b>	1	B	0.014427715	19.4977	81.7383	-1
<b>51</b>	1	B	0.007453902	20.5577	82.4847	1
<b>52</b>	1	B	0.006078841	20.4143	81.6851	1
<b>53</b>	1	B	0.009550672	19.5227	84.7312	1
<b>54</b>	1	B	0.006123619	16.9920	74.6466	1
<b>55</b>	1	B	0.003705978	18.6949	80.1175	1
<b>56</b>	1	B	0.006896455	19.9972	76.2407	1
<b>57</b>	2	B	0.006869856	18.8696	79.7714	1
<b>58</b>	2	B	0.002917472	18.4679	73.9755	1
<b>59</b>	2	B	0.004802752	19.4849	80.2423	1
<b>60</b>	2	B	0.004557056	19.5606	77.7174	1
<b>61</b>	2	B	0.006353444	18.5702	76.3386	1
<b>62</b>	2	B	0.007120232	19.6487	78.9122	1
<b>63</b>	2	B	0.004832266	17.0441	77.3120	1
<b>64</b>	2	B	0.004535360	18.0856	77.5731	1
<b>65</b>	1	B	0.007897063	17.1585	78.1917	1
<b>66</b>	1	B	0.007330963	20.5918	81.1574	1
<b>67</b>	1	B	0.008284862	20.2515	79.0624	1
<b>68</b>	1	B	0.008096144	19.9652	81.4914	1
<b>69</b>	1	B	0.013425916	19.0627	79.8650	-1
<b>70</b>	1	B	0.007966466	18.6261	75.1340	1
<b>71</b>	1	B	0.008486843	18.1093	72.8817	1
<b>72</b>	1	B	0.007759457	18.7750	77.1361	1
<b>73</b>	2	B	0.012489658	20.1018	77.2422	1
<b>74</b>	2	B	0.008122309	18.6224	77.0045	1
<b>75</b>	2	B	0.012097601	20.3431	79.8706	1
<b>76</b>	2	B	0.087776215	17.5113	77.7186	1
<b>77</b>	2	B	0.006999748	17.3791	74.1548	1
<b>78</b>	2	B	0.008223493	18.3666	76.5340	1
<b>79</b>	2	B	0.008721295	19.4229	73.9818	1
<b>80</b>	2	B	0.008375202	18.6691	73.7671	1

Fonte: Autoria própria.

Observando a tabela 14, os testes apresentaram 91.25% de sucesso, ou seja, dos 80 testes realizados, em 73 obteve-se sucesso no pressionamento do botão. A tabela 15 resume alguns indicadores importantes das três saídas do sistema (tempo de busca, tempo total da missão e erro de pose).

Os testes de normalidade apresentaram valores de  $\rho$ -valor inferiores a 0.05, ou seja, a hipótese nula não pode ser rejeitada, logo os dados seguem uma distribuição normal.

Tabela 15: Resumo dos conjuntos de dados das variáveis de saída.

	Média	Desvio padrão	$\rho$ -valor (Shapiro-Wilk)
<b>Tempo de busca</b>	23.95025	5.035711	1.103e-07
<b>Tempo total</b>	86.06149	8.343288	0.02409
<b>Erro de pose</b>	0.01095061	0.0126478	2.2e-16

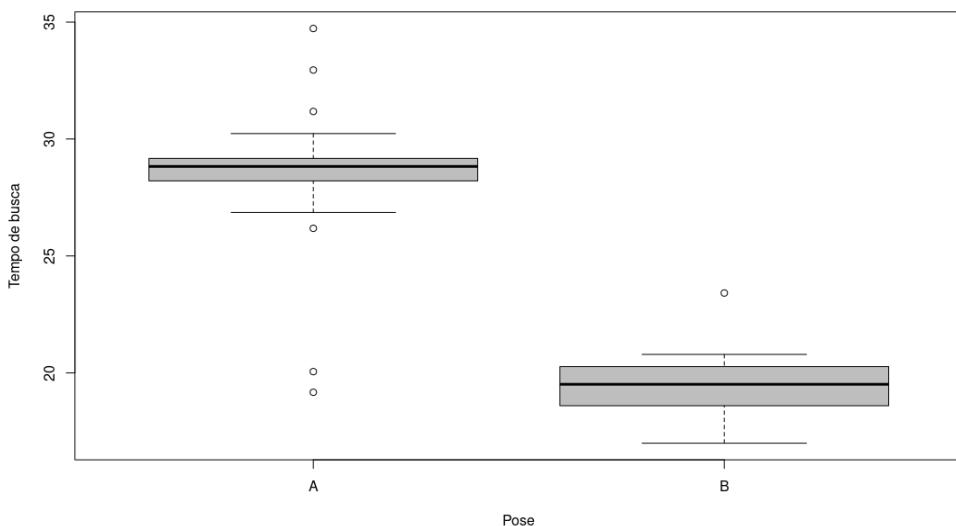
Fonte: Autoria própria.

Tabela 16: Tabela dos valores de  $\rho$ -valor do teste ANOVA.

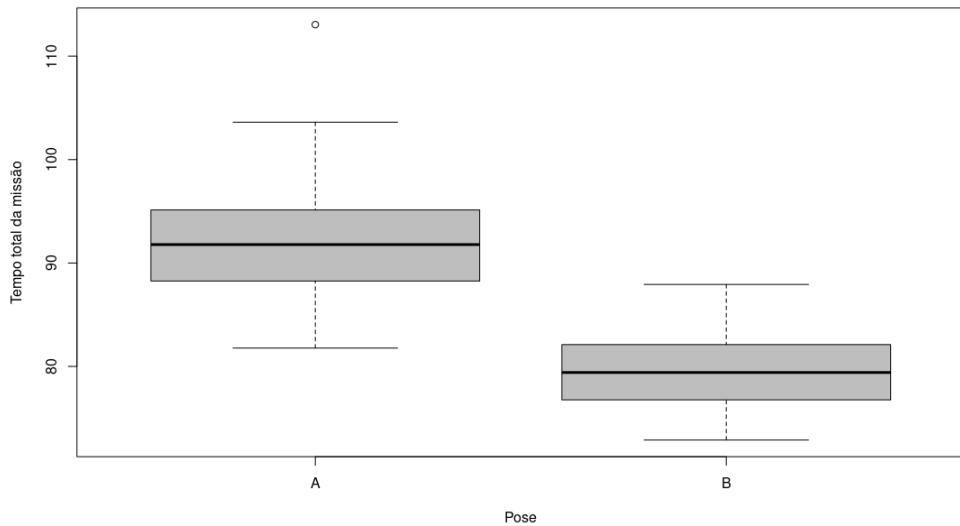
	p-valor
<b>Erro de pose</b>	0.525
<b>Tempo de busca</b>	2.2e-16
<b>Tempo total</b>	2.2e-16

Fonte: Autoria própria.

Realizando o teste do ANOVA, podemos visualizar a variância dos pontos A e B em relação ao erro de pose, tempo de busca e tempo total da missão. A tabela 16 apresenta o  $\rho$ -valor para cada conjunto de dados, e a partir dessa análise podemos chegar a conclusão que a hipótese nula não pode ser rejeitada para o erro de pose assim como visto na análise passada, ou seja, os dados obtidos do erro de posição do *end-effector* nos pontos A e B são estatisticamente iguais. Já os dados dos tempo de busca e tempo total da missão apresentam  $\rho$ -valor menor que 0.05, logo a hipótese nula é rejeitada e pode-se concluir que os dados diferem entre si, como mostram os *boxplots* das figuras 26 e 27.

Figura 26: *Boxplot* da variância do tempo de busca entre os pontos A e B.

Fonte: Autoria própria.

Figura 27: *Boxplot* da variância do tempo total da missão entre os pontos A e B.

Fonte: Autoria própria.

### 5.3.3 Análise de repetibilidade e reproduzibilidade

Como forma de avaliação do sistema robótico, também foi adotado um estudo de repetibilidade e reproduzibilidade do sistema. Com isso, foram adotados os dados da tabela 14. Foram tomadas 20 medições de cada operador para cada pose da caixa, totalizando 80 testes e assim foram obtidos os seguintes dados apresentados na figura 28 e 29.

Observando a figura 28, correspondente ao teste de R&R para o tempo busca do marcador visual, neste teste percebe-se o número de categorias igual a 6, número considerável e acima de valores de referências, o que demonstra que o sistema de medição pode ser utilizado para análise dos dados. Outro ponto importante para observado é o indicador "*Total Gage R&R*", que foi calculado em 20.37, inferior a 30% (Valor máximo aceitável adotado em algumas literaturas).

O sistema sendo aceitável para análise, pode-se observar outros indicadores importantes para análise, como exemplo do  $\rho$ -valor, que mostra a maior influência na variação devido a pose da caixa, não ocorrendo influência do operador ou da interação entre operador e pose. O indicador "*Part-to-Part*" indica que aproximadamente 98% da variação do sistema deve-se a própria variação de pose da caixa, o que demonstra que o sistema possui repetibilidade e reproduzibilidade, quanto a variável de tempo de busca.

Figura 28: Modelo completo teste R&R para tempo de busca.

```

Complete model (with interaction):
      Df Sum Sq Mean Sq F value Pr(>F)
pose_rep        1 1902.1 1902.1 403.599 0.0317
operador       1    0.1    0.1   0.012 0.9308
pose_rep:operador 1    4.7    4.7   2.450 0.1215
Repeatability   80 153.9    1.9
Total          83 2060.7

alpha for removing interaction: 0.05

Reduced model (without interaction):
      Df Sum Sq Mean Sq F value Pr(>F)
pose_rep        1 1902.1 1902.1 971.443 <2e-16
operador       1    0.1    0.1   0.029 0.866
Repeatability  81 158.6    2.0
Total          83 2060.7

Gage R&R

      VarComp %Contrib
Total Gage R&R  1.957994  4.15
  Repeatability  1.957994  4.15
  Reproducibility 0.000000  0.00
    operador     0.000000  0.00
Part-To-Part    45.240977 95.85
Total Variation 47.198971 100.00

      StdDev StudyVar %StudyVar
Total Gage R&R 1.399283 8.39570  20.37
  Repeatability 1.399283 8.39570  20.37
  Reproducibility 0.000000 0.00000  0.00
    operador     0.000000 0.00000  0.00
Part-To-Part    6.726141 40.35685  97.90
Total Variation 6.870151 41.22090 100.00

Number of Distinct Categories = 6

```

Fonte: Autoria própria.

Figura 29: Modelo completo teste R&R para tempo total da missão.

```

Complete model (with interaction):
      Df Sum Sq Mean Sq F value Pr(>F)
pose_rep        1   3659    3659 431.235 0.0306
operador       1      2      2   0.272 0.6941
pose_rep:operador 1      8      8   0.351 0.5550
Repeatability   80   1932     24
Total          83   5602

alpha for removing interaction: 0.05

Reduced model (without interaction):
      Df Sum Sq Mean Sq F value Pr(>F)
pose_rep        1   3659    3659 152.736 <2e-16
operador       1      2      2   0.096 0.757
Repeatability  81   1941     24
Total          83   5602

Gage R&R

      VarComp %Contrib
Total Gage R&R 23.95875  21.68
  Repeatability 23.95875  21.68
  Reproducibility 0.00000  0.00
    operador     0.00000  0.00
Part-To-Part    86.55716  78.32
Total Variation 110.51591 100.00

      StdDev StudyVar %StudyVar
Total Gage R&R 4.894768 29.36861  46.56
  Repeatability 4.894768 29.36861  46.56
  Reproducibility 0.000000 0.00000  0.00
    operador     0.000000 0.00000  0.00
Part-To-Part    9.303610 55.82166  88.50
Total Variation 10.512655 63.07593 100.00

Number of Distinct Categories = 2

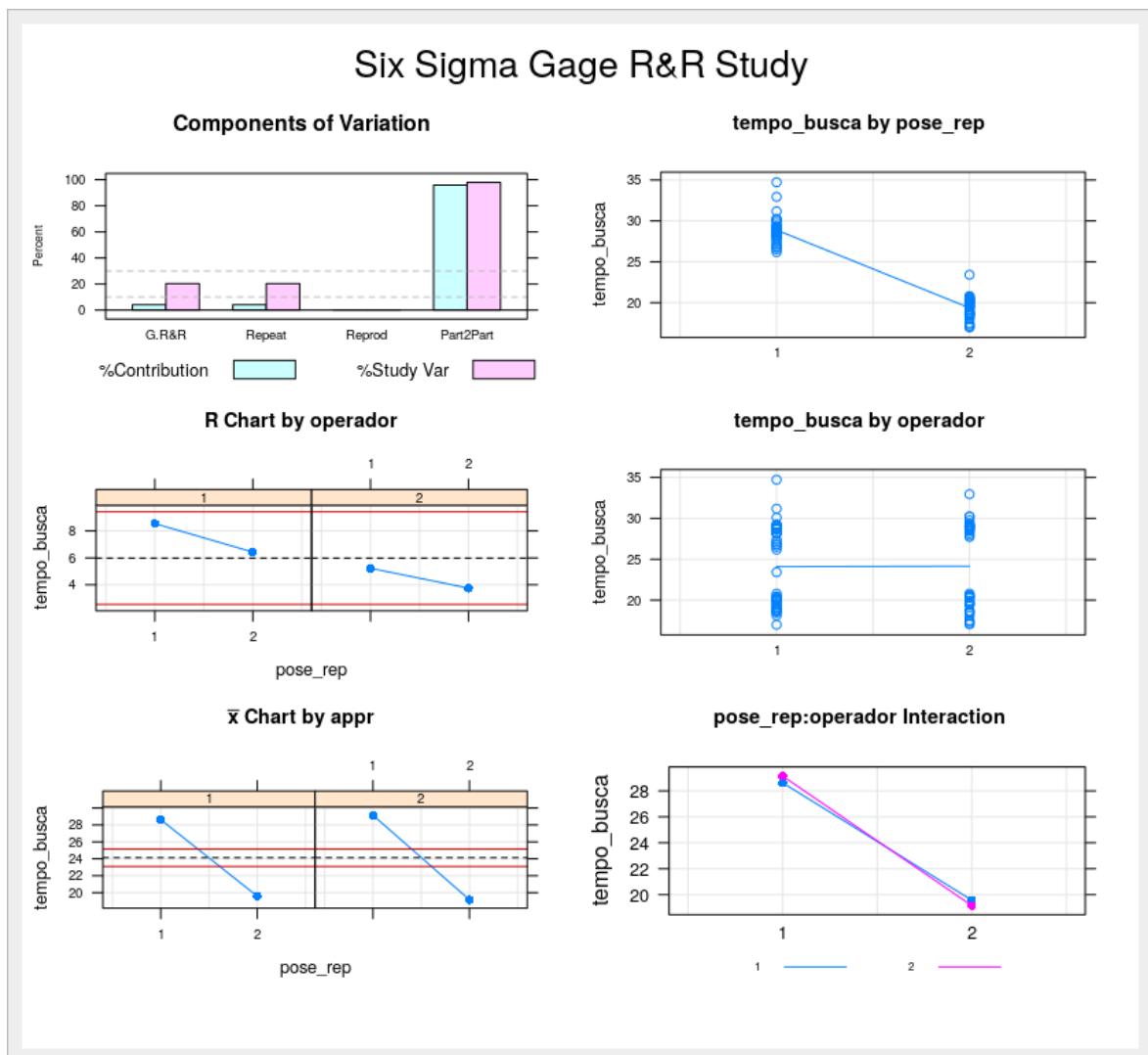
```

Fonte: Autoria própria.

O mesmo não pode ser observado na figura 29, onde o número de categorias apresentados é baixo e o índice "Total Gage R&R" foi calculado em 46.56%, acima do valor máximo de referência (30%), o que mostra que o sistema de medição não é aceitável quando observado a variável de tempo total da missão.

Segundo a análise dos dados obtidos no estudo de repetibilidade e reprodutibilidade do sistema para o tempo de busca, visualizamos cartas na figura 30 que possibilita um melhor entendimento do sistema de medição.

Figura 30: Gráficos do estudo R&R para tempo de busca.



Fonte: Autoria própria.

A carta "Componentes de variação" representa atribuições da variação do sistema, e como observado na figura 30, a variação deve-se principalmente a "peça por peça" ou "parte por parte", o que é desejável, demonstrando que o sistema de medição é aceitável para análise do tempo de busca do manipulador.

A carta R é uma carta de controle de amplitudes que representa graficamente a consistência do operador. Os pontos apresentados na figura 30 estão dentro dos limites máximos e mínimos, ou seja, é sinal de que os operadores estão realizando os testes de forma consistente.

Os pontos apresentados na carta Xbar, na figura 30, os dados estão acima ou abaixo dos limites de controle. Estes resultados indicam que a variação entre as poses são maiores que a variação do dispositivo de medição.

Outro ponto importante de ser destacado é o gráfico de interação entre o operador e o tempo de busca (figura 30), a barra horizontal visualizada, demonstra que os operadores estão realizando medições semelhantes entre si, com variações semelhantes entre as medidas máximas e mínimas do tempo de busca.

## 5.4 Conclusões dos testes estatísticos

Após estudo dos dados obtidos a partir de testes no manipulador robótico, verifica-se que os dados apresentaram distribuição normal, o que possibilita uma análise confiável dos mesmos, comprovando isso através de histogramas apresentados durante a seção.

Também é possível observar nos testes realizados que o algoritmo de planejamento de trajetória RRT CONNECT apresentou melhor resultado em relação ao Kpiece, bem como a velocidade pouco influenciou nos testes, mostrando que a melhor configuração para o sistema para as missões impostas é utilizando o algoritmo RRT.

Não houve diferença no erro de posicionamento do *end-effector* quando observado as poses A e B e com ou sem detecção do marcador visual, ou seja, pode-se concluir que o manipulador robótico apresenta erro constante nas poses analisadas. É observado também que o tempo de busca e tempo total da missão diferem entre as poses.

Por fim, conclui-se que a análise de repetibilidade e reprodutibilidade foi positiva quando observados os dados da variável tempo de busca do marcador visual, uma vez que o sistema de medição atendeu ao teste; enquanto que para a variável tempo total da missão os resultados não foram de todo satisfatórios, indicando possíveis direções de melhoria no sistema.

## 6 CONFIABILIDADE DO SISTEMA

A fim de verificar-se a confiabilidade do sistema, foi realizado o levantamento de cada componente que constitui o manipulador robótico e o estudo detalhado das prováveis falhas. Para isso, foi utilizado o *FMECA* com o propósito de analisar cada sub-sistema, como pode ser visto nas tabelas da seção 6.1.

Na seção 6.2 foi desenvolvida a análise da árvore de falhas do sistema, método que permite através de um processo lógico dedutivo chegar-se às causas básicas de um problema de proporções maiores.

### 6.1 Análise dos modos e efeitos de falhas

Nas Tabelas 17, 18, 19, 20 e 21 estão representadas informações referentes ao estudo sistemático e estrutura das falhas potenciais para os sub-sistemas de potência, de aquisição, estrutural, de atuação e de processamento, respectivamente.

Tabela 17: *FMECA* do sub-sistema de potência

Análise do Tipo e Efeito de Falha								
Sub-sistema de potência								
Função(es)	Modo(s) de falha em potencial	Efeito(s) potencial(is) da falha	S	Causa(s) potenciais	O	D	C	Ação(ões) recomendada(s)
Transmissão de dados	Incapacidade de transmissão de dados	Comprometimento dos dados transmitidos	6	Fissura dos fios	3	5	90	Manutenções preventivas
		Perda da capacidade de transmissão de dados	8	Desgaste causado pelo tempo	2	1	16	
Transmissão de energia	Incapacidade de transmissão de energia	Perda da capacidade de transmissão de energia	8	Derretimento por temperatura elevada	1	1	8	Manutenções preventivas
		Má qualidade da energia transmitida	6	Torção dos fios	3	5	90	
Alimentação do sistema	Incapacidade em fornecer energia	Não energização do sistema	9	Má conexão com fonte de tensão	3	6	162	Verificar conexão com fonte
				Desgaste nas soldas causado pelo tempo	4	7	252	

Fonte: Autoria própria.

Tabela 18: *FMECA* do sub-sistema aquisição

Análise do Tipo e Efeito de Falha								
Sub-sistema de aquisição								
Função(es)	Modo(s) de falha em potencial	Efeito(s) potencial(is) da falha	S	Causa(s) potenciais	O	D	C	Ação(ões) recomendada(s)
Aquisição de dados visuais	Incapsidade de obter dados visuais	Perda da capacidade de obter dados visuais	8	Obstrução do campo de visão da câmera	2	1	16	Verificar condições do ambiente antes das missões
	Ruptura da estrutura de fixação da câmera			Baixa visibilidade do ambiente	2	5	80	
	Coleta de dados inconsistentes ou insuficientes			Calibração incorreta da câmera	3	1	24	
		Comprometimento dos dados	6	Set-up incorreto	2	7	84	Verificar cabeamento do sistema
				Falha no canal de comunicação	4	7	168	

Fonte: Autoria própria.

Tabela 19: *FMECA* do sub-sistema estrutural

Análise do Tipo e Efeito de Falha									
Sub-sistema estrutural									
Função(es)	Modo(s) de falha em potencial	Efeito(s) potencial(is) da falha	S	Causa(s) potenciais	O	D	C	Ação(es) recomendada(s)	
Ligar o frame ao perfil de alumínio	Incapacidade de sustentar o perfil	Ruptura do conector	8	Excesso de peso aplicado	2	1	16	Verificar a fabricação da peça	
				Desgaste pelo tempo	2	5	80		
	Compor estrutura mecânica	Trinca do conector	6	Set-up incorreto	4	6	144	Verificar as condições de peso aplicadas	
				Defeito de fabricação	5	6	180		
Danificação estrutural	Ruptura do perfil	8	Sobrecarga	2	4	64	Cuidado no manuseamento dos perfis		
	Folga entre conexões	6	Colisão	4	2	48	Definir área de trabalho		

Fonte: Autoria própria.

Tabela 20: *FMECA* do sub-sistema de atuação

Análise do Tipo e Efeito de Falha									
Sub-sistema de atuação									
Função(es)	Modo(s) de falha em potencial	Efeito(s) potencial(is) da falha	S	Causa(s) potenciais	O	D	C	Ação(es) recomendada(s)	
Movimentação do manipulador	Rotação inconsistente com o desejado	Perda de desempenho (velocidade; controle)	6	Emperramento parcial	4	6	144	Inspecções periódicas	
Fornecer o torque mínimo de sustentação	Curto circuito	Parada do motor	8	Cabeamento incorreto	2	5	80	Medir corrente nos terminais	
	Perda de comunicação	Perda de desempenho	6	Falha na alimentação	4	5	120		
	Tensão insuficiente								

Fonte: Autoria própria.

Tabela 21: *FMECA* do sub-sistema de processamento

Análise do Tipo e Efeito de Falha								
Sub-sistema de processamento								
Função(es)	Modo(s) de falha em potencial	Efeito(s) potencial(is) da falha	S	Causa(s) potenciais	O	D	C	Ação(es) recomendada(s)
Processamento dos dados	Não processamento dos dados do sistema principal	Não funcionamento	9	Queima/ danificação de componentes	3	8	216	Manutenção preditiva
				Falha no sistema operacional	2	8	144	Executar teste de verificação do software
				Falha na porta de comunicação	2	2	36	Executar teste de entrada e saída de dados com <i>NUC</i>
				Problemas de contato elétrico	2	3	54	Inspecções periódicas em bancada

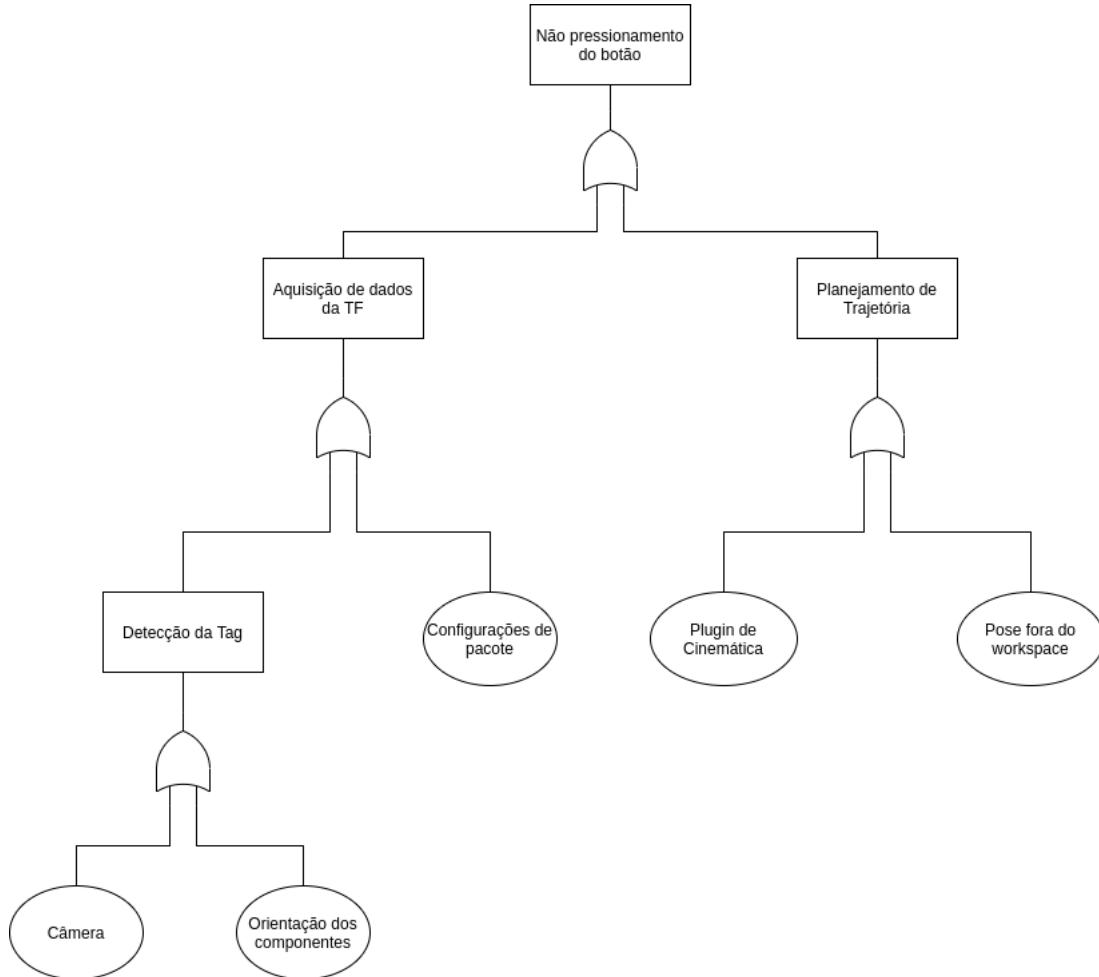
Fonte: Autoria própria.

## 6.2 Análise da árvore de falhas

Na Figura 31 encontra-se a árvore de falhas do sistema do manipulador JeRoTIMON. Observa-se que caso o manipulador não consiga realizar a tarefa de pressionar o botão, duas causas principais devem ser investigadas: a conexão entre as TF's, realizada a partir do pacote *Bir Marker Localization*, ou o planejamento de trajetória, realizado pelo *Moveit*.

Para a primeira causa, devem ser verificadas as configurações do pacote e da câmera, bem como a orientação dos componentes envolvidos no processo de detecção. Já para solucionar os problemas correspondentes ao planejamento de trajetória, deve-se observar se a pose desejada está dentro do *workspace* do manipulador e se o *plugin* de cinemática inversa utilizado está sendo capaz de realizar as conversões necessárias para a pose desejada.

Figura 31: Árvore de falhas do sistema.



Fonte: Autoria própria.



## 7 GESTÃO DO CONHECIMENTO

Neste capítulo estão descritas as lições aprendidas durante o processo de desenvolvimento do protótipo que foram criadas a partir da comparação entre o que era esperado e o que realmente aconteceu em cada etapa do projeto. Esta sub-seção engloba desde a fase inicial do planejamento até a construção do modelo real.

Além das lições aprendidas, as seções 7.2 e 7.3 trazem os guias de uso para a simulação e o modelo real, respectivamente, com o propósito de auxiliar o usuário na replicação dos experimentos realizados neste relatório.

### 7.1 Lições aprendidas

A Tabela 22 mostra como foi estruturada cada lição aprendida abordando os seguintes aspectos: Tema, Fase, Impacto, O que ocorreu?, Como resolveu?, Resultados e Recomendações para os próximos projetos. O objetivo deste estudo é a correção dos impactos negativos para os projetos subsequentes.

Tabela 22: Lições aprendidas

LIÇÕES APRENDIDAS						
Tema	Fase	Impacto	O que ocorreu?	Como resolveu?	Resultados	Recomendações para os próximos projetos
Gestão	Planejamento	Negativo	Ausência de uma metodologia de trabalho	Reunião com foco em definir metodologia de projeto	Evolução na comunicação dos membros	Antes de começar o projeto realizar reunião para definição de metodologia
Gestão	Planejamento	Negativo	Ausência de uma ferramenta para gestão de projeto	Escolha de uma ferramenta gratuita e de fácil aplicação	Melhoria na organização das atividades	Definição prévia de uma ferramenta de gestão de projeto
Gestão	Execução	Negativo	Montagens e desmontagens do manipulador	Planejamento prévio das atividades	Otimização do tempo	Cronograma definido as atividades a serem realizadas
Tecnológico	Execução	Negativo	Falha no dimensionamento das peças	Consultoria sobre materiais aplicados na confecção	Obtenção de peças com maior resistência mecânica	Pesquisa e consultoria prévia antes da modelagem das peças
Tecnológico	Execução	Negativo	Dificuldade em planejamento de trajetória para determinadas poses do robô	Utilização do plugin Track-IK	Maior eficiência de planejamento	Pesquisa e consultoria prévia do pacote mais adequado para o projeto
Tecnológico	Execução	Negativo	Motor dynamixel (H54-200-S500-R PRO) parou de funcionar	Realizado a Análise 8D para fazer investigação do ocorrido	Compreensão de que há procedimentos a serem feitos antes de utilizar um produto	Realizar leitura meticolosa do manual do produto para saber quais são os procedimentos

Fonte: Autoria própria.

### 7.2 Guia de uso para simulação

Para replicar a simulação do manipulador robótico Timon-HM é necessário seguir os passos descritos nesta seção. Recomenda-se a utilização do Ubuntu 18.04 LTS e o *ROS Melodic Morenia*.

Antes de inserir o pacote do manipulador no *workspace* é fundamental que sejam instalados os pacotes requeridos para este. Primeiramente, no terminal, segue-se os comandos listados:

- Instalar MoveIt:

```
$ sudo apt-get install ros-melodic-moveit
```

- Instalar pacote de ferramentas visuais do MoveIt:

```
$ sudo apt-get install ros-melodic-moveit-visual-tools
```

- Instalar TRAC-IK para resolução da cinemática:

```
$ sudo apt-get install ros-melodic-trac-ik-kinematics-plugin
```

- Instalar pacote de controle no *Gazebo ROS*:

```
$ sudo apt-get install ros-melodic-gazebo-ros-control
```

- Instalar pacote do controlador do *ROS*:

```
$ sudo apt-get install ros-melodic-controller-*
```

- Instalar pacote controlador de posição do *ROS*:

```
$ sudo apt-get install ros-melodic-position-controller
```

- Instalar pacote controlador de esforço do *ROS*:

```
$ sudo apt-get install ros-melodic-effort-controller
```

- Instalar pacote de juntas do *ROS*:

```
$ sudo apt install ros-melodic-joint-*
```

- Instalar pacote de controle do *ROS*:

```
$ sudo apt install ros-melodic-ros-control
```

Antes de instalar o pacote *bir\_marker\_localization* é necessária a instalação do *OpenCV* versão 3.3.1, este possui o guia de instalação próprio disponível em:

<https://www.learnopencv.com/install-opencv3-on-ubuntu/>.

Após a instalação do *OpenCV*, para clonar o repositório do *bir\_marker\_localization* segue-as as seguintes etapas, no terminal:

- Criar um *workspace* para adicionar dentro deste os pacotes que serão usados na simulação.

```
$ mkdir nomedoworkspace_ws
```

- Entrar no *workspace*:

```
$ cd nomedoworkspace_ws
```

- Criar a pasta *source*<sup>1</sup>:

```
$ mkdir src
```

- Entrar no *source*:

```
$ cd src
```

- Clonar o repositório *Bir Marker Localization* para *workspace* (Verificar qual a *branch* estável):

```
$ git clone https://github.com/Brazilian-Institute-of-Robotics/bir_marker_localization.git
```

- Clonar o pacote do manipulador para dentro da pasta src:

```
$ git clone -b feature/simulation https://github.com/Brazilian-Institute-of-Robotics/timon_hm_manipulator.git
```

- Clonar o pacote do *Open Manipulator* para dentro da pasta src:

```
$ git clone https://github.com/ROBOTIS-GIT/open_manipulator_msgs.git
```

- Retornar para a raiz do *workspace*:

```
$ cd ..
```

- Compilar o *workspace*:

```
$ catkin_make
```

- Ativar o ambiente virtual do *workspace*:

```
$ source devel/setup.bash
```

---

<sup>1</sup> Pasta onde contém os arquivos fonte.

Após realizar os procedimentos citados o *workspace* estará configurado para executar a simulação, com os seguintes comandos (cada comando terá que ser inserido em uma aba do terminal):

- Iniciar a simulação no *Gazebo*:

```
$ roslaunch manipulator_gazebo gazebo.launch
```

- Iniciar pacote MoveIt do Timon-HM:

```
$ roslaunch manipulator_gazebo moveit_demo.launch
```

- Iniciar o *bir\_marker\_localization*:

```
$ roslaunch timon_demo bir_marker_localization.launch
```

- Iniciar o algoritmo de busca do marcador visual e para acionar o painel elétrico:

```
$ roslaunch timon_demo push_button_simulation.launch
```

### 7.3 Guia de uso para o modelo real

Para replicar o modelo real do manipulador a versão do Ubuntu, do *ROS* e os pacotes necessários são os mesmos descritos na seção 7.2. Após a instalação dos pacotes, a seguir estão descritas as etapas subsequentes:

- Criar *workspace* e a pasta *source*:

```
$ mkdir -p catkin_ws/src
```

- Entrar no *workspace* e no *source*:

```
$ cd catkin_ws/src
```

- Clonar para dentro da pasta *source* o repositório do manipulador:

```
$ git clone https://github.com/Brazilian-Institute-of-Robotics/timon_hm_manipulator.git
```

- Clonar para dentro do *source* o repositório do *Bir Marker Localization*:

```
$ git clone -b final_settings https://github.com/Brazilian-Institute-of-Robotics/bir_marker_localization.git
```

- Clonar para dentro do *source* o repositório do *Open Manipulator*:

```
$ git clone https://github.com/ROBOTIS-GIT/  
open_manipulator_msgs.git
```

- Clonar para dentro do *source* o repositório do *Dynamixel workbench*:

```
$ git clone https://github.com/ROBOTIS-GIT/dynamixel-  
workbench.git
```

- Clonar para dentro do *source* o repositório da câmera *Teledyne*:

```
$ git clone -b refactor_code https://github.com/Brazilian  
-Institute-of-Robotics/def_cam_teledyne_nano.git
```

- Retornar para a raiz do *workspace*:

```
$ cd ..
```

- Compilar o *workspace*:

```
$ catkin_make
```

- Ativar o ambiente virtual do *workspace*:

```
$ source devel/setup.bash
```

Para executar a aplicação é necessário realizar os seguintes comandos (cada comando terá que ser inserido em uma aba do terminal):

- Iniciar os controladores do manipulador:

```
$ roslaunch timon_arm_controller dxl_controllers.launch
```

```
$ roslaunch timon_arm_controller moveit.launch
```

```
$ roslaunch timon_arm_controller dxl_moveit_bridge.launch
```

- Iniciar a câmera:

```
$ roslaunch def_cam_teledyne_nano camera_example.launch
```

- Iniciar o *Bir Maker Localization*:

```
$ roslaunch timon_demo bir_marker_localization.launch
```

- Executar a missão:

```
$ roslaunch timon_demo push_button_real.launch
```

## 8 CONCLUSÃO

O presente relatório descreveu a idealização, simulação e construção do JeRoTIMON, um manipulador robótico com 5 *DoF*, integrado ao *ROS*, capaz de acionar um painel elétrico a partir da localização de um marcador visual *ArUco*. A estrutura física do robô foi concebida de acordo com os modelos *URDF* projetados.

Para possibilitar o uso da ferramenta *Moveit*, arquivos de configuração foram gerados e sua comunicação com o modelo simulado no *Gazebo* foi estabelecida. Para resolver as equações de cinemática inversa optou-se pelo plugin TRAC-IK e para o planejamento de trajetória foi utilizada a biblioteca *OMPL*.

Foi desenvolvido em linguagem C++ um pacote capaz de comunicar o sistema de escaneamento e o sistema de planejamento/execução. A utilização da câmera Teledyne Genie Nano C2590 possibilitou a identificação da *tag ArUco* com precisão, tornando o sistema capaz de localizar o painel elétrico e planejar uma trajetória que leve o *endeffector* do manipulador até o alvo estabelecido.

Foram realizados 80 testes considerando diferentes algoritmos, posições e orientações para o painel elétrico. Os resultados alcançados mostram que JeRoTIMON foi capaz de realizar a tarefa em 91.25% dos casos, com tempo médio de busca de 23.95 segundos e o tempo médio de execução de 86.06 segundos, resultados estes considerados satisfatórios para o prosseguimento do projeto.

Para sua próxima aplicação, o manipulador será instalado em um robô móvel *Warthog* que estará equipado com sensores de localização, mapeamento e detecção de obstáculos. De forma autônoma, este conjunto irá localizar e desarmar uma bomba hipotética instalada em ambiente aberto.



## REFERÊNCIAS

16MM C Series VIS-NIR Fixed Focal Length Lens. s.d. <<https://bit.ly/35zzQMq>>. Acessado: 26-10-2020. Citado na página 36.

BEESON, P.; AMES, B. Trac-ik: An open-source library for improved solving of generic inverse kinematics. In: IEEE. *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. [S.l.], 2015. p. 928–935. Citado na página 29.

BIR Marker Localization. 2020. <[https://github.com/Brazilian-Institute-of-Robotics/bir\\_marker\\_localization/](https://github.com/Brazilian-Institute-of-Robotics/bir_marker_localization/)>. Accessed: 2020-04-24. Citado na página 27.

ERTHAL, J. L. Estudo de métodos para a solução da cinemática inversa de robôs industriais para implementação computacional. *UFSC, Programa de Pós-Graduação*, 10 1992. Citado na página 13.

HERNANDEZ-MENDEZ, S. et al. Design and implementation of a robotic arm using ros and moveit! In: IEEE. *2017 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*. [S.l.], 2017. p. 1–6. Citado na página 18.

HERNÁNDEZ-ORDÓÑEZ, M. et al. An education application for teaching robot arm manipulator concepts using augmented reality. *Mobile Information Systems*, Hindawi, v. 2018, 2018. Citado na página 18.

INDUSTRIAL Robots: Robot Investment Reaches Record 16.5 billion USD. 2019. <<https://ifr.org/ifr-press-releases/news/robot-investment-reaches-record-16.5-billion-usd/>>. Acesso em: 15-04-2020. Citado na página 13.

ISO-8373. *Robots and robotic devices — Vocabulary*. Geneva, CH, 2012. Citado na página 17.

JAVEED, A.; PRAKASH, V. G.; KULKARNI, S. P. Autonomous service robot. In: *Proceedings of the Advances in Robotics 2019*. [S.l.: s.n.], 2019. p. 1–6. Citado na página 19.

KHAN, K. A.; KONDA, R. R.; RYU, J.-C. Ros-based control for a robot manipulator with a demonstration of the ball-on-plate task. *Advances in robotics research*, v. 2, n. 2, p. 113, 2018. Citado na página 19.

LEITE, A. C. *Controle Híbrido de Força e Visão de um Manipulador Robótico Sobre Superfícies Desconhecidas*. Tese (Doutorado) — Tese de Mestrado, Departamento de Engenharia Elétrica, 2005. Citado na página 14.

PIMENTA, T. T. Controle de manipuladores robóticos. *PUC-Rio de Janeiro, Departamento de Engenharia de Controle e automação*, 12 2009. Citado na página 13.

ROBOTIS. s.d. <<http://www.robotis.us/>>. Acessado: 19-10-2020. Citado 2 vezes nas páginas 38 e 81.

ROMANO, V. F. *Robótica Industrial: Aplicação na indústria de manufatura e processos*. São Paulo: Edgar Blucher, 2002. v. 256. Citado na página 13.

SANTOS, V. M. Robótica industrial. *Universidade de Aveiro-Departamento de Engenharia Mecânica*, 2004. Citado na página 18.

SPONG, M. W.; HUTCHINSON, S.; VIDYASAGAR, M. Robot modeling and control, john wiley & sons. Inc., ISBN-100-471-649, 2005. Citado na página 13.

SUCAN, I. A.; MOLL, M.; KAVRAKI, L. E. The open motion planning library. *IEEE Robotics & Automation Magazine*, IEEE, v. 19, n. 4, p. 72–82, 2012. Citado na página 29.

TELEDYNE Dalsa. s.d. <<https://www.teledynedalsa.com/en/products/imaging/cameras/genie-nano-1gige/>>. Acessado: 19-10-2020. Citado na página 36.

U2D2. s.d. <<https://emanual.robotis.com/docs/en/parts/interface/u2d2/>>. Acessado: 26-10-2020. Citado 2 vezes nas páginas 38 e 39.

UWE-12/10-Q12PB-C. s.d. <<https://bit.ly/31i2SyP>>. Acessado: 19-10-2020. Citado na página 38.

ZHANG, S.; LIN, Z.; WU, G. Motion planning of a 5-dof anthropomorphic robotic arm under ros environment. In: SPRINGER. *IFToMM International Conference on Mechanisms, Transmissions and Applications*. [S.l.], 2019. p. 409–418. Citado na página 19.



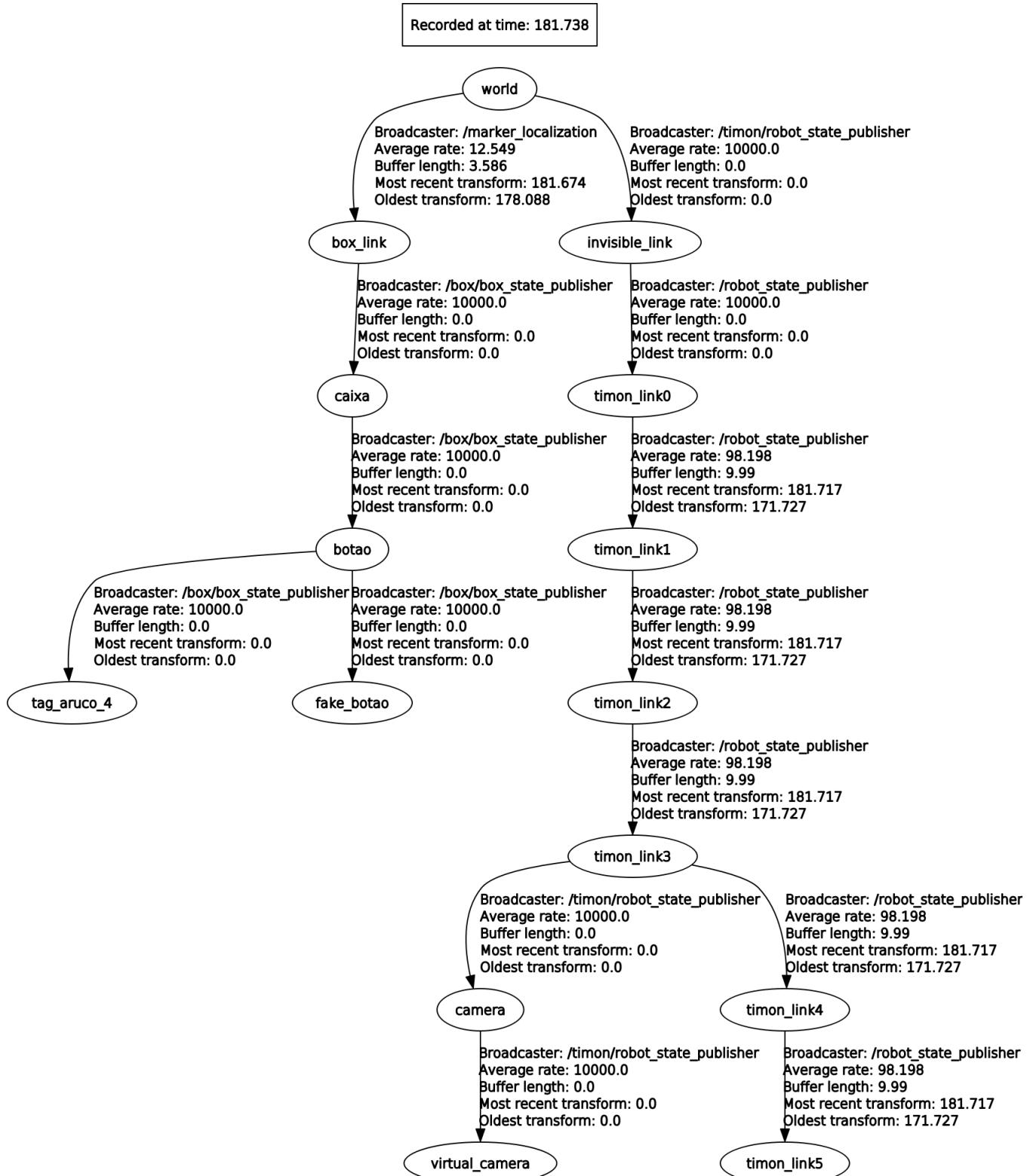
## APÊNDICE A

## Árvores de TF desconectadas



## APÊNDICE B

## Árvores de TF conectadas

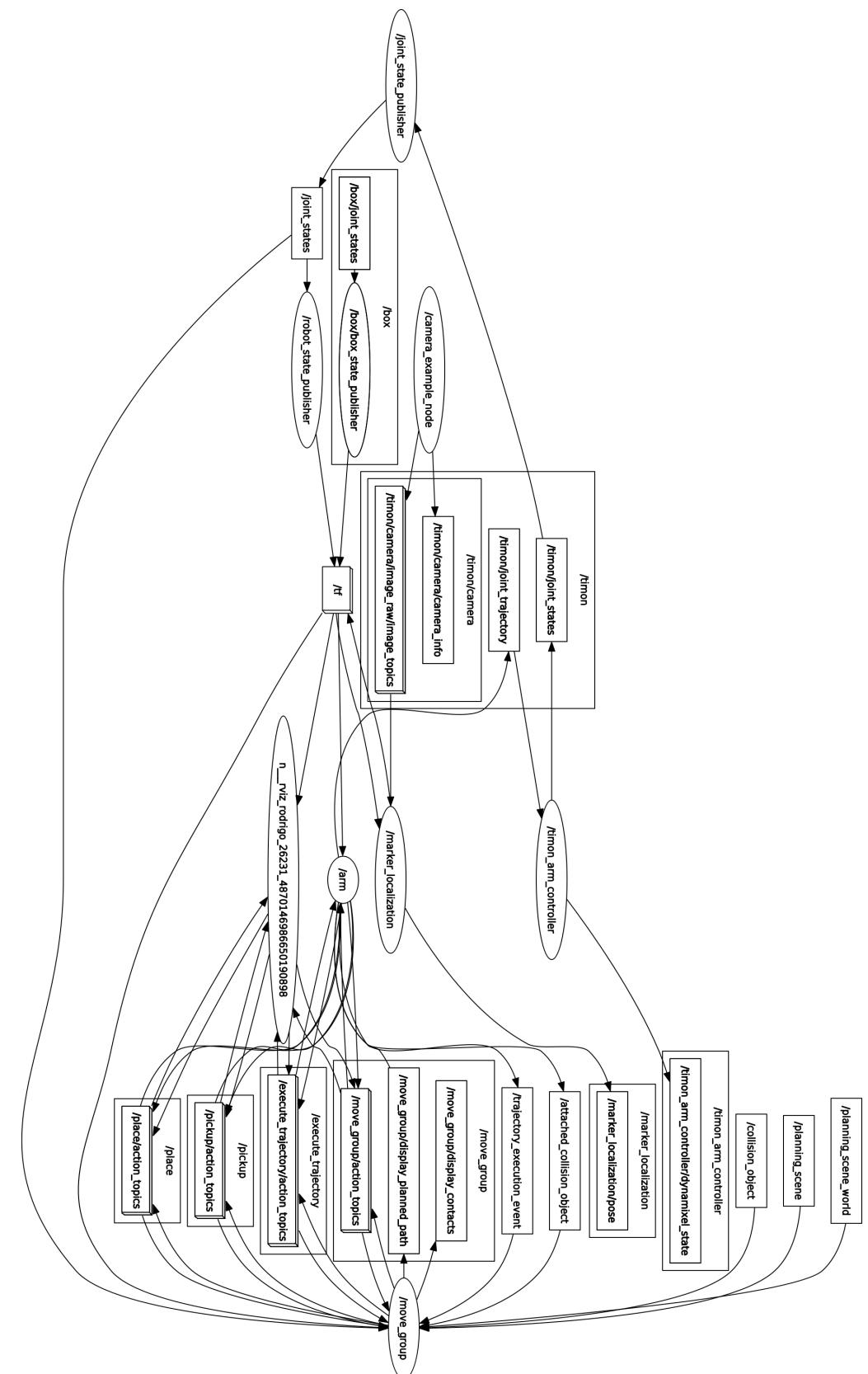






## APÊNDICE C

## Diagrama de nós do sistema



## APÊNDICE D

### Propriedades de Massa do JeRoTIMON

Figura 32: Coordenadas.

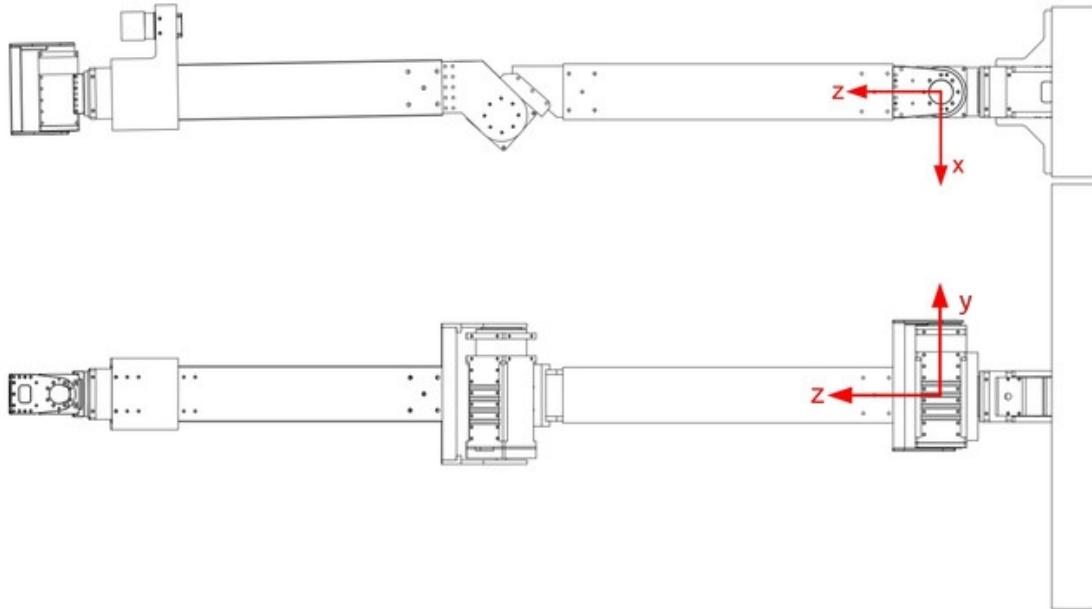
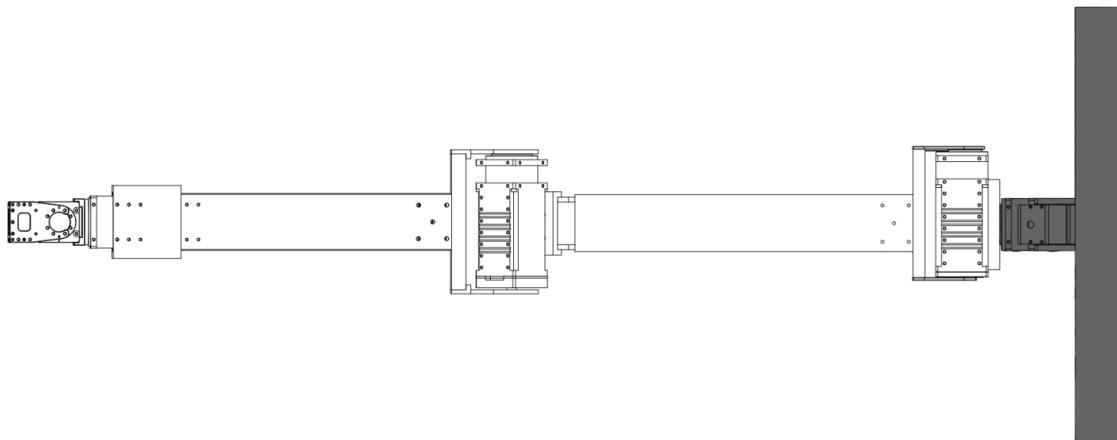


Figura 33: Link 0.



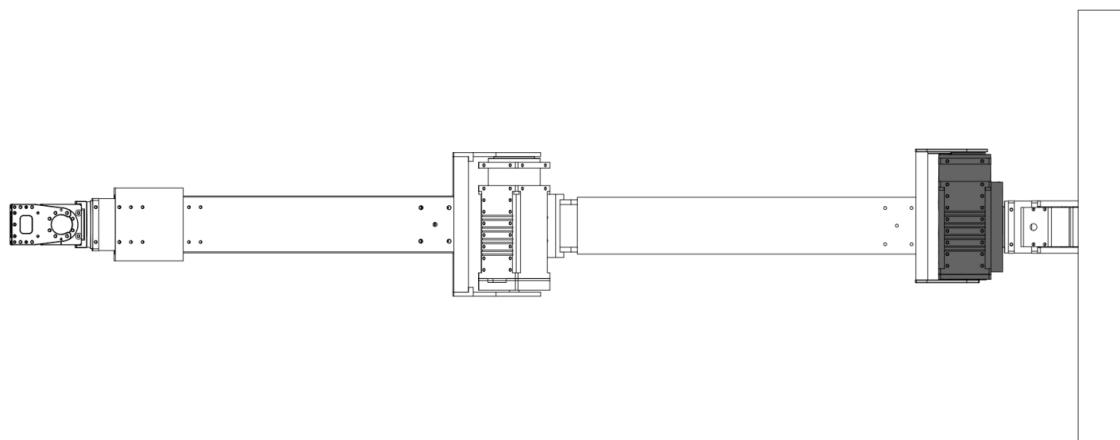
- **Massa:** 3.72384654 kg
- **Volume:** 0.00413037 $m^3$
- **Área:** 0.28622123 $m^2$
- **Centro de Massa:**

- X: -0.00000147 m
- Y: 0.00000000 m
- Z: 0.03504069 m

- **Momento de inércia:** kg  $m^2$

- **L<sub>xx</sub>:** 0.01072644 **L<sub>xy</sub>:**-9.465e-9 **L<sub>xz</sub>:**-3.247e-8
- **L<sub>yx</sub>:** -9.465e-9 **L<sub>yy</sub>:** 0.04865651 **L<sub>yz</sub>:** 6.830e-13
- **L<sub>zx</sub>:** -3.247e-8 **L<sub>zy</sub>:** 6.830e-13 **L<sub>zz</sub>:** 0.05388213

Figura 34: Link 1.



- **Massa:** 1.03781084 kg

- **Volume:** 0.00040169  $m^3$

- **Área:** 0.05853078  $m^2$

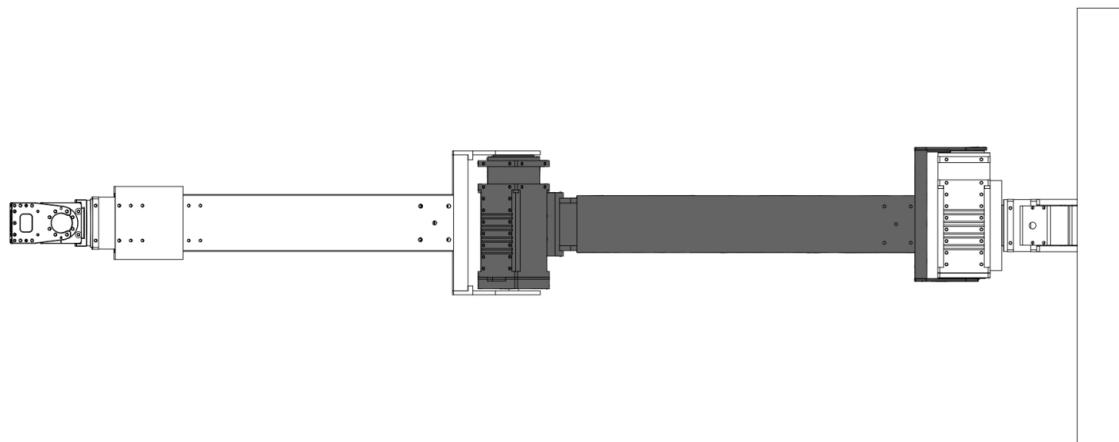
- **Centro de Massa:**

- X: 0.00814457 m
- Y: 4.45597047e - 8 m
- Z: 0.16022275 m

- **Momento de inércia:** kg  $m^2$

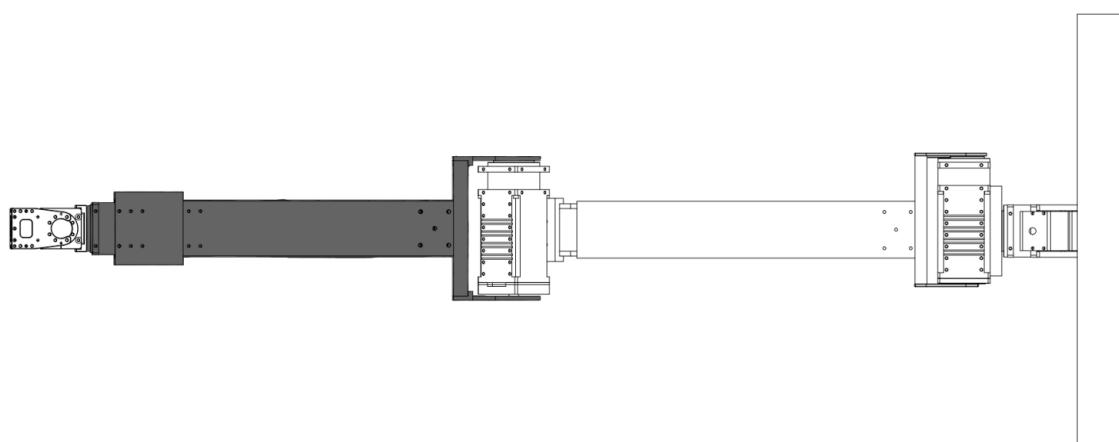
- **L<sub>xx</sub>:** 0.0005687 **L<sub>xy</sub>:** 2.602e-10 **L<sub>xz</sub>:** -0.00004027
- **L<sub>yx</sub>:** 2.602e-10 **L<sub>yy</sub>:** 0.00166759 **L<sub>yz</sub>:** -2.222e-10
- **L<sub>zx</sub>:** -0.00004027 **L<sub>zy</sub>:** -2.222e-10 **L<sub>zz</sub>:** 0.00155695

Figura 35: Link 2.



- **Massa:** 2.05026959 kg
- **Volume:** 0.00077667  $m^3$
- **Área:** 0.29790369  $m^2$
- **Centro de Massa:**
  - X: 0.00294129 m
  - Y: -0.01058166 m
  - Z: 0.48960322 m
- **Momento de inércia:** kg  $m^2$ 
  - **L<sub>xx</sub>:** 0.06174738 **L<sub>xy</sub>:** -0.00003476 **L<sub>xz</sub>:** 0.00098626
  - **L<sub>yx</sub>:** -0.00003476 **L<sub>yy</sub>:** 0.06319525 **L<sub>yz</sub>:** 0.00311701
  - **L<sub>zx</sub>:** 0.00098626 **L<sub>zy</sub>:** 0.00311701 **L<sub>zz</sub>:** 0.0034029

Figura 36: Link 3.



- **Massa:** 1.83580505 kg

- **Volume:** 0.00081683  $m^3$

- **Área:** 0.32055793  $m^2$

- **Centro de Massa:**

- X: 0.00257384 m

- Y: 0.00207951 m

- Z: 0.91287054 m

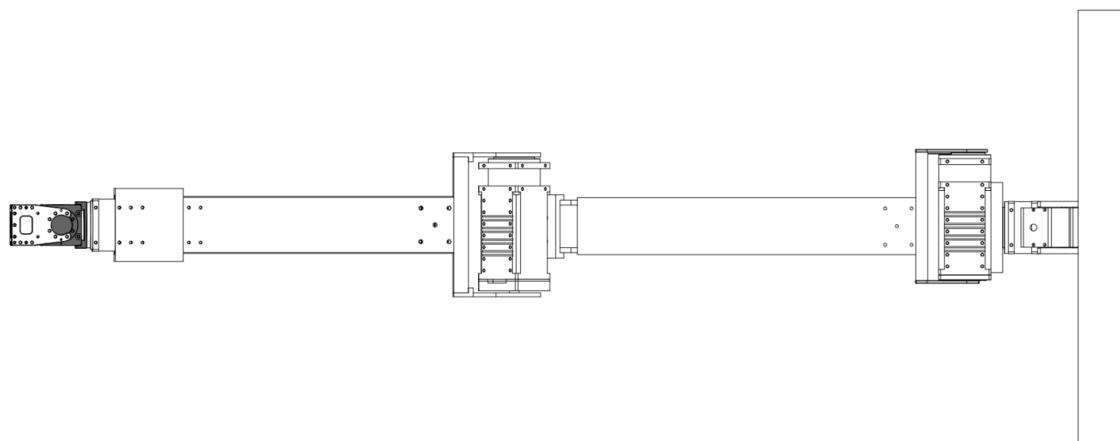
- **Momento de inércia:** kg  $m^2$

- **L<sub>xx</sub>:** 0.03439469 **L<sub>xy</sub>:** -0.00000261 **L<sub>xz</sub>:** -0.00001194

- **L<sub>yx</sub>:** -0.00000261 **L<sub>yy</sub>:** 0.03489631 **L<sub>yz</sub>:** -0.0001288

- **L<sub>zx</sub>:** -0.00001194 **L<sub>zy</sub>:** -0.0001288 **L<sub>zz</sub>:** 0.0023687

Figura 37: Link 4.



- **Massa:** 0.39425655 kg

- **Volume:** 0.00014602  $m^3$

- **Área:** 0.03018022  $m^2$

- **Centro de Massa:**

- X: 0.00257156 m

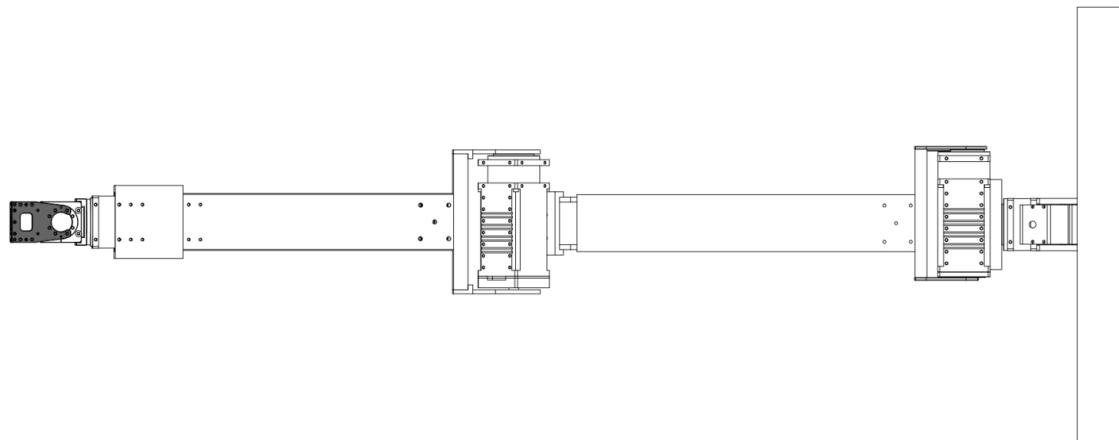
- Y: 0.00133073 m

- Z: 1.09673426 m

- **Momento de inércia:** kg  $m^2$

- **Lxx:** 0.00028795 **Lxy:** 2.010e-10 **Lxz:** -8.166e-13
- **Lyx:** 2.010e-10 **Lyy:** 0.00011981 **Lyz:** -0.00000446
- **Lzx:** -8.166e-13 **Lzy:** -0.00000446 **Lzz:** 0.0002842

Figura 38: Link 5.



- **Massa:** 0.08694786 kg
- **Volume:** 0.0000322  $m^3$
- **Área:** 0.02435944  $m^2$
- **Centro de Massa:**
  - X: 0.00257155 m
  - Y: 0.00684212 m
  - Z: 1.13578562 m
- **Momento de inércia:** kg  $m^2$ 
  - **Lxx:** 0.00012963 **Lxy:** -5.235e-13 **Lxz:** -9.996e-14
  - **Lyx:** -5.235e-13 **Lyy:** 0.00003703 **Lyz:** 0.00000247
  - **Lzx:** -9.996e-14 **Lzy:** 0.00000247 **Lzz:** 0.00012613



## APÊNDICE E

### Lista de suportes.

Tabela 23: Tabela de suportes

Imagen	Junta	Fornecedor	Tipo	Modelo	Quantidade	Medidas (mm)
	Id_1	CCRoSA	Cantoneira	Original	2	59 x 59 x 40.5
	Id_2	CCRoSA	Frame Perpendicular	Original	1	92.6 x 51.1 x 12
	Id_2	ROBOTIS	Rolamento	FRP54-I110K	1	54 x 54 x 5
	Id_3	ROBOTIS/CCRoSA	Frame de Rotação	FRP54-H221K/Original	1	105.5 x 138 x 54.3
	Id_3	CCRoSA	Fixador	Original	1	95 x 54 x 54
	Id_3	CCRoSA	Fixador	Original	1	60 x 65.2 x 17
	Id_3	CCRoSA	Calço	Original	1	54 x 54 x 2
	Id_3	CCRoSA	Rolamento	Original	1	54 x 54 x 10
	Id_4	CCRoSA	Frame de Rotação	Original	1	85.03 x 89.46 x 148.2
	Id_5	ROBOTIS	Frame Perpendicular	FRP42-A110K	1	48 x 42 x 11.5
	Id_5	ROBOTIS	Rolamento	FRP42-I110K	1	42 x 42 x 5
	Id_6	ROBOTIS	Frame de Rotação	FRP42-H121K	1	96 x 42 x 56.6

Fonte: Adaptado de ([ROBOTIS, s.d.](#))



## APÊNDICE F

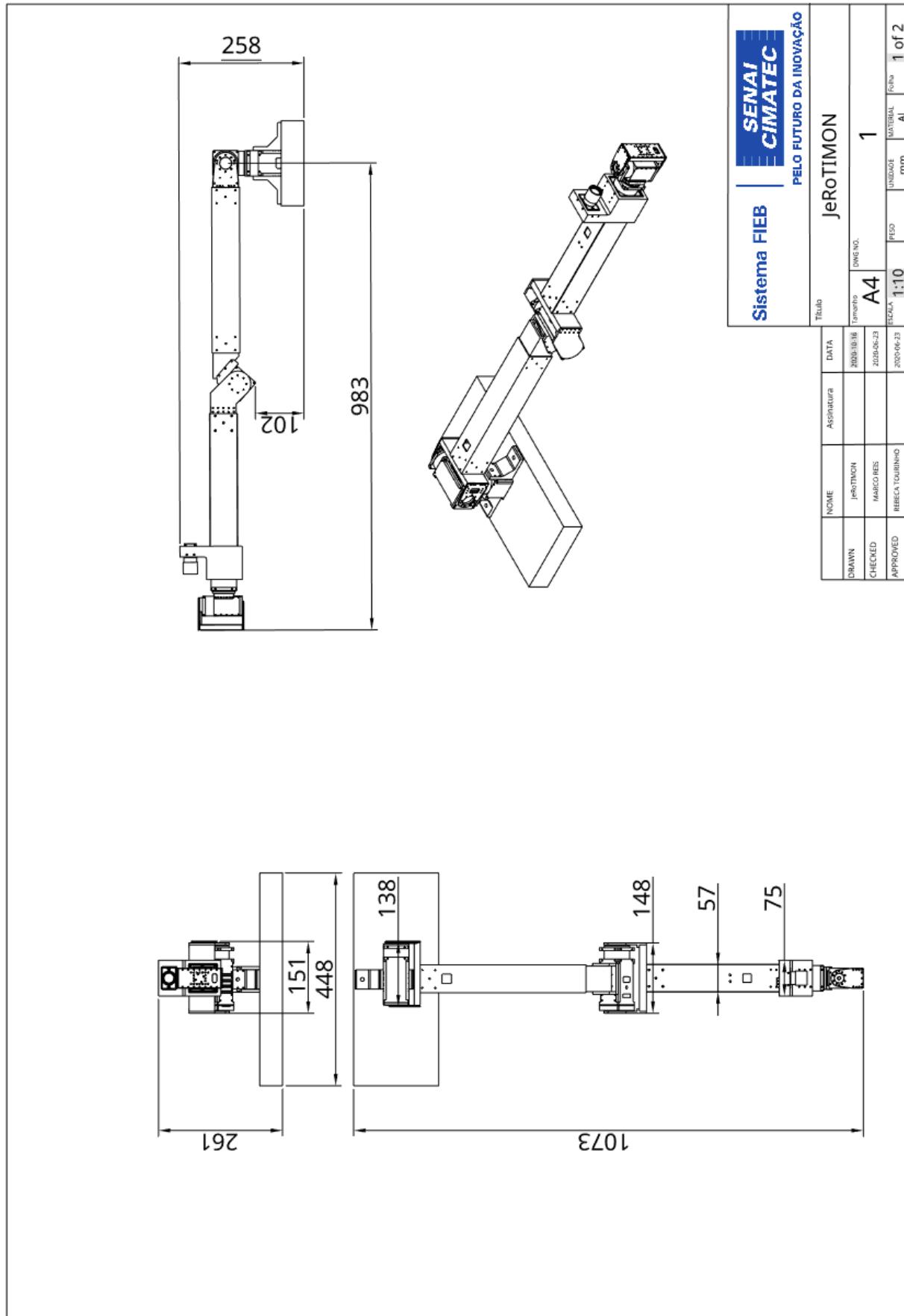
### Projeto mecânico

O projeto mecânico do manipulador JeRoTIMON foi desenvolvido em duas etapas: Modelagem 3D e Cálculos Estáticos. A etapa de modelagem levou em consideração, além da análise estrutural e requisitos do cliente, as restrições físicas para que o manipulador pudesse ser anexado no UGV *Warthog*. Por fim, foi realizado os cálculos de esforços estáticos em que analisou-se os esforços sofridos pelo manipulador, na situação em que exige uma maior carga dos atuadores, e com isso garanta a funcionalidade do mesmo. A seguir, será visto o desenho mecânico final do manipulador e como foi realizado a análise estática.

#### F.1 Desenho mecânico

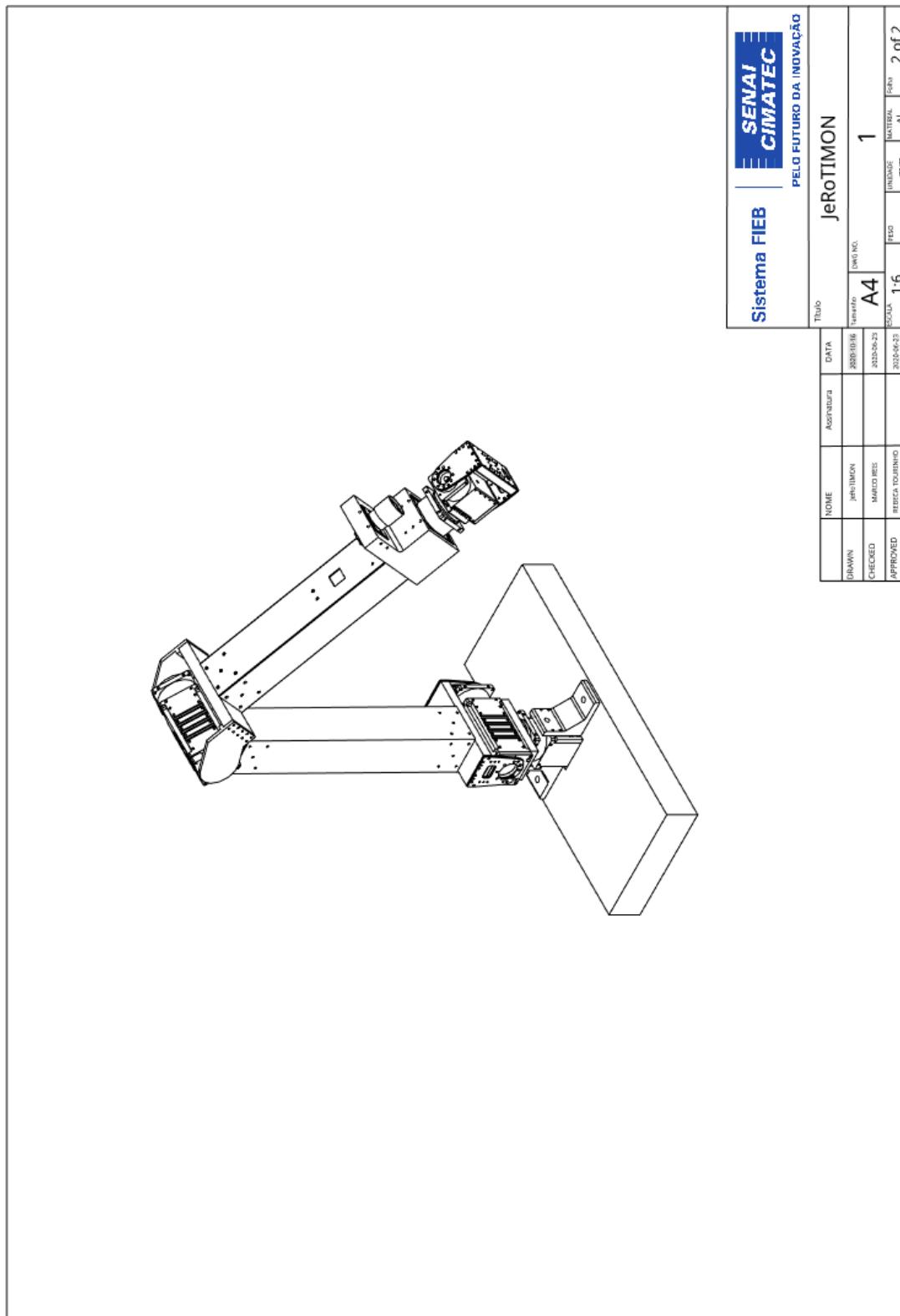
O desenho do manipulador foi todo elaborado utilizando o *software* “*OnShape*”. A Figura 39 mostra o desenho técnico do manipulador com as vistas: frontal, lateral esquerda, superior e isométrica, contendo todos os detalhes necessários para execução do projeto. A figura 40 mostra a vista isométrica expandida do manipulador em sua posição *home*.

Figura 39: Desenho técnico do JeRoTIMON, vistas: frontal, lateral esquerda e superior.



Fonte: Autoria própria.

Figura 40: Desenho técnico do JeRoTIMON, vista isométrica.

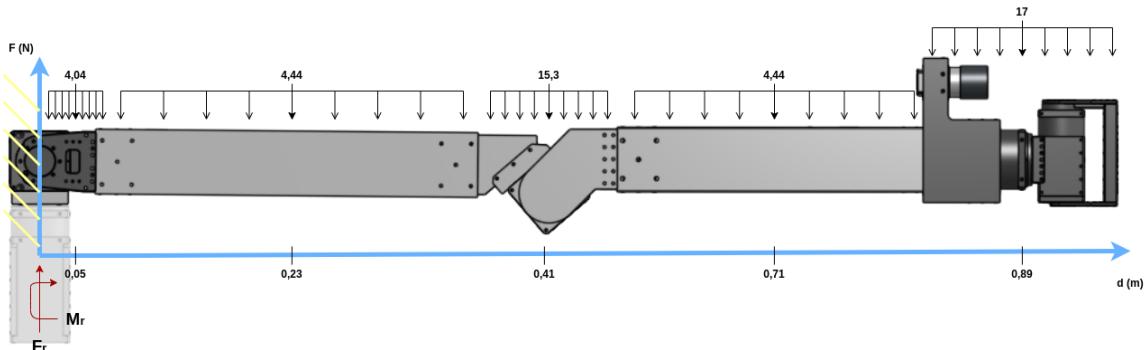


Fonte: Autoria própria.

## F.2 Analise estática

Para calcular a carga máxima suportada pelo manipulador foi realizada uma análise estática. Para isso, o mesmo foi posto na situação em que exige um maior esforço das juntas 0 e 1, conforme visto na Figura 41.

Figura 41: Analise estática - condição de maior esforço exigido.



fonte: Autoria Própria

Para analisar os esforços sofridos pela junta 1 (motor 2), admiti-se a extremidade esquerda como engastada e em sua situação de equilíbrio estático, como visto na Figura 41. A força resultante no eixo y é calculada da seguinte forma:

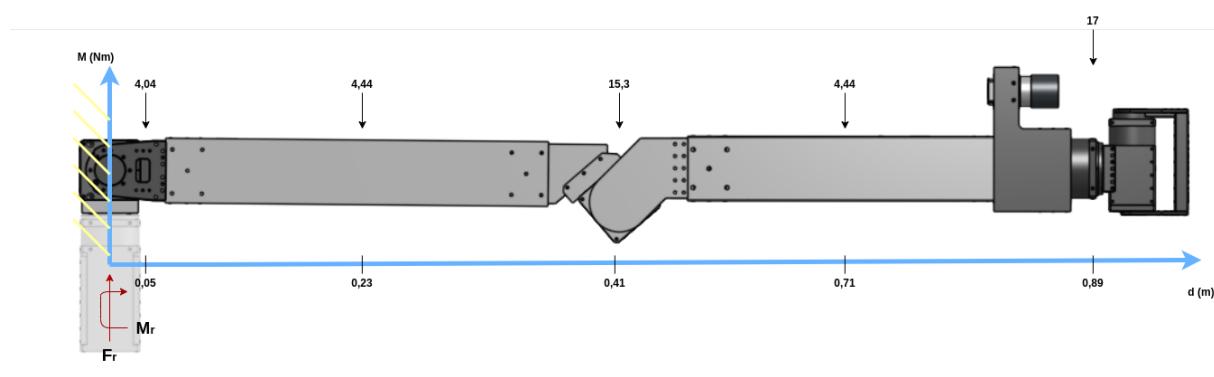
$$\sum F_{ry} = 0$$

$$F_{ry} - 4,04 - 4,44 - 15,3 - 4,44 - 17 = 0$$

$$F_{ry} = 45,22 \text{ [N]}$$

Para calcular o torque sofrido pela junta, as cargas distribuídas de cada componente do robô foram colocadas no seu centróide, conforme visto na Figura 42.

Figura 42: Analise estática - condição de maior esforço exigido (cargas no centróide).



fonte: Autoria Própria

Em seguida, foi calculado o momento torsor gerado nos motores 1 e 2 por cada carga pontual da seguinte maneira:

$$\sum M_r = 0$$

$$M_r - (4,04 \times 0,05) - (4,44 \times 0,23) - (15,3 \times 0,41) - (4,44 \times 0,71) - (17 \times 0,89) = 0$$

$$M_r = \mathbf{25,8} \text{ [N x m]}$$

Por fim, foi calculada a carga máxima suportada pelo manipulador, também conhecida por *payload*. Para este cálculo, foi necessário obter e comparar as informações sobre o torque máximo fornecido pelo motor que compõe a junta 1 (PH54-200-S500-R) que é de 44,7 N x m, segundo o *datasheet* disponível no anexo D. Sendo o alcance máximo de 0,89 m, temos:

$$T_{max/motor} = T_{sofrido} - T_{payload}$$

$$44,7 = 25,8 - (F_{payload} \times 0,89)$$

$$F_{payload} = 21,23[N]$$

De acordo com a Segunda Lei de Newton e considerando a aceleração da gravidade como  $10 \text{ m/s}^2$ , obtemos uma carga máxima suportada pelo manipulador de:

$$F_r = m \times a$$

$$21,23 = m \times 10$$

$$m = \mathbf{2,1} \text{ [kg]}$$



## APÊNDICE G

### Algoritmo de busca e acionamento do painel

```
#include <ros/ros.h>
#include <geometry_msgs/PoseWithCovarianceStamped.h>
#include <moveit/move_group_interface/move_group_interface.h>
#include <moveit/planning_scene_interface/
    planning_scene_interface.h>
#include <moveit_msgs/DisplayRobotState.h>
#include <moveit_msgs/DisplayTrajectory.h>
#include <moveit_msgs/AttachedCollisionObject.h>
#include <moveit_msgs/CollisionObject.h>
#include <moveit_visual_tools/moveit_visual_tools.h>
#include <trajectory_msgs/JointTrajectory.h>
#include <trajectory_msgs/JointTrajectoryPoint.h>
#include <tf/transform_listener.h>
#include <tf/tf.h>
#include <geometry_msgs/Pose.h>
#include <moveit_msgs/PlanningScene.h>
#include <moveit_msgs/ApplyPlanningScene.h>

int main(int argc, char** argv)
{
    ros::init(argc, argv, "push_button_real");
    ros::NodeHandle nh;
    ros::AsyncSpinner spinner(2);
    spinner.start();
    static const std::string PLANNING_GROUP = "arm";
    moveit::planning_interface::MoveGroupInterface move_group(
        PLANNING_GROUP);
    moveit::planning_interface::PlanningSceneInterface
        planning_scene_interface;
    const robot_state::JointModelGroup* joint_model_group =
        move_group.getCurrentState()->getJointModelGroup(
            PLANNING_GROUP);
    namespace rvt = rviz_visual_tools;
    moveit_visual_tools::MoveItVisualTools visual_tools(
        "invisible_link");
```

```
visual_tools.deleteAllMarkers();
visual_tools.loadRemoteControl();
Eigen::Isometry3d text_pose = Eigen::Isometry3d::Identity();
text_pose.translation().z() = 1.75;
visual_tools.publishText(text_pose, "MoveGroupInterface Demo",
    rvt::WHITE, rvt::XLARGE);
visual_tools.trigger();
ROS_INFO_NAMED( "Reference frame: %s", move_group.
    getPlanningFrame().c_str());
ROS_INFO_NAMED( "End effector link: %s", move_group.
    getEndEffectorLink().c_str());

ros::Time start_time = ros::Time::now();

//JUNTAS
moveit::core::RobotStatePtr current_state = move_group.
    getCurrentState();
std::vector<double> joint_group_positions;
current_state->copyJointGroupPositions(joint_model_group,
    joint_group_positions); //COMANDO NO DOMÍNIO DAS JUNTAS
joint_group_positions[0] = 0.0;
joint_group_positions[1] = -0.12;
joint_group_positions[2] = 2.2;
joint_group_positions[3] = 1.57;
joint_group_positions[4] = 0.68;
move_group.setJointValueTarget(joint_group_positions);
moveit::planning_interface::MoveGroupInterface::Plan my_plan;
move_group.plan(my_plan);
while(move_group.plan(my_plan).val == -1){
    move_group.plan(my_plan);
}
//move_group.execute(my_plan)

sleep(5);

// Define variables for xyz for the box
double transform_fake_button_x;
double transform_fake_button_y;
double transform_fake_button_z;
```

```
double transform_fake_button_or_x;
double transform_fake_button_or_y;
double transform_fake_button_or_z;
double transform_fake_button_or_w;
// Define a collision object ROS message.
moveit_msgs::CollisionObject collision_object; //floor
collision_object.header.frame_id = move_group.getPlanningFrame()
();
moveit_msgs::CollisionObject collision_object1; //box
collision_object1.header.frame_id = move_group.
    getPlanningFrame();
// The id of the object is used to identify it.
collision_object.id = "floor";
collision_object1.id = "box";
// Define a box to add to the world.
// floor
shape_msgs::SolidPrimitive primitive;
primitive.type = primitive.BOX;
primitive.dimensions.resize(3);
primitive.dimensions[0] = 2.0;
primitive.dimensions[1] = 3.0;
primitive.dimensions[2] = 0.0;
// Define a pose for the box (specified relative to frame_id)
geometry_msgs::Pose box_pose;
box_pose.orientation.w = 0.0;
box_pose.position.x = 0.0;
box_pose.position.y = 0.0;
box_pose.position.z = 0.0;

tf::TransformListener listener;
//GETTING FAKE BUTTON POSE
tf::StampedTransform transform, transform_button_;
int i=2;
posicao:
try{
std :: cout << i ;
i = i+1;
listener.waitForTransform("/invisible_link", "/fake_botao",

```

```
    ros::Time(0), ros::Duration(2.0));
listener.lookupTransform("/invisible_link", "/fake_botao",
                       ros::Time(0), transform);

transform_fake_button_x = transform.getOrigin().x();
transform_fake_button_y = transform.getOrigin().y();
transform_fake_button_z = transform.getOrigin().z();
transform_fake_button_or_x = transform.getRotation().x();
transform_fake_button_or_y = transform.getRotation().y();
transform_fake_button_or_z = transform.getRotation().z();
transform_fake_button_or_w = transform.getRotation().w();

}

catch (tf::TransformException &ex) {

if(i == 1){
moveit::core::RobotStatePtr current_state = move_group.
    getCurrentState();
std::vector<double> joint_group_positions;
current_state->copyJointGroupPositions(joint_model_group,
    joint_group_positions);
joint_group_positions[0] = 0;
joint_group_positions[1] = -0.006;
joint_group_positions[2] = 2.052;
joint_group_positions[3] = 1.57;
joint_group_positions[4] = 0.68;
move_group.setJointValueTarget(joint_group_positions);
move_group.plan(my_plan);
while(move_group.plan(my_plan).val == -1){
    move_group.plan(my_plan);
}
// move_group.execute(my_plan);
sleep(5.0);
}

if(i == 2){
moveit::core::RobotStatePtr current_state = move_group.
    getCurrentState();
std::vector<double> joint_group_positions;
```

```
current_state->copyJointGroupPositions(joint_model_group ,  
    joint_group_positions);  
joint_group_positions[0] = 0;  
joint_group_positions[1] = -0.4999;  
joint_group_positions[2] = 2.499;  
joint_group_positions[3] = 1.57;  
joint_group_positions[4] = 0.68;  
move_group.setJointValueTarget(joint_group_positions);  
move_group.plan(my_plan);  
while(move_group.plan(my_plan).val == -1){  
    move_group.plan(my_plan);  
}  
// move_group.execute(my_plan);  
sleep(5.0);  
}  
  
if(i == 3){  
moveit::core::RobotStatePtr current_state = move_group.  
    getCurrentState();  
std::vector<double> joint_group_positions;  
current_state->copyJointGroupPositions(joint_model_group ,  
    joint_group_positions);  
joint_group_positions[0] = 0.4;  
joint_group_positions[1] = -0.769;  
joint_group_positions[2] = 2.757;  
joint_group_positions[3] = 1.57;  
joint_group_positions[4] = 0.68;  
move_group.setJointValueTarget(joint_group_positions);  
move_group.plan(my_plan);  
while(move_group.plan(my_plan).val == -1){  
    move_group.plan(my_plan);  
}  
// move_group.execute(my_plan);  
sleep(5.0);  
}  
  
if(i == 4){  
moveit::core::RobotStatePtr current_state = move_group.  
    getCurrentState();
```

```
std :: vector<double> joint_group_positions;
current_state->copyJointGroupPositions(joint_model_group ,
    joint_group_positions);
joint_group_positions[0] = -0.4;
joint_group_positions[1] = -0.769;
joint_group_positions[2] = 2.757;
joint_group_positions[3] = 1.57;
joint_group_positions[4] = 0.68;
move_group.setJointValueTarget(joint_group_positions);
move_group.plan(my_plan);
while(move_group.plan(my_plan).val == -1){
    move_group.plan(my_plan);
}
// move_group.execute(my_plan);
i=0;
sleep(5.0);
}

goto posicao;

}

//GETTING BUTTON POSE
try{
    // ros::Time now = ros::Time::now();
    listener.waitForTransform("/invisible_link", "/botao_press",
        ros::Time(0), ros::Duration(2.0));
    listener.lookupTransform("/invisible_link", "/botao_press",
        ros::Time(0), transform_button_);
    std::cout<<"Pose do botão: x: "<< transform_button_.getOrigin
    () .x()<<
        " y: "<< transform_button_.getOrigin
    () .y()<<
        " z: "<< transform_button_.getOrigin
    () .z();

}

catch ( tf::TransformException &ex) {
i=0;
goto posicao;
```

```
}

ROS_INFO( "Aruco Detectado" );
ros::Duration delta_t1 = ros::Time::now() - start_time;
double delta_t1_sec = delta_t1.toSec();

std::cout << "Tempo de busca: " << delta_t1_sec;

if ( transform.getRotation().x() <= 0.5) {
    ROS_INFO( "Caixa na Horizontal" );
    // box
    shape_msgs::SolidPrimitive primitive1;
    primitive1.type = primitive.BOX;
    primitive1.dimensions.resize(3);
    primitive1.dimensions[0] = 0.2;
    primitive1.dimensions[1] = 0.3;
    primitive1.dimensions[2] = 0.2;
    // Define a pose for the box (specified relative to frame_id
    )
    geometry_msgs::Pose box_pose1;
    box_pose1.position.x = transform_fake_button_x ;
    box_pose1.position.y = transform_fake_button_y;
    box_pose1.position.z = transform_fake_button_z-0.354;
    box_pose1.orientation.w = transform_fake_button_or_w;
    box_pose1.orientation.x = transform_fake_button_or_x;
    box_pose1.orientation.y = transform_fake_button_or_y;
    box_pose1.orientation.z = transform_fake_button_or_z;
    //Collision box
    collision_object.primitives.push_back(primitive);
    collision_object.primitive_poses.push_back(box_pose);
    collision_object.operation = collision_object.ADD;
    collision_object1.primitives.push_back(primitive1);
    collision_object1.primitive_poses.push_back(box_pose1);
    collision_object1.operation = collision_object.ADD;

    std::vector<moveit_msgs::CollisionObject> collision_objects;
    collision_objects.push_back(collision_object);
    collision_objects.push_back(collision_object1);
```

```
planning_scene_interface.addCollisionObjects(  
    collision_objects);  
  
geometry_msgs::Pose target_pose;  
target_pose.orientation.w = transform.getRotation().w();  
target_pose.orientation.x = transform.getRotation().x();  
target_pose.orientation.y = transform.getRotation().y();  
target_pose.orientation.z = transform.getRotation().z();  
target_pose.position.x = transform.getOrigin().x();  
target_pose.position.y = transform.getOrigin().y();  
target_pose.position.z = transform.getOrigin().z();  
move_group.setGoalPositionTolerance(0.001);  
move_group.setGoalOrientationTolerance(0.35);  
move_group.setPlanningTime(25);  
move_group.setPoseTarget(target_pose);  
  
move_group.plan(my_plan);  
while (move_group.plan(my_plan).val != 1){  
    move_group.plan(my_plan);  
}  
// move_group.execute(my_plan);  
  
sleep(16.0);  
  
moveit::core::RobotStatePtr current_state_1 = move_group.  
    getCurrentState();  
std::vector<double> joint_group_positions_garra;  
current_state_1->copyJointGroupPositions(joint_model_group,  
    joint_group_positions_garra);  
joint_group_positions_garra[3] = 1.57;  
joint_group_positions_garra[4] = 1.21;  
move_group.setJointValueTarget(joint_group_positions_garra);  
moveit::planning_interface::MoveGroupInterface::Plan my_plan  
;  
move_group.plan(my_plan);  
while(move_group.plan(my_plan).val == -1){  
    move_group.plan(my_plan);
```

```
}

sleep(11.0);
// move_group.execute(my_plan);

//PRESS BUTTON

target_pose.position.z -=0.175;
std::vector<geometry_msgs::Pose> waypoints;
waypoints.push_back(target_pose);
moveit_msgs::RobotTrajectory trajectory;
const double jump_threshold = 0.0;
const double eef_step = 0.1;
move_group.computeCartesianPath(waypoints, eef_step,
                               jump_threshold, trajectory);
sleep(5.0);

geometry_msgs::Pose pose_real;
pose_real = move_group.getCurrentPose().pose;
std::cout << "Pose Real: " << pose_real.position;

target_pose.position.z +=0.11;
move_group.setPoseTarget(target_pose);
move_group.plan(my_plan);
while (move_group.plan(my_plan).val != 1){
    move_group.plan(my_plan);
}

else
{ // box
    shape_msgs::SolidPrimitive primitive1;
    primitive1.type = primitive.BOX;
    primitive1.dimensions.resize(3);
    primitive1.dimensions[0] = 0.2;
    primitive1.dimensions[1] = 0.3;
    primitive1.dimensions[2] = 0.2;
    // Define a pose for the box (specified relative to frame_id
}
```

```
geometry_msgs::Pose box_pose1;
    box_pose1.orientation.w = transform_fake_button_or_w;
    box_pose1.orientation.x = transform_fake_button_or_x;
    box_pose1.orientation.y = transform_fake_button_or_y;
    box_pose1.orientation.z = transform_fake_button_or_z;
    box_pose1.position.x = transform_fake_button_x -0.1;
    box_pose1.position.y = transform_fake_button_y+0.318;
    box_pose1.position.z = transform_fake_button_z - 0.1;
//Collision box
collision_object.primitives.push_back(primitive);
collision_object.primitive_poses.push_back(box_pose);
collision_object.operation = collision_object.ADD;
collision_object1.primitives.push_back(primitive1);
collision_object1.primitive_poses.push_back(box_pose1);
collision_object1.operation = collision_object.ADD;

std::vector<moveit_msgs::CollisionObject> collision_objects;
collision_objects.push_back(collision_object);
collision_objects.push_back(collision_object1);

planning_scene_interface.addCollisionObjects(
    collision_objects);

geometry_msgs::Pose target_pose;

target_pose.orientation.w = transform.getRotation().w();
target_pose.orientation.x = transform.getRotation().x();
target_pose.orientation.y = transform.getRotation().y();
target_pose.orientation.z = transform.getRotation().z();
target_pose.position.x = transform.getOrigin().x();
target_pose.position.y = transform.getOrigin().y();
target_pose.position.z = transform.getOrigin().z();
move_group.setGoalPositionTolerance(0.001);
move_group.setGoalOrientationTolerance(1.0);
move_group.setPlanningTime(25);
move_group.setPoseTarget(target_pose);
move_group.plan(my_plan);
while (move_group.plan(my_plan).val != 1){
```

```
move_group.plan(my_plan);
}
// move_group.execute(my_plan);
sleep(10.0);
moveit::core::RobotStatePtr current_state_1 = move_group.
    getCurrentState();
std::vector<double> joint_group_positions_garra;
current_state_1->copyJointGroupPositions(joint_model_group,
    joint_group_positions_garra);
joint_group_positions_garra[3] = -2.12;
joint_group_positions_garra[4] = 1.21;
move_group.setJointValueTarget(joint_group_positions_garra);
moveit::planning_interface::MoveGroupInterface::Plan my_plan
    ;
move_group.plan(my_plan);
while(move_group.plan(my_plan).val == -1){
    move_group.plan(my_plan);
}
sleep(10.0);

//PRESS BUTTON
move_group.setGoalOrientationTolerance(0.9);
float or_w, or_x, or_y, or_z, x, y;
or_x = move_group.getCurrentPose().pose.orientation.x;
or_y = move_group.getCurrentPose().pose.orientation.y;
or_z = move_group.getCurrentPose().pose.orientation.z;
or_w = move_group.getCurrentPose().pose.orientation.w;
target_pose.orientation.w = or_w;
target_pose.orientation.x = or_x;
target_pose.orientation.y = or_y;
target_pose.orientation.z = or_z;
target_pose.position.y += 0.172;
target_pose.position.z += 0.018;
move_group.setPoseTarget(target_pose);
// moveit::planning_interface::MoveGroupInterface::Plan
my_plan4;
move_group.plan(my_plan);
while (move_group.plan(my_plan).val != 1){
    move_group.plan(my_plan);
```

```
    }

    sleep(10.0);

    target_pose.position.y -=0.2;
    move_group.setPoseTarget(target_pose);
    move_group.plan(my_plan);
    while (move_group.plan(my_plan).val != 1){
        move_group.plan(my_plan);
    }
    // move_group.execute(my_plan);

}

//BACK TO HOME POSITION

current_state->copyJointGroupPositions(joint_model_group,
    joint_group_positions);
joint_group_positions[0] = 0.0;
joint_group_positions[1] = -0.12;
joint_group_positions[2] = 2.2;
joint_group_positions[3] = 0;
joint_group_positions[4] = 0.0;
move_group.setJointValueTarget(joint_group_positions);

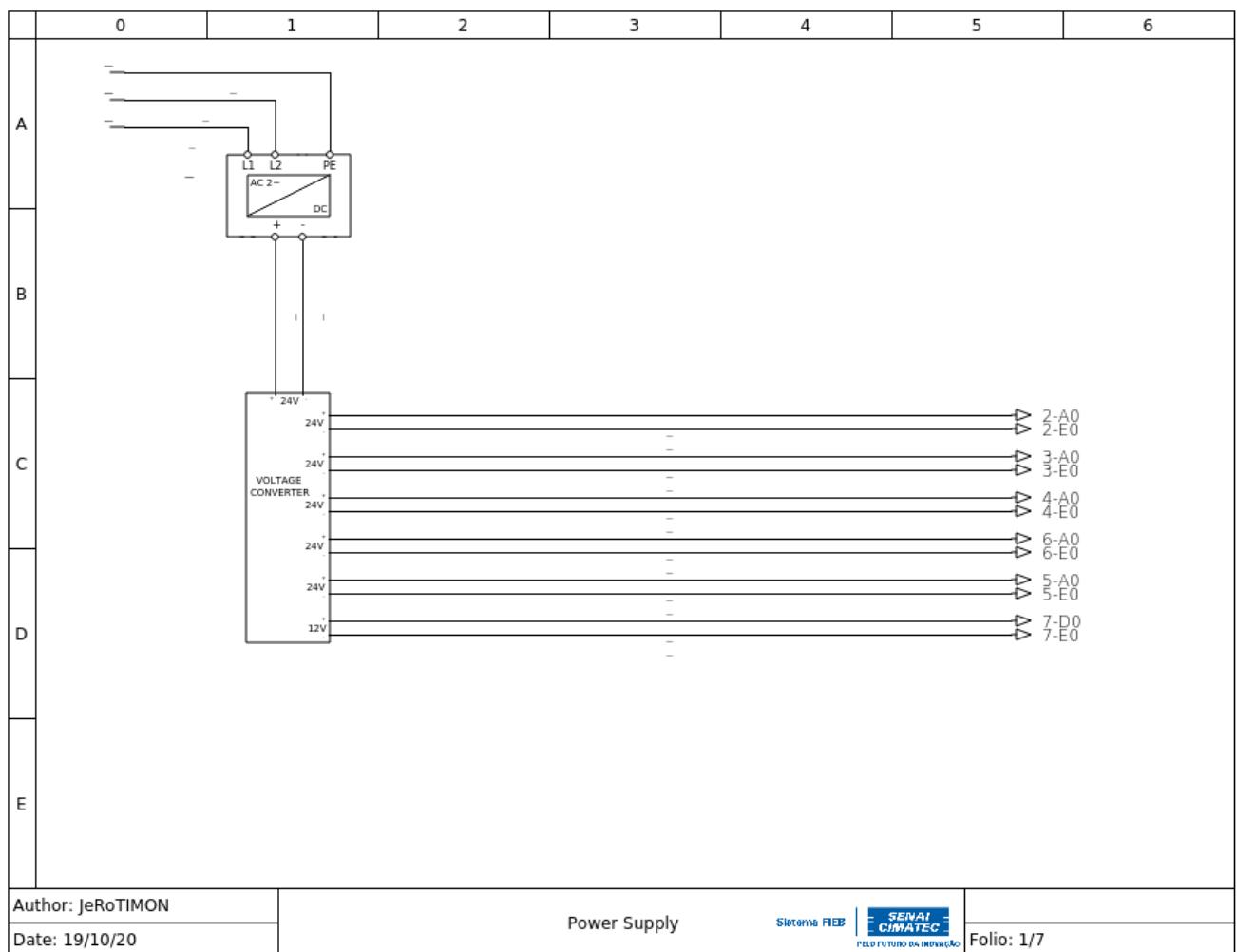
move_group.plan(my_plan);
while (move_group.plan(my_plan).val != 1){
    move_group.plan(my_plan);
}
// move_group.execute(my_plan);
sleep(5.0);
ros::Duration delta_t = ros::Time::now() - start_time;
double delta_t_sec = delta_t.toSec();

std::cout<< "Tempo de missão: " << delta_t_sec;

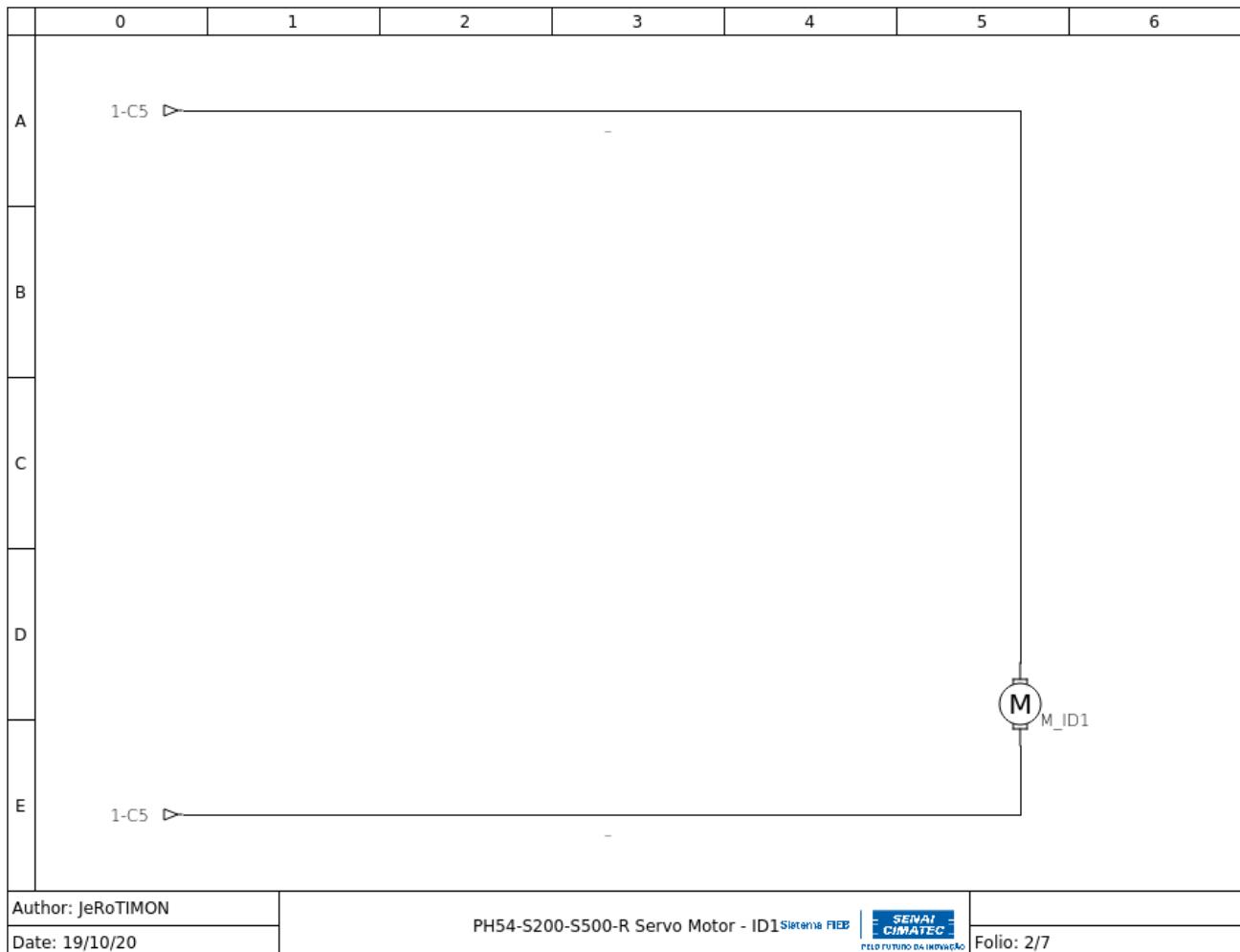
ros::shutdown();
return 0;
}
```

## APÊNDICE H

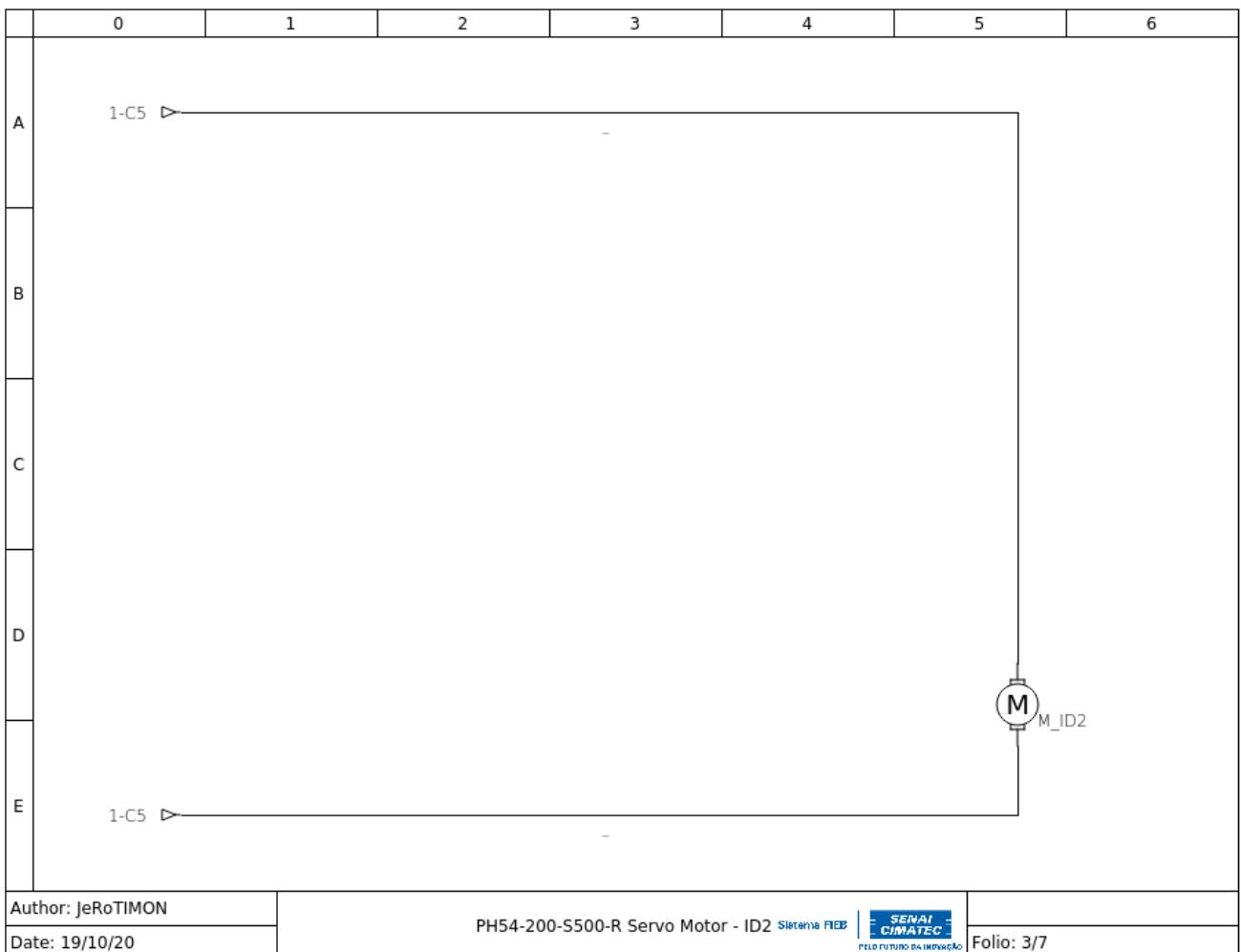
### Diagrama elétrico do JeRoTIMON



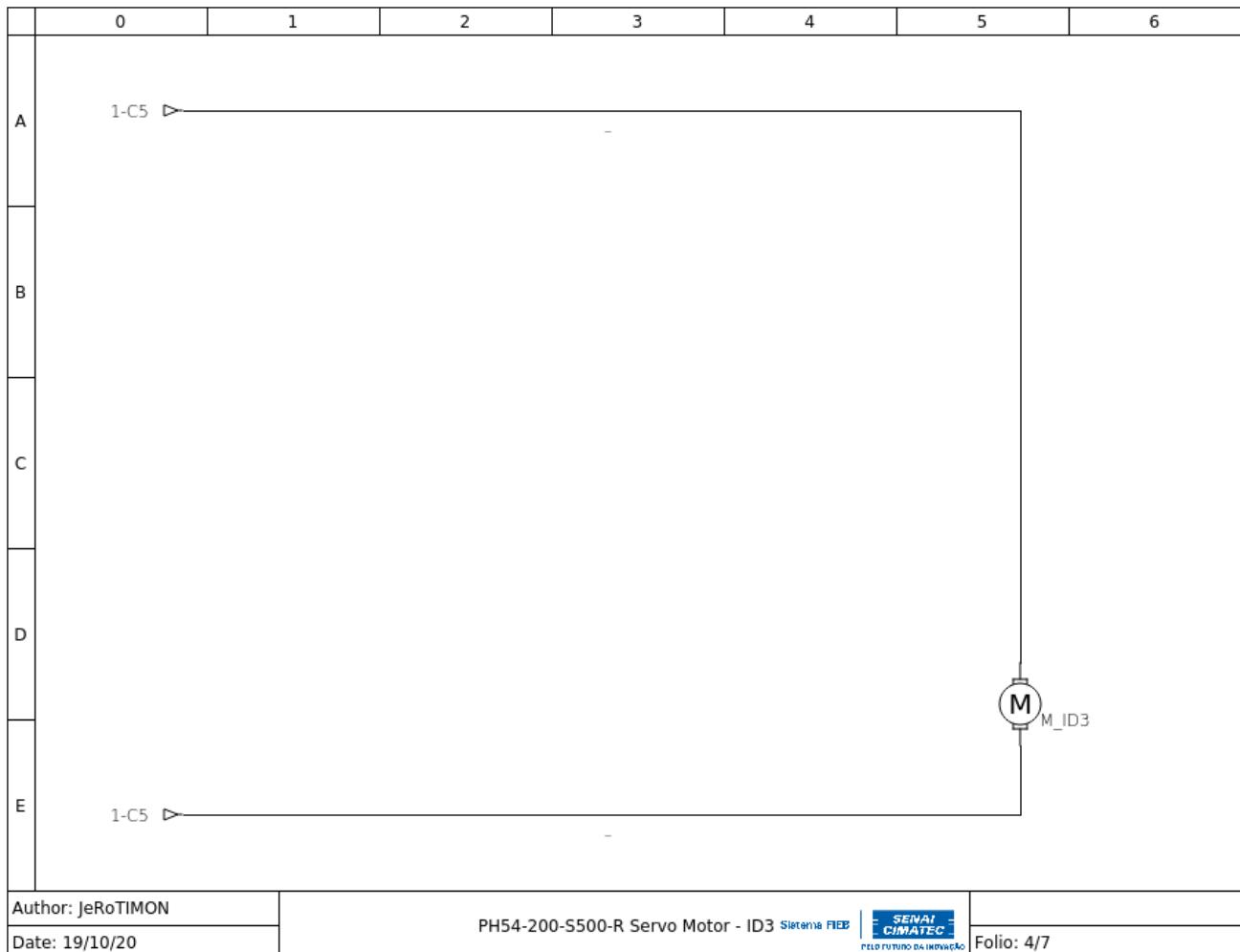
## Apêndice B



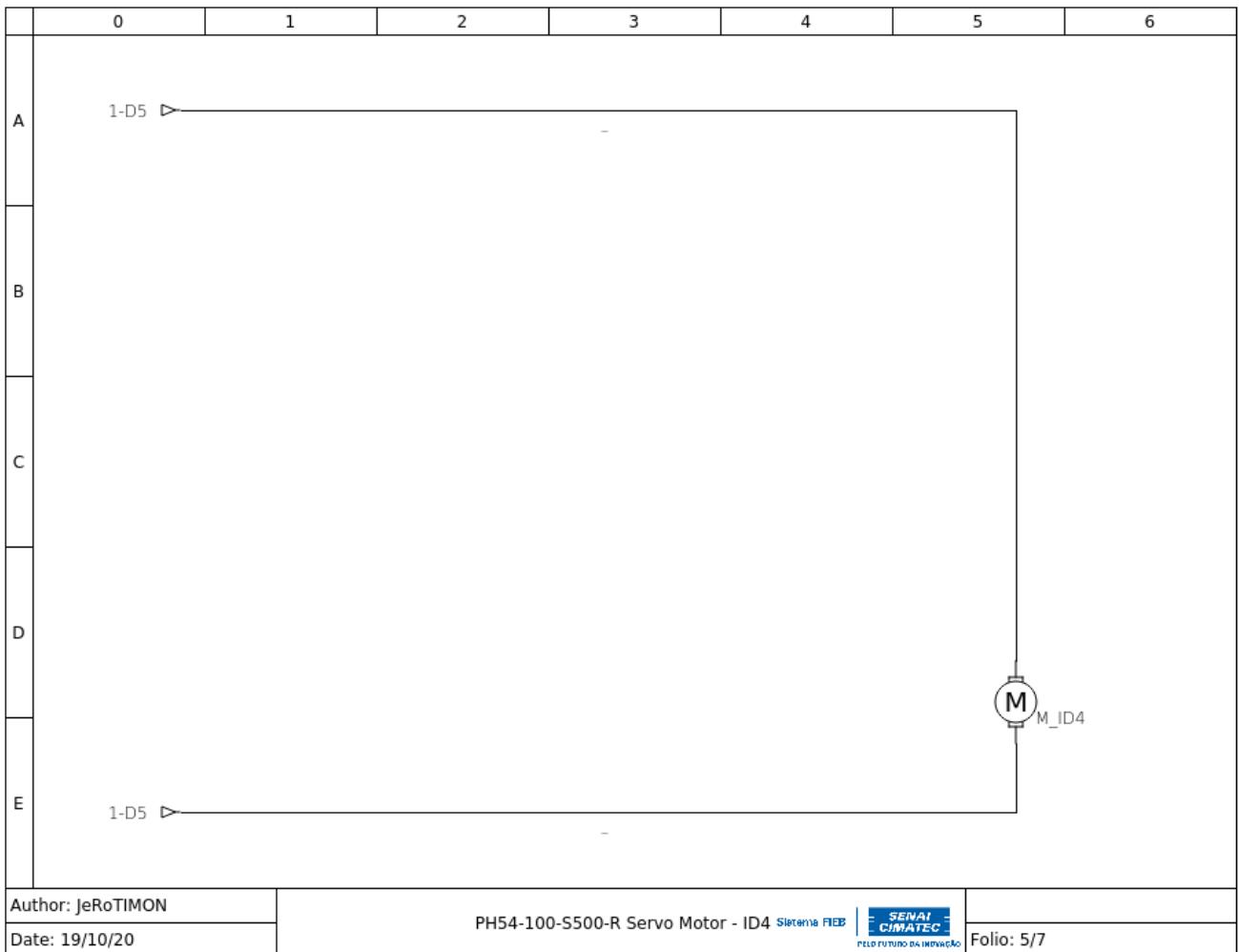
## Apêndice B



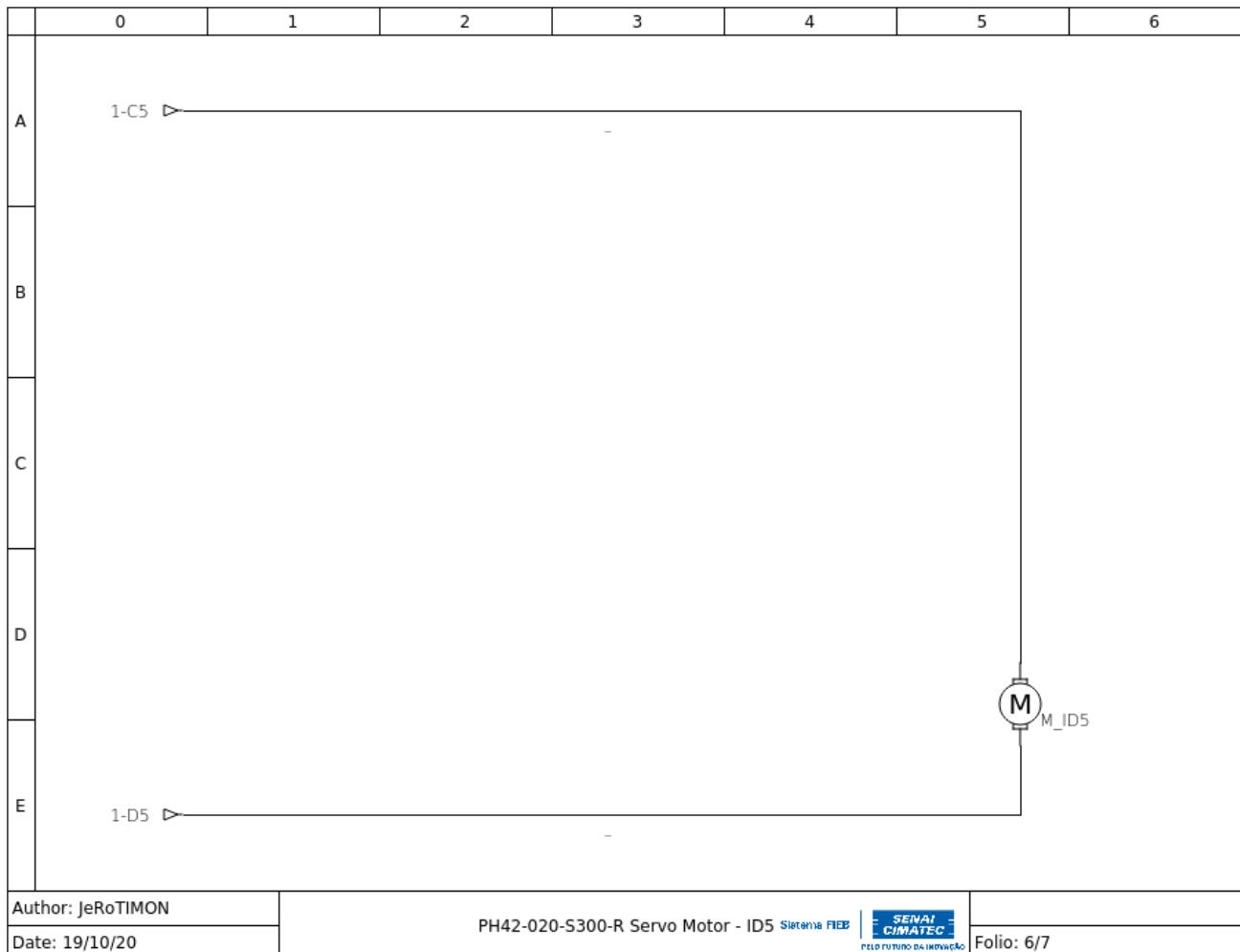
## Apêndice B



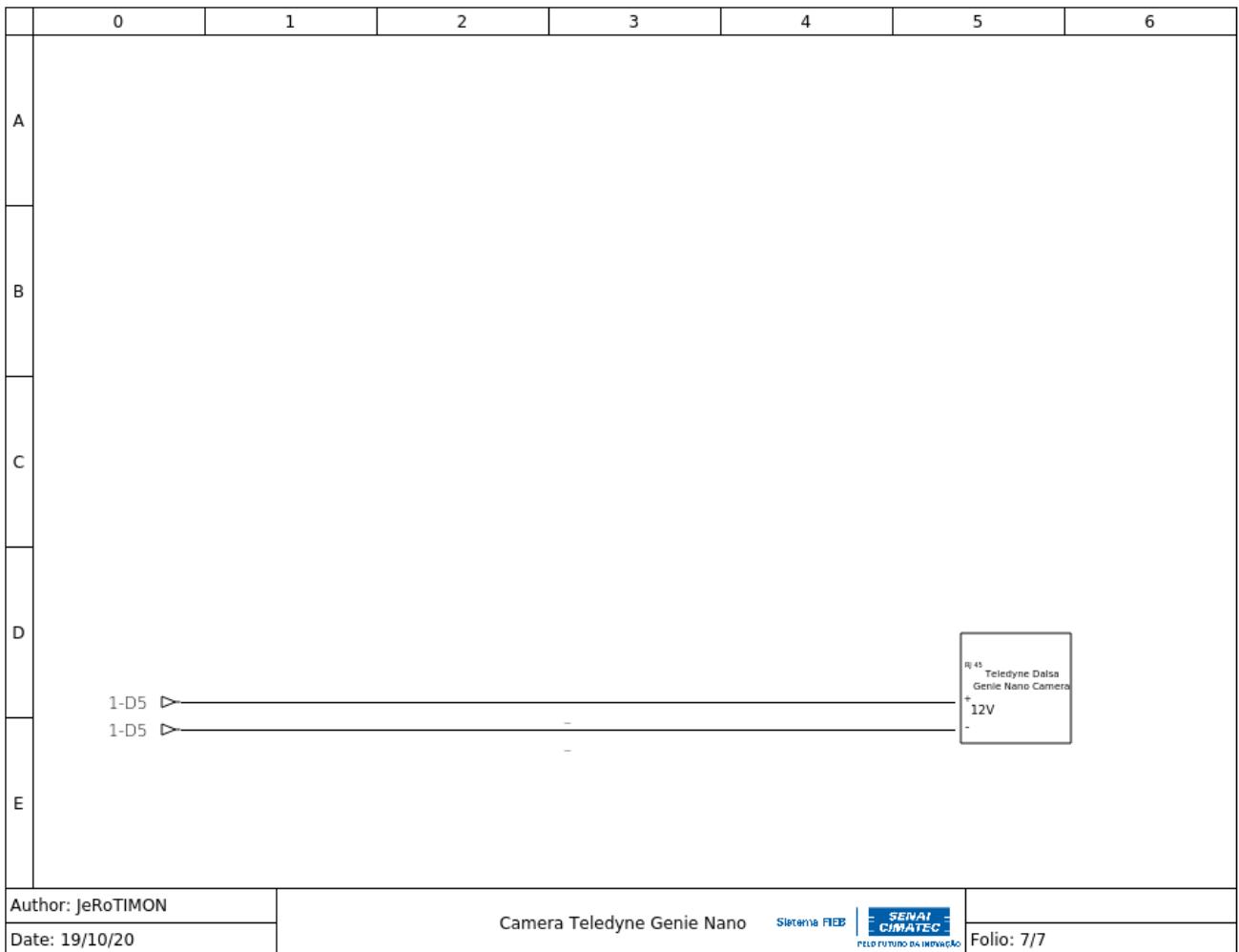
## Apêndice B



## Apêndice B



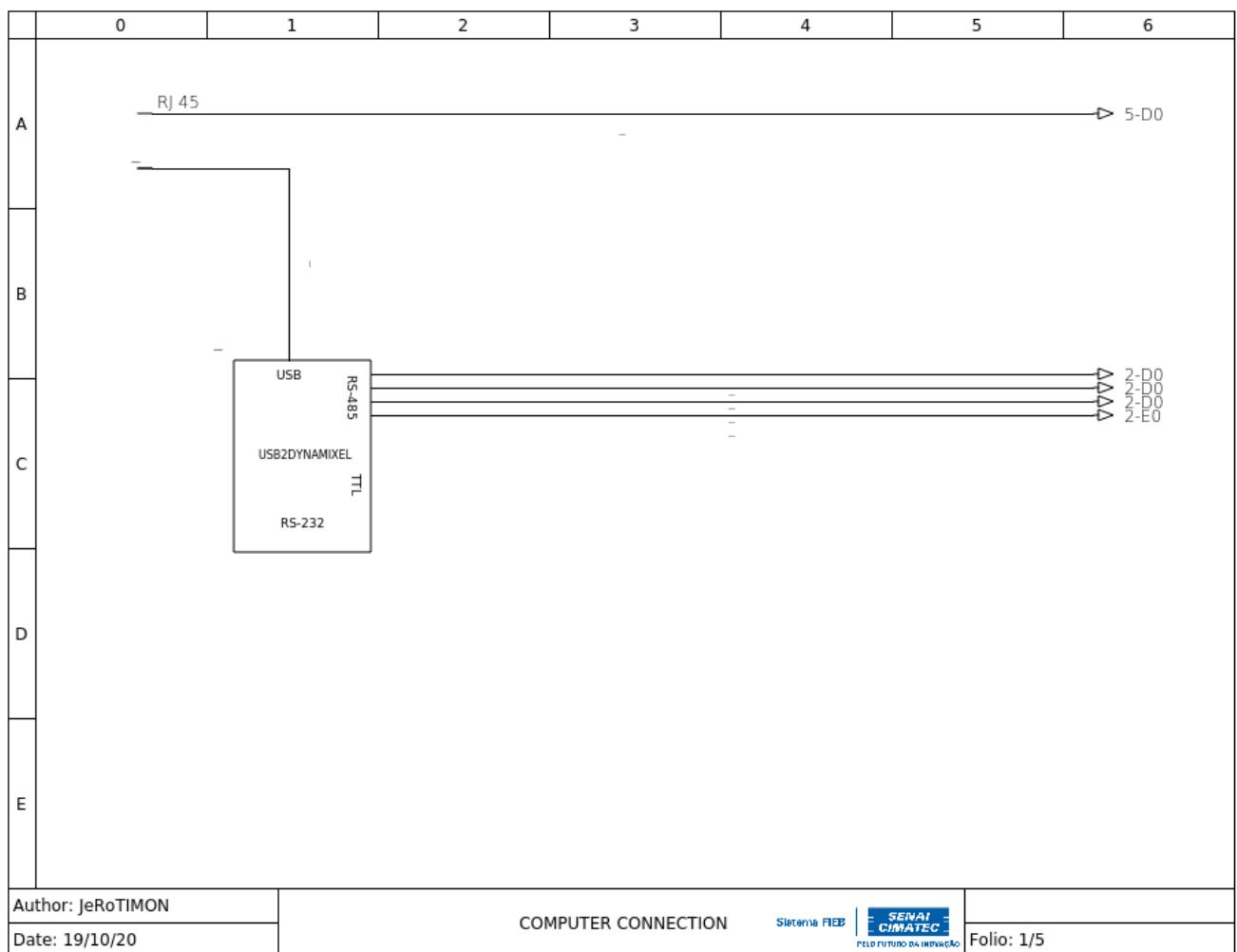
## Apêndice B



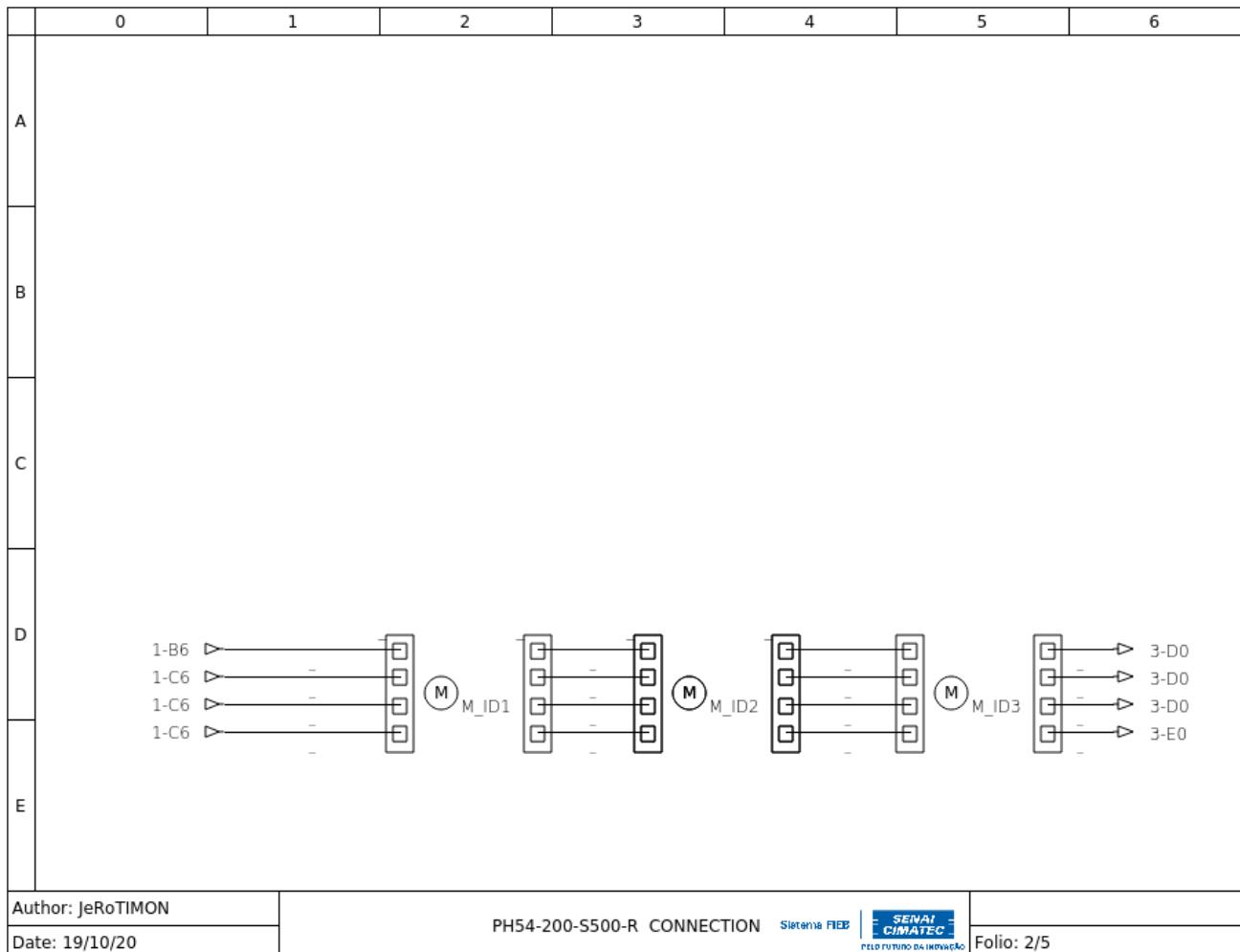


## APÊNDICE I

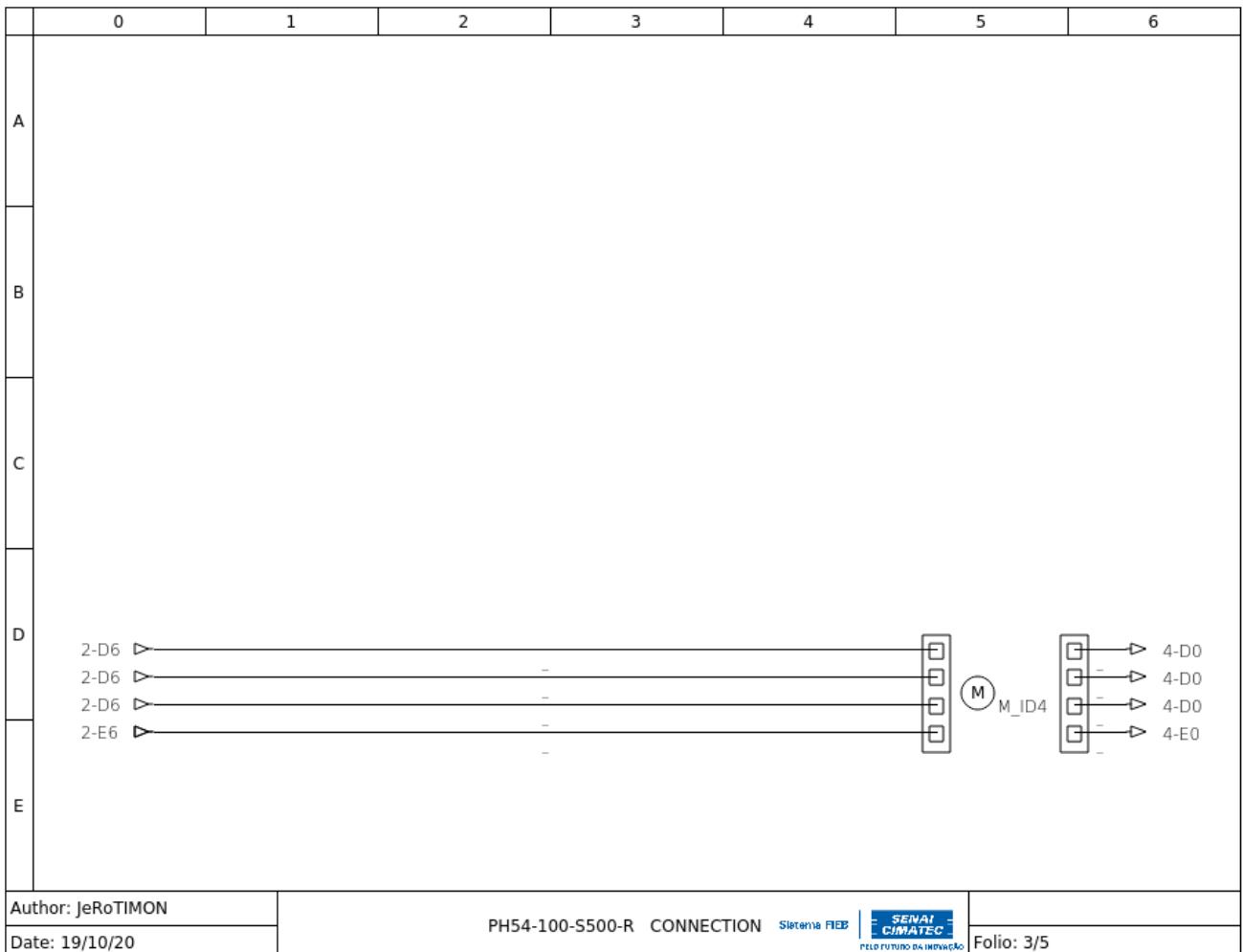
### Diagrama de conexão do JeRoTIMON



Apêndice B



## Apêndice B



## Apêndice B

	0	1	2	3	4	5	6
A							
B							
C							
D	3-D6 ►						
	3-D6 ►	-					
	3-D6 ►	-					
	3-E6 ►	-					
E							

Author: JeRoTIMON  
Date: 19/10/20

PH42-020-S300-R CONNECTION Sistema FIEB | SENAI CIMATEC  
FELIZ FUTURO DA INovação

Folio: 4/5

## Apêndice B

---

	0	1	2	3	4	5	6
A							
B							
C							
D	1-A6	►		—			
E							
Author: JeRoTIMON		CAMERA CONNECTION			Sistema FIEP   SENAI CIMATEC PELO FUTURO DA INOVAÇÃO		Folio: 5/5
Date: 19/10/20							



## ANEXO A

### Especificações da câmera Dalsa Genie Nano

## Specifications: M2590, M2590-NIR, C2590

Supported Features	M2590, M2590-NIR	Nano-C2590	
Resolution	2592 x 2048		
Sensor	OnSemi Python5000 P1 (5.1M)		
Pixel Size	4.8 µm x 4.8 µm		
Shutter type	Full frame electronic global shutter function		
Full Well charge	10ke (max)		
Firmware option (Field programmable)	Standard Design Monochrome (factory default)	Standard Design Bayer (factory default)	RGB-Output Design
Max. Internal Frame Rate Full Resolution (2592 x 2048)	51.8 fps (Fast Readout Enable) 24.7 fps (Normal Readout Enable)		
Maximum Sustained Frame Rate Output (with TurboDrive v1)*	42.7 fps (8-bit) 24.9 fps (10-bit)		N/A
Maximum Sustained Frame Rate Output (without TurboDrive)	22 fps (8-bit)		5.5 fps (RGBA) 8.7 fps (RGB) 11 fps (Yuv422) 22 fps (8-bit mono)
Pixel Data Formats	Mono 8-bit Mono 10-bit	Bayer 8-Bit Bayer 10-Bit	RGBA 32-bit RGB 24-bit Yuv422 16-bit Mono 8-bit
Trigger to Exposure Minimum delay (Synchronous Exposure Alignment)	8 µs if exposureAlignment = Synchronous With No Overlap between the new exposure and the previous readout  26.2 µs if exposureAlignment = Synchronous With Overlap between the new exposure and the previous readout		
Trigger to Exposure Minimum delay (Reset Exposure Alignment)	3 µs		
Trigger to Exposure Start jitter (best case with Synchronous Exposure Alignment)	Up to 1 line time		
Trigger to Exposure Start jitter (Reset Exposure Alignment) †	0 µs		
Exposure Time Minimum (see "exposureTimeActual" in <a href="#">Sensor Control</a> )	87 µs (increment steps of 1µs)		
Min. Time from End of Exposure to Start of Next Exposure (second frame)	49 µs – Normal Readout 47 µs – Fast Readout		
Horizontal Line Time:	11.33 µs – Normal Readout 9.33 µs – Fast Readout		
Readout Time	23242 µs – <b>Normal Readout</b> for 2592 x 2048 Add 76µs when overlapping Exposure and Readout 19142 µs – <b>Fast Readout</b> for 2592 x 2048 Add 64µs when overlapping Exposure and Readout <i>Specifically: ( Horizontal line time at current resolution * number of lines ) + ( 3 * ( line time of the 2590 model ) )</i>		
Auto-Brightness	Yes , with Auto-Exposure and AGC (FPGA Gain)		
Black offset control	Yes (in DN)		

Gain Control	In-sensor Analog Gain (1.0x to 8x) in 11 gain steps (1.0, 1.14, 1.33, 1.6, 2.0, 2.29, 2.67, 3.2, 4.0, 5.33, 8.0) In-sensor Digital Gain (1x to 32x) in 0.01x steps In-FPGA Digital Gain (1x to 4x) in 0.007x steps	
Binning Support	Yes In-FPGA (summing and average, 2x2, 4x4 ) Yes In- Sensor (averaging 2x2)	No
Color Correction Support	No	Yes
Decimation Support	No	
Defective Pixel Replacement	Yes, up to 512 positions	
Image Correction	No	
Image Flip Support	Yes, In-Sensor, Vertical Only	
Multi-ROI Support	Yes, in Sensor, up to 16 ROI (mutually exclusive with binning)	
On-Board Image Memory	90MB	
Output Dynamic Range (dB)	62.1 dB (in 10-Bit Pixel Format)	
SNR (dB)	39.8 dB (in 10-Bit Pixel Format)	

\*TurboDrive internal limitation of 250MB/sec

† Note: The actual internal minimum exposure may be different than what is programmed. Use the feature "exposureTimeActual" from the [Sensor Control](#) category to read back the actual sensor exposure. The exposure start sensor event is delayed 4 µs from the actual start.

## Firmware Files for Models 1280, 1930, 2590

The latest firmware files for all Nano models are available on the Teledyne DALSA support web site:  
<http://www.teledynedalsa.com/imaging/support/downloads/firmware/>

The firmware files for these models are listed below. The xx denotes the build number.

### M1280, M1930, M2590

- Standard  
"Genie\_Nano\_OnSemi\_Python\_0.3M-0.5M-1.3M-2M-5M\_Mono\_STD\_Firmware\_5CA18.xx.cbf"

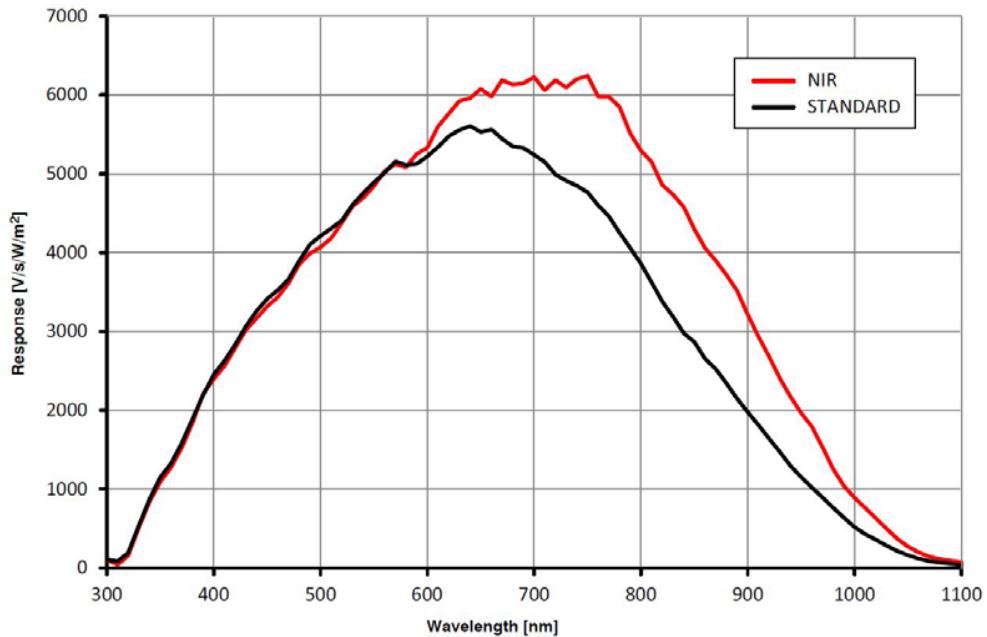
### C1280, C1930, C2590

- Bayer Output  
"Genie\_Nano\_OnSemi\_Python\_0.3M-0.5M-1.3M-2M-5M\_Bayer\_STD\_Firmware\_6CA18.xx.cbf"
- RGB Output  
"Genie\_Nano\_OnSemi\_Python\_0.3M-0.5M-1.3M-2M-5M\_RGB\_Output\_Firmware\_6CA18.xx.cbf"

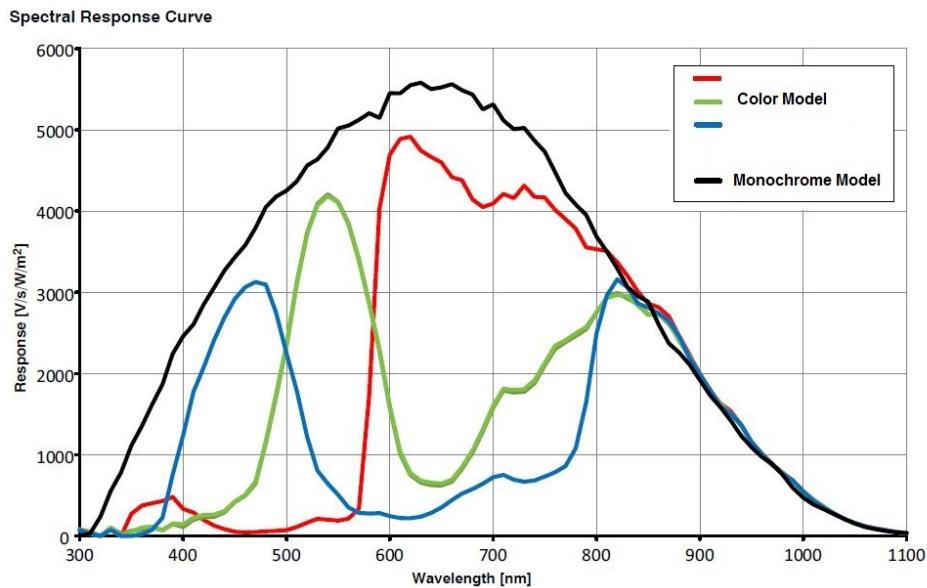
## Spectral Response (Python 4.8 µm series)

Model specific specifications and response graphics for the On-Semi Python (VGA to 5M) series are provided here. The response curves describe the sensor, excluding lens and light source characteristics.

### On-Semi Python Series (with 4.8 µm pixels) — Monochrome and NIR



### On-Semi Python Series (with 4.8 µm pixels) — Monochrome and Color



## ANEXO B

### Especificações do motor *Dynamixel PH42-020-S300-R*

Item	Especificação
Microcontrolador	ARM CORTEX-M4 (168 [MHz], 32Bit)
Motor	Coreless (Maxon)
Taxa de transmissão	9,600 [bps] ~10.5 [Mbps]
Modos de operação	Torque Control Mode
	Velocity Control Mode
	Position Control Mode
	Extended Position Control Mode
	PWM Control Mode (Voltage Control Mode)
Massa	340 [g]
Dimensões (L x A x P)	42 x 84 x 42 [mm]
Resolução	607,500 [pulsos/rev]
Relação de transmissão	303.75:1
Folga	<6 [arcmin], 0.1 [°]
Carga radial	280 [N] (10 [mm] afastado da face do eixo)
Carga axial	100 [N]
Velocidade sem carga	32.7 [rev/min]
Corrente sem carga	0.57 [A]
Velocidade contínua	29.2 [rev/min]
Torque contínuo	5.1 [Nm]
Corrente contínua	1.5 [A]
Potência de saída	20 [W]
Temperatura de operação	5 ~55 [°C]
Tensão de operação	24.0 [V]
Sinal de comando	Pacotes digitais
Protocolo de comunicação	RS485 Comunicação serial asíncrona (8bit, 1 bit de parada, sem paridade)
Conexão física	RS485 Barramento multiponto
ID	253 ID (0 ~252)
Corrente de espera	80 [mA]

Tabela 24: Especificações do motor *Dynamixel HP42-020-S300-R*

Enter Search Terms



DYNAMIXEL

PLATFORM

STEAM

SOFTWARE

PARTS

FAQ

Apêndice B

**PH42-020-S300-R**

- 1. Specifications >
- 2. Control Table >
- 3. How to Assemble >
- 4. Maintenance >
- 5. Reference >

[Edit on GitHub](#)

ROBOTIS



Robot Source



GitHub



TOP



PH42-020-S300-R

**1. Specifications**

Item	Specifications
MCU	ARM CORTEX-M4 (168 [MHz], 32Bit)
Motor	Coreless (Maxon)
Baud Rate	9,600 [bps] ~ 10.5 [Mbps]
Operating Modes	Torque Control Mode Velocity Control Mode Position Control Mode Extended Position Control Mode PWM Control Mode(Voltage Control Mode)
Weight	340 [g]
Dimensions (W x H x D)	42 x 84 x 42 [mm]
Resolution	607,500 [pulse/rev]
Gear Ratio	303.75:1
Backlash	< 6 [arcmin], 0.1 [°]
Radial Load	280 [N] (10 [mm] away from the horn)
Axial Load	100 [N]
No Load Speed	32.7 [rev/min]
No Load Current	0.57 [A]
Continuous Speed	29.2 [rev/min]
Continuous Torque	5.1 [Nm]
Continuous Current	1.5 [A]
Output	20 [W]
Operating Temperature	-5 ~ 55 [°C]
Input Voltage	24.0 [V]
Command Signal	Digital Packet
Protocol Type	RS485 Asynchronous Serial Communication (8bit, 1stop, No Parity)
Physical Connection	RS485 Multidrop Bus

Apêndice B

**PH42-020-S300-R**

- 1. Specifications
- 2. Control Table
- 3. How to Assemble
- 4. Maintenance
- 5. Reference

Item	Specifications
ID	253 ID (0 ~ 252)
Standby Current	30 [mA]

These specifications are calculated based on the specifications of the core motor. Please consult ROBOTIS for the long term use or special use, or else refer to the Performance Graph for general use.



TOP

**DANGER**

(May cause serious injury or death)

- Never place items containing water, flammables, and solvents near product.
- Never place fingers, arms, toes, and other body parts near product during operation.
- Cut power off if product emits strange odors or smoke.
- Keep product out of reach of children.
- Check the power's polarity before wiring.

**CAUTION**

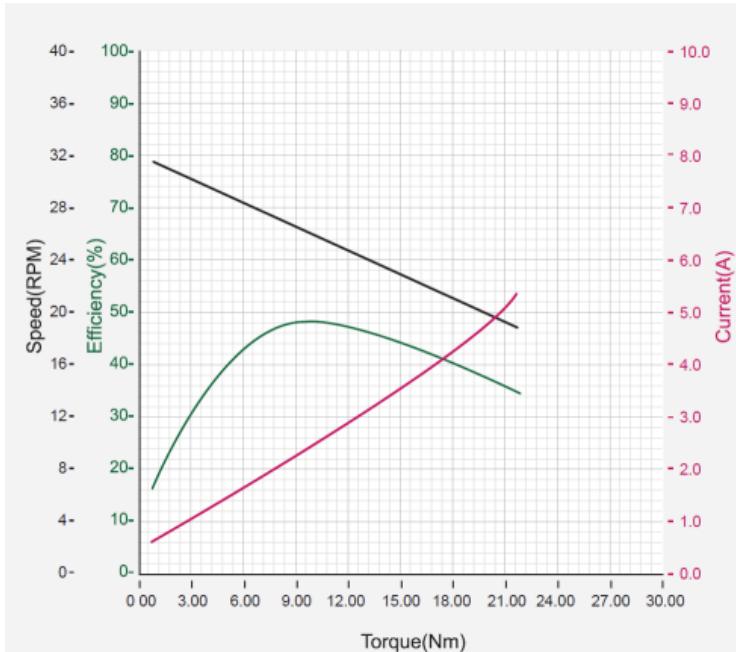
(May cause injury or damage to product)

- Do not operate the product at a temperature exceeding -5 ~ 55 [°C] range.
- Do not insert sharp blades nor pins during product operation.

**ATTENTION**

(May cause injury or damage to product)

- Do not disassemble or modify product.
- Do not drop or apply strong shock to product.

**1. 1. Performance Graph**[Show Enlarged Graph](#)

Enter Search Terms



## Apêndice B

## PH42-020-S300-R

- 1. Specifications >
- 2. Control Table >
- 3. How to Assemble >
- 4. Maintenance
- 5. Reference >



TOP

**NOTE :** The Max Torque and the Stall Torque of Performance Graph are different in measurement methods. Stall torque is a measured value of the momentary torque that it can reach. This is generally how RC servos are measured. The Performance graph is also called as N-T curves, which is measured with the gradually increasing load. The actual motor operation environment is closer to the performance graph, not stall torque method. For this reason, the performance graph is broadly used in the industrial field. Generally, Max Torque of the Performance Graph is less than the Stall Torque.

**CAUTION : When supplying power**

- It is recommended using ROBOTIS controller or SMPS2DYNAMIXEL.
- Do not connect or disconnect DYNAMIXEL when power is being supplied.
- In case of DYNAMIXEL PRO and DYNAMIXEL-P series, please supply power through 24V power port.

## 2. Control Table

The Control Table is a structure of data implemented in the device. Users can read a specific Data to get status of the device with Read Instruction Packets, and modify Data as well to control the device with WRITE Instruction Packets.

**WARNING :** DYNAMIXEL-P series use different Control Table from DYNAMIXEL PRO series. Please pay attention when replacing DYNAMIXEL PRO with DYNAMIXEL-P series.

### 2. 1. Control Table, Data, Address

The Control Table is a structure that consists of multiple Data fields to store status or to control the device. Users can check current status of the device by reading a specific Data from the Control Table with Read Instruction Packets. WRITE Instruction Packets enable users to control the device by changing specific Data in the Control Table. The Address is a unique value when accessing a specific Data in the Control Table with Instruction Packets. In order to read or write data, users must designate a specific Address in the Instruction Packet. Please refer to [Protocol 2.0](#) for more details about Instruction Packets.

**NOTE :** Two's complement is applied for the negative value. For more information, please refer to [Two's complement](#) from Wikipedia.

#### 2. 1. 1. Area (EEPROM, RAM)

The Control Table is divided into 2 Areas. Data in the RAM Area is reset to initial values when the power is reset(Volatile). On the other hand, data in the EEPROM Area is maintained even when the device is powered off(Non-Volatile).

**Data in the EEPROM Area can only be written to if Torque Enable(512) is cleared to '0'(Off).**

#### 2. 1. 2. Size

The Size of data varies from 1 ~ 4 bytes depend on their usage. Please check the size of data when updating the data with an Instruction Packet. For data larger than 2 bytes will be saved according to [Little Endian](#).

#### 2. 1. 3. Access

The Control Table has two different access properties. 'RW' property stands for read and write access permission while 'R' stands for read only access permission. Data with the read only property cannot be changed by the WRITE Instruction. Read only property('R') is generally used for measuring and monitoring purpose, and read write property('RW') is used for controlling device.

## ANEXO C

### Especificações do motor Dynamixel PH54-100-S500-R

Item	Especificação
Microcontrolador	ARM CORTEX-M4 (168 [MHz], 32Bit)
Motor	BLDC (Maxon)
Taxa de transmissão	9,600 [bps] ~10.5 [Mbps]
Modos de operação	Controle de torque
	Controle de velocidade
	Controle de posição
	Controle de posição extendida (mais do que uma revolução)
	Controle por PWM (Controle de tensão)
Massa	740 [g]
Dimensões (L x A x P)	54 x 108 x 54 [mm]
Resolução	1,003,846 [pulsos/rev]
Relação de transmissão	501.923 : 1
Folga	<6 [arcmin], 0.1 [°]
Carga radial	370 [N] (10 [mm] afastado da face do eixo)
Carga axial	130 [N]
Velocidade sem carga	33.3 [rev/min]
Corrente sem carga	1.13 [A]
Velocidade contínua	29.2 [rev/min]
Torque contínuo	25.3 [Nm]
Corrente contínua	5.5 [A]
Potência de saída	100 [W]
Temperatura de operação	5 ~55 [°C]
Tensão de operação	24.0 [V]
Sinal de comando	Pacotes digitais
Protocolo de comunicação	RS485 Comunicação serial asíncrona (8bit, 1 bit de parada, sem paridade)
Conexão física	RS485 Barramento multiponto
ID	253 ID (0 ~252)
Corrente de espera	40 [mA]

Tabela 25: Especificações do motor *Dynamixel PH54-100-S500-R*

**ROBOTIS e-Manual**

DYNAMIXEL PLATFORM STEAM SOFTWARE PARTS FAQ

Enter Search Terms 

[Edit on GitHub](#)  ROBOTIS  Robot Source  GitHub  

## PH54-100-S500-R

- 1. Specifications >
- 2. Control Table >
- 3. How to Assemble >
- 4. Maintenance >
- 5. Reference >



*PH54-100-S500-R*

### 1. Specifications

Item	Specifications
MCU	ARM CORTEX-M4 (168 [MHz], 32Bit)
Motor	BLDC (Maxon)
Baud Rate	9,600 [bps] ~ 10.5 [Mbps]
Operating Modes	Torque Control Mode Velocity Control Mode Position Control Mode Extended Position Control Mode PWM Control Mode(Voltage Control Mode)
Weight	740 [g]
Dimensions (W x H x D)	54 x 108 x 54 [mm]
Resolution	1,003,846 [pulse/rev]
Gear Ratio	501.923 : 1
Backlash	< 6 [arcmin], 0.1 [°]
Radial Load	370 [N] (10 [mm] away from the horn)
Axial Load	130 [N]
No Load Speed	33.3 [rev/min]
No Load Current	1.13 [A]
① Continuous Speed	29.2 [rev/min]
① Continuous Torque	25.3 [Nm]
① Continuous Current	5.5 [A]
Output	100 [W]
Operating Temperature	-5 ~ 55 [°C]
Input Voltage	24.0 [V]

**PH54-100-S500-R**

- 1. Specifications >
- 2. Control Table >
- 3. How to Assemble >
- 4. Maintenance
- 5. Reference >

Item	Specifications
Command Signal	Digital Packet
Protocol Type	RS485 Asynchronous Serial Communication (8bit, 1stop, No Parity)
Physical Connection	RS485 Multidrop Bus
ID	253 ID (0 ~ 252)
Standby Current	40 [mA]

1 These specifications are calculated based on the specifications of the core motor. Please consult ROBOTIS for the long term use or special use, or else refer to the Performance Graph for general use.

**DANGER**

(May cause serious injury or death)

- Never place items containing water, flammables, and solvents near product.
- Never place fingers, arms, toes, and other body parts near product during operation.
- Cut power off if product emits strange odors or smoke.
- Keep product out of reach of children.
- Check the power's polarity before wiring.

**CAUTION**

(May cause injury or damage to product)

- Do not operate the product at a temperature exceeding -5 ~ 55 [°C] range.
- Do not insert sharp blades nor pins during product operation.

**ATTENTION**

(May cause injury or damage to product)

- Do not disassemble or modify product.
- Do not drop or apply strong shock to product.

**1. 1. Performance Graph**

TOP

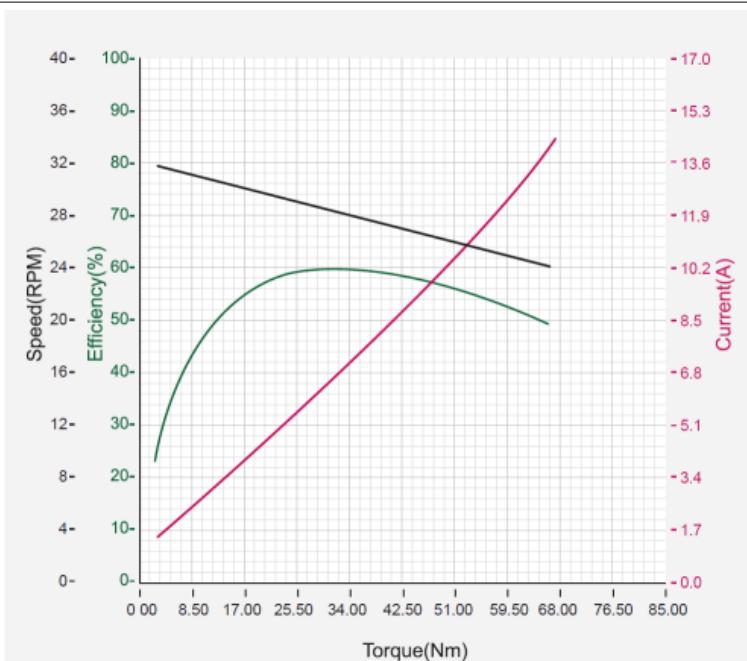
🔍

G
▲ TOP

---

**PH54-100-S500-R**

- 1. Specifications ➤
- 2. Control Table ➤
- 3. How to Assemble ➤
- 4. Maintenance ➤
- 5. Reference ➤



[Show Enlarged Graph](#)

**NOTE :** The Max Torque and the Stall Torque of Performance Graph are different in measurement methods. Stall torque is a measured value of the momentary torque that it can reach. This is generally how RC servos are measured. The Performance graph is also called as N-T curves, which is measured with the gradually increasing load. The actual motor operation environment is closer to the performance graph, not stall torque method. For this reason, the performance graph is broadly used in the industrial field. Generally, Max Torque of the Performance Graph is less than the Stall Torque.

#### CAUTION : When supplying power

- It is recommended using ROBOTIS controller or SMPS2DYNAMIXEL.
- Do not connect or disconnect DYNAMIXEL when power is being supplied.
- In case of DYNAMIXEL PRO and DYNAMIXEL-P series, please supply power through 24V power port.

## 2. Control Table

The Control Table is a structure of data implemented in the device. Users can read a specific Data to get status of the device with Read Instruction Packets, and modify Data as well to control the device with WRITE Instruction Packets.

**WARNING :** DYNAMIXEL-P series use different Control Table from DYNAMIXEL PRO series. Please pay attention when replacing DYNAMIXEL PRO with DYNAMIXEL-P series.

### 2. 1. Control Table, Data, Address

The Control Table is a structure that consists of multiple Data fields to store status or to control the device. Users can check current status of the device by reading a specific Data from the Control Table with Read Instruction Packets. WRITE Instruction Packets enable users to control the device by changing specific Data in the Control Table. The Address is a unique value when accessing a specific Data in the Control Table with Instruction Packets. In order to read or write data, users must

## ANEXO D

### Especificações do motor Dynamixel PH54-200-S500-R

Item	Especificação
MCU	ARM CORTEX-M4 (168 [MHz], 32Bit)
Motor	BLDC (Maxon)
Taxa de transmissão	9,600 [bps] ~10.5 [Mbps]
Modos de operação	Modo de controle de torque Modo de controle de velocidade Modo de controle de posição Modo de controle de posição estendida Modo de controle PWM (modo de controle de tensão)
Peso	855 [g]
Dimensões (c x l x h)	54 x 126 x 54 [mm]
Resolução	1,003,846 [pulse/rev]
Relação de engrenagem	501.923 : 1
Folga	<6 [arcmin], 0.1 [°]
Carga radial	370 [N] (10 [mm] afastado da face do eixo)
Carga axial	130 [N]
Velocidade (sem carga)	33.1 [rev/min]
Corrente (sem carga)	1.65 [A]
Velocidade contínua	29.0 [rev/min]
Torque contínuo	44.7 [Nm]
Corrente contínua	9.3 [A]
Saída	200 [W]
Temperatura de operação	5 ~55 [°C]
Tensão operacional	24.0 [V]
Sinal de comando	Pacote digital
Tipo de protocolo	RS485 Comunicação Serial Assíncrona (8bit, 1stop, Sem paridade)
Conexão física	RS485 Barramento multiponto
ID	253 ID (0 ~252)
Corrente de repouso	40 [mA]

Tabela 26: Especificações do motor *Dynamixel PH54-200-S500-R*

Enter Search Terms



## PH54-200-S500-R

- 1. Specifications
- 1. 1. Performance Graph
- 2. Control Table
- 3. How to Assemble
- 4. Maintenance
- 5. Reference

[Edit on GitHub](#)

ROBOTIS



Robot Source



GitHub



TOP

*PH54-200-S500-R*

## 1. Specifications

Item	Specifications
MCU	ARM CORTEX-M4 (168 [MHz], 32Bit)
Motor	BLDC (Maxon)
Baud Rate	9,600 [bps] ~ 10.5 [Mbps]
Operating Modes	Torque Control Mode Velocity Control Mode Position Control Mode Extended Position Control Mode PWM Control Mode(Voltage Control Mode)
Weight	855 [g]
Dimensions (W x H x D)	54 x 126 x 54 [mm]
Resolution	1,003,846 [pulse/rev]
Gear Ratio	501.923 : 1
Backlash	< 6 [arcmin], 0.1 [°]
Radial Load	370 [N] (10 [mm] away from the horn)
Axial Load	130 [N]
No Load Speed	33.1 [rev/min]
No Load Current	1.65 [A]
Continuous Speed	29.0 [rev/min]
Continuous Torque	44.7 [Nm]
Continuous Current	9.3 [A]
Output	200 [W]
Operating Temperature	-5 ~ 55 [°C]
Input Voltage	24.0 [V]
Command Signal	Digital Packet
Protocol Type	RS485 Asynchronous Serial Communication (8bit, 1stop, No Parity)
Physical Connection	RS485 Multidrop Bus

## Apêndice B

Enter Search Terms



## PH54-200-S500-R

- 1. Specifications
- 1. 1. Performance Graph
- 2. Control Table
- 3. How to Assemble
- 4. Maintenance
- 5. Reference



TOP

1 These specifications are calculated based on the specifications of the core motor. Please consult ROBOTIS for the long term use or special use, or else refer to the Performance Graph for general use.

**DANGER**

(May cause serious injury or death)

- Never place items containing water, flammables, and solvents near product.
- Never place fingers, arms, toes, and other body parts near product during operation.
- Cut power off if product emits strange odors or smoke.
- Keep product out of reach of children.
- Check the power's polarity before wiring.

**CAUTION**

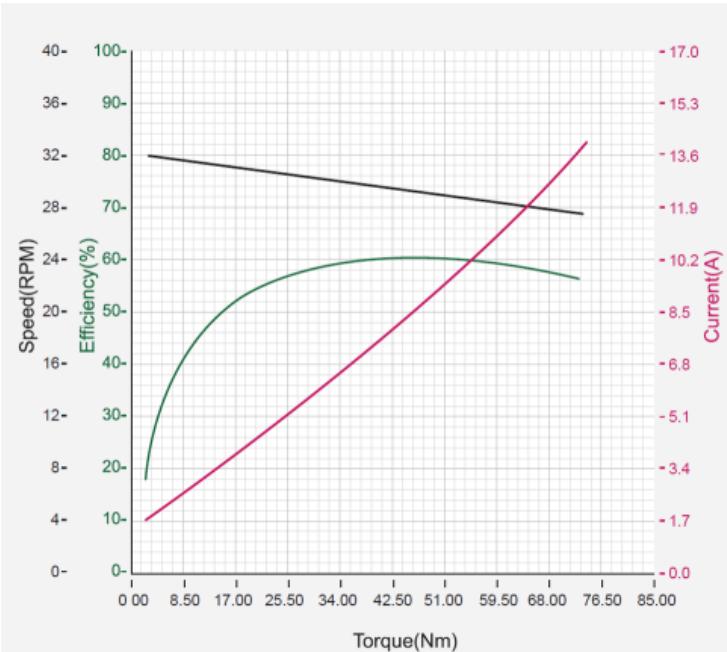
(May cause injury or damage to product)

- Do not operate the product at a temperature exceeding -5 ~ 55 [°C] range.
- Do not insert sharp blades nor pins during product operation.

**ATTENTION**

(May cause injury or damage to product)

- Do not disassemble or modify product.
- Do not drop or apply strong shock to product.

**1. 1. Performance Graph**[Show Enlarged Graph](#)

## Apêndice B

Enter Search Terms	
<b>PH54-200-S500-R</b>	
1. Specifications	▼
1. 1. Performance Graph	▶
2. Control Table	▶
3. How to Assemble	▶
4. Maintenance	▶
5. Reference	▶

**NOTE :** The Max Torque and the Stall Torque of Performance Graph are different in measurement methods. Stall torque is a measured value of the momentary torque that it can reach. This is generally how RC servos are measured. The Performance graph is also called as N-T curves, which is measured with the gradually increasing load. The actual motor operation environment is closer to the performance graph, not stall torque method. For this reason, the performance graph is broadly used in the industrial field. Generally, Max Torque of the Performance Graph is less than the Stall Torque.



TOP

**CAUTION : When supplying power**

- It is recommended using ROBOTIS controller or SMPS2DYNAMIXEL.
- Do not connect or disconnect DYNAMIXEL when power is being supplied.
- In case of DYNAMIXEL PRO and DYNAMIXEL-P series, please supply power through 24V power port.

## 2. Control Table

The Control Table is a structure of data implemented in the device. Users can read a specific Data to get status of the device with Read Instruction Packets, and modify Data as well to control the device with WRITE Instruction Packets.

**WARNING :** DYNAMIXEL-P series use different Control Table from DYNAMIXEL PRO series. Please pay attention when replacing DYNAMIXEL PRO with DYNAMIXEL-P series.

### 2. 1. Control Table, Data, Address

The Control Table is a structure that consists of multiple Data fields to store status or to control the device. Users can check current status of the device by reading a specific Data from the Control Table with Read Instruction Packets. WRITE Instruction Packets enable users to control the device by changing specific Data in the Control Table. The Address is a unique value when accessing a specific Data in the Control Table with Instruction Packets. In order to read or write data, users must designate a specific Address in the Instruction Packet. Please refer to [Protocol 2.0](#) for more details about Instruction Packets.

**NOTE :** Two's complement is applied for the negative value. For more information, please refer to [Two's complement](#) from Wikipedia.

#### 2. 1. 1. Area (EEPROM, RAM)

The Control Table is divided into 2 Areas. Data in the RAM Area is reset to initial values when the power is reset(Volatile). On the other hand, data in the EEPROM Area is maintained even when the device is powered off(Non-Volatile).

**Data in the EEPROM Area can only be written to if Torque Enable(512) is cleared to '0'(Off).**

#### 2. 1. 2. Size

The Size of data varies from 1 ~ 4 bytes depend on their usage. Please check the size of data when updating the data with an Instruction Packet. For data larger than 2 bytes will be saved according to [Little Endian](#).

#### 2. 1. 3. Access

The Control Table has two different access properties. 'RW' property stands for read and write access permission while 'R' stands for read only access permission. Data with the read only property cannot be changed by the WRITE Instruction. Read only property('R') is generally used for measuring and monitoring purpose, and read write property('RW') is used for controlling device.

---

## Avaliação de medição para o desafio 2.5

---

## Avaliação da medição para o desafio 2.5

Anderson Queiroz do Vale

Aziel Martins de Freitas Júnior  
Rodrigo Formiga Farias

Jean Paulo Silva

22 julho 2020

### Introdução

Este trabalho foi realizado com intuito de analisar os resultados obtidos da corrida de revezamento utilizando o método de análise de variância (ANOVA). Esse método é um procedimento usado para comparar a distribuição de três ou mais grupos em amostras independentes. Ele também é uma forma de resumir um modelo de regressão linear através da decomposição da soma dos quadrados para cada fonte de variação no modelo e, utilizando o teste F, testar se a hipótese de qualquer fonte de variação no modelo é igual a zero.

### Coleta dos dados

Os dados coletados foram retirados de 4 computadores distintos, como mostra a tabela abaixo.

Table 1: Características dos computadores.

	Máquina 1 (Rodrigo)	Máquina 2 (Aziel)	Máquina 3 (Anderson)	Máquina 4 (Jean)
Processador	i7-8550U	i5-8300H	i3-8130U	i5-9400F
Memória	8 GB RAM	8 GB RAM	6 GB RAM	16 GB RAM
GPU	GTX 1050	GTX 1050	MX 920	GTX 1650

A partir das medições foi possível extrair os tempos dos quatro robôs nas corridas de testes. O tempo, em segundos, é resultado do percurso total de corrida concluído por cada robô individualmente para cada computador distinto, como mostra a Tabela 2. Os robôs são classificados como RAJA 1, RAJA 2, RAJA 3 e RAJA 4, e o número das máquinas são equivalentes o da Tabela 1.

Table 2: Tempos na corrida de revezamento.

Máquina	RAJA	Teste	Tempo (s)	Máquina	RAJA	Teste	Tempo (s)
1	1	1	16.83	3	1	1	16.35
1	1	2	16.74	3	1	2	16.29
1	1	3	16.81	3	1	3	16.4
1	2	1	17.51	3	2	1	17.8
1	2	2	17.75	3	2	2	17.87
1	2	3	17.73	3	2	3	17.77
1	3	1	22.32	3	3	1	22.27
1	3	2	21.52	3	3	2	21.39
1	3	3	21.6	3	3	3	20.37
1	4	1	19.97	3	4	1	19.46
1	4	2	19.62	3	4	2	19.68
1	4	3	19.8	3	4	3	21.54

Máquina	RAJA	Teste	Tempo (s)	Máquina	RAJA	Teste	Tempo (s)
2	1	1	16.35	4	1	1	16.39
2	1	2	16.31	4	1	2	16.38
2	1	3	16.32	4	1	3	16.29
2	2	1	17.71	4	2	1	17.71
2	2	2	17.72	4	2	2	17.72
2	2	3	17.72	4	2	3	17.72
2	3	1	21.34	4	3	1	21.51
2	3	2	21.53	4	3	2	22.24
2	3	3	21.49	4	3	3	21.68
2	4	1	19.68	4	4	1	19.58
2	4	2	21.12	4	4	2	19.65
2	4	3	19.75	4	4	3	19.71

A avaliação do sistema de medição resultou nas tabelas acima e uma representação gráfica, que são explicados abaixo.

## Interpretação da avaliação do sistema de medição

O “Total Gage R&R” calculado para a variabilidade no estudo (%StudyVar) foi de 19,17%, como mostra a Tabela 3. Segundo Automotive Industry Action Group (AIAG), no qual diz que o percentual de aceitabilidade do processo pode ser entre 10%-30% dependendo da aplicação, pode-se afirmar que o estudo da análise da variabilidade do sistema de corrida de revezamento pode ser aceito. A variação entre medidas mais expressiva se dá pelo item “Part-to-part”, medindo 98,14%, relacionado aos objetos medidos, encontra-se próxima dos 100%. O “Number of Distinct Categories” encontrado foi 7, e está acima do valor aceitável de 2.

Table 3: *Total gage R&R*

	StdDev	StudyVar	%StudyVar
Total Gage R&R	0.4475435	2.685261	19.17
Repeatability	0.4475435	2.685261	19.17
Reproducibility	0.0000000	0.000000	0.00
Máquina	0.0000000	0.000000	0.00
Part-To-Part	2.2909039	13.745424	98.14
Total Variation	2.3342099	14.005260	100.00

## Componentes da variação

A variabilidade recebe contribuição inexpressiva da repetição dos testes bem como dos computadores utilizados no teste, o que demonstra consistência aceitável do sistema de medição.

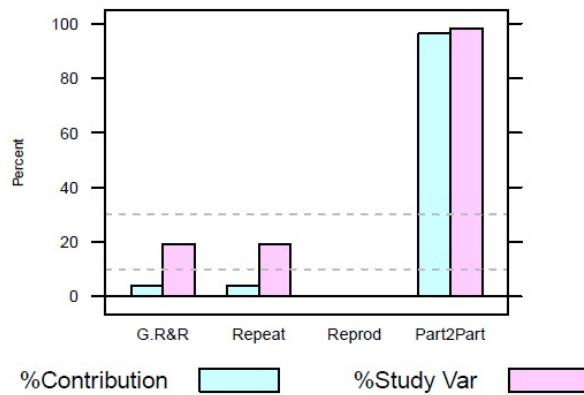


Figure 1: Componentes da variação

### Variância por máquina

Este gráfico indica a variância nos tempos na avaliação em cada uma das máquinas. Os valores são próximos e estão dentro de uma faixa aceitável, delimitada pelas linhas horizontais vermelhas, exceto pela Máquina 3, que apresentou problemas de hardware que acabaram sendo traduzidos na simulação.

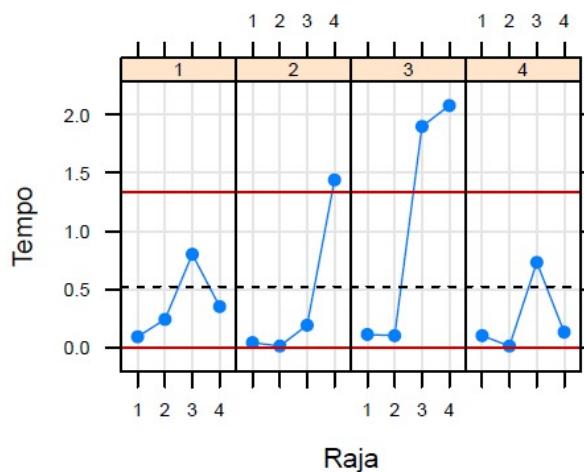


Figure 2: Variância por máquina

### Gráfico de tempo médio por avaliador

Este gráfico indica as médias nos sistemas de medição, que são os *avaliadores*, e seu comportamento é bastante similar em todos eles. Os RAJA 3 e 4 são os que percorrem um trajeto curvilíneo e com velocidade deliberadamente menor e por isso apresentam um tempo médio maior os outros.

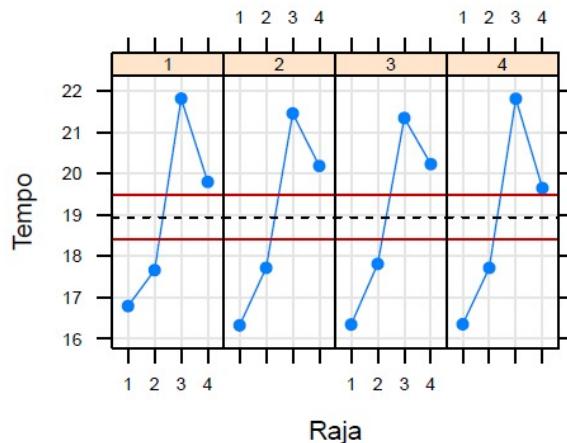


Figure 3: Tempo médio por avaliador

### Tempo por cada RAJA

As medidas de tempo para cada RAJA estão aglomeradas verticalmente de modo a exibir como as medições se dispersam em cada um dos sistemas de medição. Nota-se a similaridade entre este gráfico e o anterior nos pontos onde há maior densidade de medições.

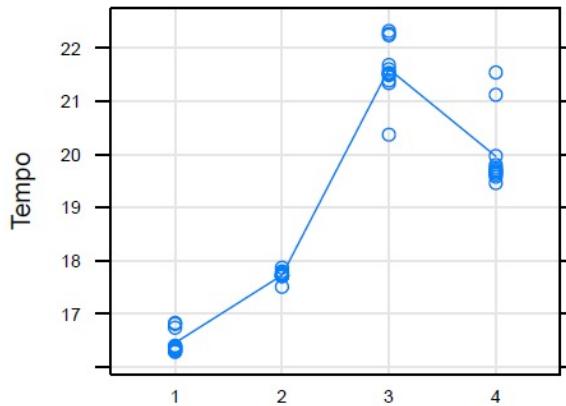


Figure 4: Tempo por RAJA

### Tempo por máquina

As medidas de tempo para cada um dos RAJA estão representadas na vertical bem como o valor médio, representado pela linha horizontal azul. O teste pode ser considerado bom se a amplitude das medidas em cada máquina for similar, bem como a horizontalidade da linha. As amplitudes das medições são devidas aos objetos testados e não instrumentos de medição utilizados.

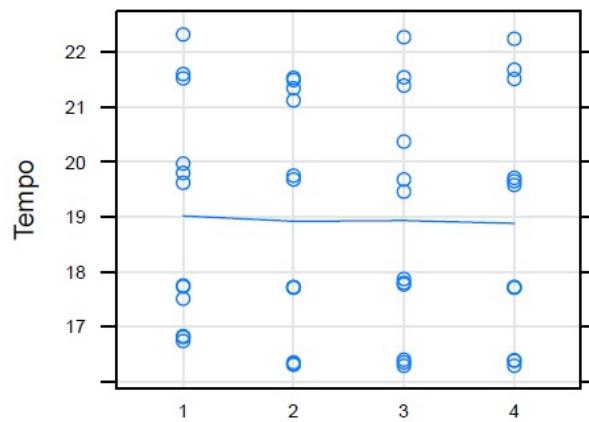


Figure 5: Tempo por máquina

### Interação entre os avaliadores

Este gráfico é uma sobreposição dos gráficos anteriores, a fim de ilustrar a semelhança entre os valores das médias para cada RAJA.

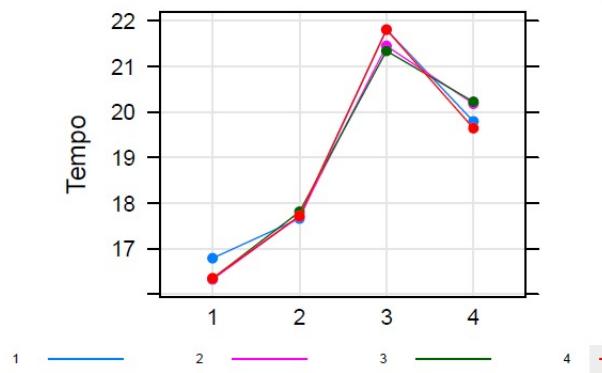


Figure 6: Interação entre os avaliadores

---

**SISTEMA PARA IDENTIFICAÇÃO E  
MANIPULAÇÃO DE OBJETOS EM AMBIENTES  
EXTERNOS UTILIZANDO PLATAFORMA  
MÓVEL**

---

**SISTEMA PARA IDENTIFICAÇÃO E MANIPULAÇÃO DE  
OBJETOS EM AMBIENTES EXTERNOS UTILIZANDO  
PLATAFORMA MÓVEL**

Relatório Final do Projeto Curupira

**Autores:**

Jean Paulo Silva  
Pedro Paulo Ventura Tecchio  
Vinicius José Gomes de Araújo Felismino

**Orientadores:**

Rebeca Tourinho Lima  
Tiago Pereira de Souza  
Marco Antonio dos Reis

**Salvador  
Bahia, Brasil**

**Dezembro de 2020**



Título: Sistema para identificação e manipulação de objetos em ambientes externos utilizando plataforma móvel <b>PROD. TEC. RoSA -</b> Classificação: ( ) Confidencial (X) Restrito ( ) Uso Interno ( ) Público	Versão 01
---	--------------

**Informações Confidenciais** - Informações estratégicas para o RoSA e Senai Cimatec.

Seu manuseio é restrito a usuários previamente autorizados pelo Gestor da área.

**Informações Restritas** - Informação cujo conhecimento, manuseio e controle de acesso devem estar limitados a um grupo restrito de pesquisadores que necessitam utilizá-la para exercer suas atividades profissionais.

**Informações de Uso Interno** - São informações destinadas à utilização interna por pesquisadores e parceiros.

**Informações Públicas** - Informações que podem ser distribuídas ao público externo, o que, usualmente, é feito através dos canais apropriados.

### Dados Internacionais de Catalogação na Publicação (CIP)

Jean Paulo Silva Pedro Paulo Ventura Tecchio Vinicius José Gomes de Araújo Felismino 000 Rebeca Tourinho Lima Tiago Pereira de Souza Marco Antonio dos Reis	Sistema para identificação e manipulação de objetos em ambientes externos utilizando plataforma móvel Salvador Bahia, Brasil Dezembro de 2020	Keywords: 1. Mobile Robots. 2. Robotic Manipulator. 3. Autonomy. 000
---	--	--



## SUMÁRIO EXECUTIVO

O projeto realizado pela equipe Curupira se configura sob o programa de formação de Novos Talentos do [Centro de Competência em Robótica e Sistemas Autônomos \(CCRoSA\)](#) do SENAI/CIMATEC. Devido a necessidade em aplicar os conceitos vistos em robótica móvel e manipuladores durante o ano letivo, foi proposto pelos os orientadores e pesquisadores do laboratório um desafio que consiste em integrar um manipulador robótico a uma dada plataforma móvel. O SENAI/CIMATEC firmou compromisso financeiro e humano para o desenvolvimento da solução e com isso é o principal fomentador do projeto. O *kick-off* do projeto foi no dia 30/10/2020. O prazo de execução foi planejado para 35 dias.

## RESUMO

O presente relatório apresenta a solução alcançada pela equipe Curupira para o desafio proposto pelos os orientadores e pesquisadores do CCRoSA do SENAI/CIMATEC. O problema proposto consiste em utilizar o manipulador robótico JeRoTIMON integrado a plataforma móvel *Warthog* da *Clearpath Robotics* para realizar a missão de navegação, mapeamento, localização, busca e “desarme” de uma “bomba” no pátio do CIMATEC 4 de forma autônoma. Para realização desta missão foi utilizado o *Simultaneous Localization and Mapping (SLAM) Cartographer* da *Google LLC* em conjunto com os seguintes sensores: *Light Detection And Ranging (LiDAR)*, *Global Positioning System (GPS)* e uma câmera *Red Green Blue (RGB)*. Também utilizou-se o *software Gazebo*, a ferramenta de visualização *Rviz* e o *framework MoveIt* para o manipulador. A solução em ambiente simulado apresentou melhores resultados em relação ao real. Contudo, ambas necessitam de ajustes para sua finalização. Ao final desse documento, podem ser encontradas as lições aprendidas durante a elaboração desta solução.

## ABSTRACT

This report shows the solutions provided by Team Curupira in order to solve a challenge proposed by SENAI/CIMATEC's CCRoSA's advisors. The proposed problem consists of integrating the robotic manipulator JeRoTIMON to the mobile platform Warthog from Clearpath Robotics in order to complete the following mission: navigate, map, localize, search for and disable “the bomb” placed inside the open space in front of CIMATEC 4 in an autonomous manner. The provided solution makes use of Google LLC’s Cartographer to do SLAM with the following sensors: LiDAR, GPS and a RGB camera. Simulations were done using Gazebo, while visualization tasks were done with Rviz. The MoveIt framework was used to control the robotic manipulator. Although, the proposed solution achieved better results inside the simulated environment, both still requires some adjustments in order to be completed. The learned lessons during this project development are available at the end of this document.

## LISTA DE FIGURAS

Figura 1:	Arquitetura Geral Simplificada.	16
Figura 2:	Estrutura Analítica do Projeto.	20
Figura 3:	UGV <i>Warthog</i> da <i>Clearpath Robotics</i>	21
Figura 4:	Manipulador <i>JeRoTimon</i> .	22
Figura 5:	<i>Lidar Velodyne VLP-16</i>	23
Figura 6:	GPS <i>Piksi V2</i> da <i>Swiftnav</i>	23
Figura 7:	Câmera <i>Basler acA4600-10uc</i> .	24
Figura 8:	Câmera <i>Kowa LM8HC</i> .	25
Figura 9:	Intel NUC5i5RYK.	26
Figura 10:	<i>Switch RB951G, MikroTik</i> .	26
Figura 11:	Leitura do GPS.	27
Figura 12:	Leitura do Velodyne VLP-16.	28
Figura 13:	Identificação da “bomba” utilizando câmera <i>Basler</i> .	28
Figura 14:	Mapeamento exemplo - <i>Cartographer</i> .	29
Figura 15:	Exemplo de exploração - <i>RRT</i> .	30
Figura 16:	Suporte mecânico e sensores integrados.	31
Figura 17:	Resultado de mapeamento errôneo com o <i>Google Cartographer</i> .	34
Figura 18:	Resultado de mapeamento correto com o <i>Google Cartographer</i> .	34
Figura 19:	Exploração, localização e mapeamento autônomos.	35
Figura 20:	Simulação de manipulação.	36
Figura 21:	Integração do protótipo.	36
Figura 22:	Resultado de mapeamento com o <i>Google Cartographer</i> em ambiente real.	37
Figura 23:	Integração da manipulação.	38
Figura 24:	Resultado da detecção da “bomba”.	39
Figura 25:	Esquema elétrico das conexões do Warthog.	52
Figura 26:	Esquema elétrico das conexões da NUC.	53
Figura 27:	Esquema elétrico das conexões USB.	54
Figura 28:	Esquema elétrico das conexões do manipulador.	55
Figura 29:	Desenho mecânico do conversor ROBOTIS U2D2 (RS485 - USB).	111

## LISTA DE TABELAS

Tabela 1:	Lições aprendidas.	41
Tabela 2:	Especificações da plataforma móvel <i>Warthog</i> da <i>Clearpath Robotics</i> .	61
Tabela 3:	Especificações do computador Intel NUC 5i5RYK.	63
Tabela 4:	Especificações do <i>switch</i> MikroTik RB951G-2HnD.	66
Tabela 5:	Especificações do <i>lidar</i> Velodyne VLP-16.	68
Tabela 6:	Especificações do GPS Swiftnav Piksi V2.	71
Tabela 7:	Especificações da câmera Basler acA4600-10uc.	76
Tabela 8:	Especificações da lente Kowa LM8HC para sensor de 1 polegada de diâmetro.	85
Tabela 9:	Especificações do manipulador JeRoTimon.	87
Tabela 10:	Especificações do conversor ROBOTIS U2D2 (RS485-USB).	101



## LISTRA DE SÍMBOLOS E ABREVIATURAS

**CCRoSA** Centro de Competência em Robótica e Sistemas Autônomos

**CPU** *Central Processing Unit*

**EAP** Estrutura Analítica do Projeto

**FPS** *Frames Per Second*

**GPS** *Global Positioning System*

**GPU** *Graphic Processing Unit*

**IMU** *Inertial Measurement Unit*

**LiDAR** *Light Detection And Ranging*

**NUC** *Next Unit of Computing*

**OMS** Organizaçāo Mundial da Saúde

**RGB** *Red Green Blue*

**ROS** *Robot Operating System*

**RTK** *Real Time Kinematics*

**SLAM** *Simultaneous Localization and Mapping*

**SSH** *Secure Socket Shell*

**QFD** *Quality Function Deployment*

**UGV** *Unmanned Ground Vehicle*

**USB** *Universal Serial Bus*

# SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>11</b>
1.1 Objetivo . . . . .	11
1.2 Justificativa . . . . .	11
1.3 Requisitos do Cliente . . . . .	11
<b>2 O SISTEMA</b>	<b>13</b>
2.1 Escopo . . . . .	13
2.2 Condições do ambiente para o desenvolvimento . . . . .	13
2.3 Requisitos Técnicos . . . . .	13
2.4 Desdobramento da Função Qualidade . . . . .	14
2.5 Arquitetura Geral . . . . .	15
<b>3 DESENVOLVIMENTO</b>	<b>19</b>
3.1 Estrutura Analítica do Sistema . . . . .	19
3.2 Especificação de componentes . . . . .	20
3.3 Funcionalidades . . . . .	27
<b>4 SIMULAÇÃO E INTEGRAÇÃO</b>	<b>33</b>
4.1 Simulação . . . . .	33
4.2 Integração . . . . .	36
<b>5 GESTÃO DO CONHECIMENTO</b>	<b>41</b>
5.1 Lições aprendidas . . . . .	41
<b>6 CONCLUSÃO</b>	<b>43</b>
<b>REFERÊNCIAS</b>	<b>45</b>
<b>APÊNDICE A Desenho mecânico</b>	<b>47</b>
<b>APÊNDICE B Esquema elétrico de potência</b>	<b>51</b>
<b>APÊNDICE C Desdobramento da Função Qualidade(QFD)</b>	<b>57</b>

<b>ANEXO A Especificação de componentes utilizados</b>	<b>61</b>
A.1 Plataforma móvel Warthog . . . . .	61
A.2 Computador Intel NUC . . . . .	63
A.3 Switch MikroTik RB951G . . . . .	66
A.4 Lidar Velodyne VLP-16 . . . . .	68
A.5 GPS Swiftnav Piksi V2 . . . . .	71
A.6 Câmera Basler ace acA4600 . . . . .	76
A.7 Lente Kowa LM8HC . . . . .	85
A.8 Manipulador JeRoTimon . . . . .	87
A.9 Conversor ROBOTIS U2D2 (RS485 - USB) . . . . .	101



# 1 INTRODUÇÃO

Este documento visa agrupar todos os conteúdos abordados no processo de desenvolvimento e mostrar os resultados da integração e seleção de soluções que compõem a fase final do projeto: “Sistema para identificação e manipulação de objetos em ambientes externos utilizando plataforma móvel”.

## 1.1 Objetivo

O objetivo deste projeto é integrar uma plataforma móvel, um manipulador robótico e um conjunto de sensores com intuito de explorar autonomamente um ambiente externo de 4,000 m<sup>2</sup> de área utilizando [SLAM](#), identificar um objeto via câmera e interagir com este objeto por meio do manipulador robótico.

## 1.2 Justificativa

O presente trabalho tem como impulsionador principal a capacitação de novos pesquisadores preparados para solucionar os mais diversos problemas relacionados a robótica e sistemas autônomos.

Este relatório descreve o processo de integração de um sistema robótico desenvolvido no [CCRoSA](#) do SENAI/CIMATEC que constitui o trabalho final do programa de formação Novos Talentos.

## 1.3 Requisitos do Cliente

Através de reuniões com os orientadores, foram levantados os seguintes requisitos do cliente:

### **O sistema deverá ser autônomo**

As tarefas realizadas pelo sistema integrado precisam ser executadas sem nenhuma intervenção do operador. Para isso, há um computador central que determina as tomadas de decisões do sistema.

### **O sistema deverá localizar e desviar de obstáculos**

Para o sistema navegar no seu ambiente de trabalho é necessário que ele consiga reconhecer um obstáculo de a fim de desviá-lo. Com o seu sistema de percepção, uma nova rota deve ser designada para a base móvel.

## O sistema deverá localizar a região de interesse

Utilizando a funcionalidade de exploração, o robô se desloca autonomamente por toda a área definida, mapeando cada espaço deslocado.

## O sistema deverá identificar a “bomba” no local de trabalho estabelecido

Utilizando a funcionalidade de percepção, as câmeras conseguem detectar a “bomba”, determinando sua distância em relação à base móvel.

## O sistema deverá desarmar a “bomba” utilizando o manipulador robótico

A interação que o sistema possui com o objeto no seu ambiente de trabalho é feita a partir de manipulador, representando o corte de fios elétricos com o seu *endeffector*.

## O sistema possuirá monitoramento remoto

Durante o seu procedimento de trabalho, as tomadas de decisão e suas atividades podem ser monitoradas por um computador que esteja conectado na mesma rede. Também é possível parar o sistema com botões de emergência no controlador sem fio da base móvel.

## O sistema possuirá modularidade

Foi adicionado um suporte mecânico à base móvel com o objetivo de acoplar sensores, atuadores e periféricos ao sistema.

## O sistema deverá trabalhar em ambientes externos

A resistência do suporte mecânico foi definido para o ambiente de trabalho escolhido.

## O sistema possuirá uma versão simulada

Caso seja necessário a adição de novos obstáculos ou modificar as configurações da movimentação do sistema, ele possui uma versão simulada que pode ser utilizada para validação.

## 2 O SISTEMA

O desenvolvimento da solução do desafio proposto tem alta relevância pois parte da necessidade de aplicar os conceitos vistos em manipuladores e robótica móvel durante os encontros que ocorreram neste ano. Além disso, este projeto não se limitou apenas ao uso da simulação mas também foi testado e executado em ambiente externo.

### 2.1 Escopo

A solução consiste na plataforma móvel *Warthog* da *Clearpath Robotics* capaz de navegar em ambientes internos ou externos. Para realizar a missão proposta pelos orientadores, foi necessário projetar e anexar um suporte mecânico na estrutura do robô móvel com o intuito de garantir a integração do manipulador JeRoTIMON e dos sensores **GPS**, **LiDAR** e câmera RGB, sendo estes, necessários para o cumprimento da atividade.

### 2.2 Condições do ambiente para o desenvolvimento

O ambiente para a concepção da solução, devido aos efeitos da pandemia do vírus COVID-19, foi realizado de forma híbrida, ou seja, foi elaborado pelos orientadores um plano em que designava os dias que o grupo teria acesso ao laboratório de Robótica e Sistemas Autônomos do SENAI/CIMATEC. Nos dias em que o grupo não estava presente no laboratório o trabalho era feito de forma remota (*home office*). Portanto foi de suma importância seguir as recomendações da **Organização Mundial da Saúde (OMS)** para preservar a saúde de todos os integrantes do laboratório.

### 2.3 Requisitos Técnicos

Os requisitos técnicos são as características necessárias para alcançar o objetivo do projeto. Para estabelecer estas premissas primeiramente se fez necessário, através de reuniões, fazer um levantamento das exigências solicitadas pelo cliente conforme visto na seção 1.3. Por fim o grupo responsável pelo desenvolvimento da solução levantou os requisitos técnicos que estão listados abaixo:

- A elaboração da solução prezou pela utilização do sistema operacional *Linux Ubuntu 18.04 LTS*;
- Utilizou o *framework* de robótica **Robot Operating System (ROS)** na versão *Melodic Morenia*;
- Utilizou o *framework* de manipulação *MoveIt!*;
- O *back-end* foi construído em *Python* e *C++*;

- A simulação da tarefa foi realizada através do *software Gazebo* com auxílio pela ferramenta de visualização *Rviz*;
- O terminal nativo do sistema operacional foi utilizado como *software* de interface;
- Utilizou-se o *software* de visão *OpenCV 3.3.1*;
- Utilizou-se a câmera *RGB Basler aca4600* como sensor óptico;
- O *SLAM* utilizado foi o *Cartographer* da *Google LLC*;
- Para a detecção de obstáculos foi utilizado o *Velodyne VLP-16*;
- O tipo de locomoção utilizado foi autônoma;
- Foi utilizado para a localização o sensor *GPS - Swiftnav Pixsi V2*;
- A área para realização da missão foi o pátio do CIMATEC 4;
- O sistema de comunicação com o robô utilizou *Wi-Fi 2.4GHz*
- A distância de comunicação variou entre 5 a 15m;
- A elaboração da solução buscou obter um sistema de alimentação fornecendo 12 e 24V através da base móvel;
- A autonomia operacional do sistema foi de aproximadamente 3h.

## 2.4 Desdobramento da Função Qualidade

O *Quality Function Deployment* (*QFD*), traduzido para o português como o Desdobramento da Função da Qualidade é uma ferramenta sistemática de melhoria contínua que foi desenvolvida pelo japonês Yoji Akai em 1966 no Japão. O objetivo principal desta ferramenta é ouvir, traduzir e transmitir, de forma priorizada, a voz do cliente para dentro do processo de desenvolvimento do produto (NUNES, 2019).

No apêndice C está representado a matriz *QFD* desenvolvida para a concepção da solução. Na primeira parte da matriz podemos verificar as relações entre os requisitos técnicos e os requisitos do cliente, as correlações entre os requisitos técnicos e a direção de melhoria destes requisitos caracterizando o primeiro ciclo de construção da matriz. Na segunda parte, é mostrada a relação ou confronto entre os requisitos técnicos e as funcionalidades do sistema, as correlações entre as funcionalidades e a direção de melhoria destas premissas, portanto, se encerra o segundo ciclo de construção da matriz.

A pirâmide localizada no canto superior da matriz de cada ciclo demonstra a correlação entre os requisitos técnicos e as funcionalidades, respectivamente. Ela pode ser definida por positiva, negativa ou sem correlação. Essas informações são importantes para definir

o nível de interferência de um requisito técnico ou funcionalidade em outro(a), seja de maneira positiva ou negativa.

O alvo define a variável de importância do requisito técnico e o seu valor desejado. A direção de melhoria mostra que requisitos técnicos devem focar em manter esta variável estável ou trabalhar em melhorias contínuas, o mesmo ocorre para as funcionalidades.

Essa avaliação demonstra os requisitos que receberam mais atenção durante o desenvolvimento da solução:

- *Framework* de manipulação;
- *Software* de visão;
- Sistema de percepção;
- *Software* de mapeamento;
- *Software* de localização;
- Tipo de locomoção.

Apesar de alguns requisitos receberem mais atenção que outros, não muda o fato que todos os requisitos deverão ter suas metas alcançadas como indica a matriz.

## 2.5 Arquitetura Geral

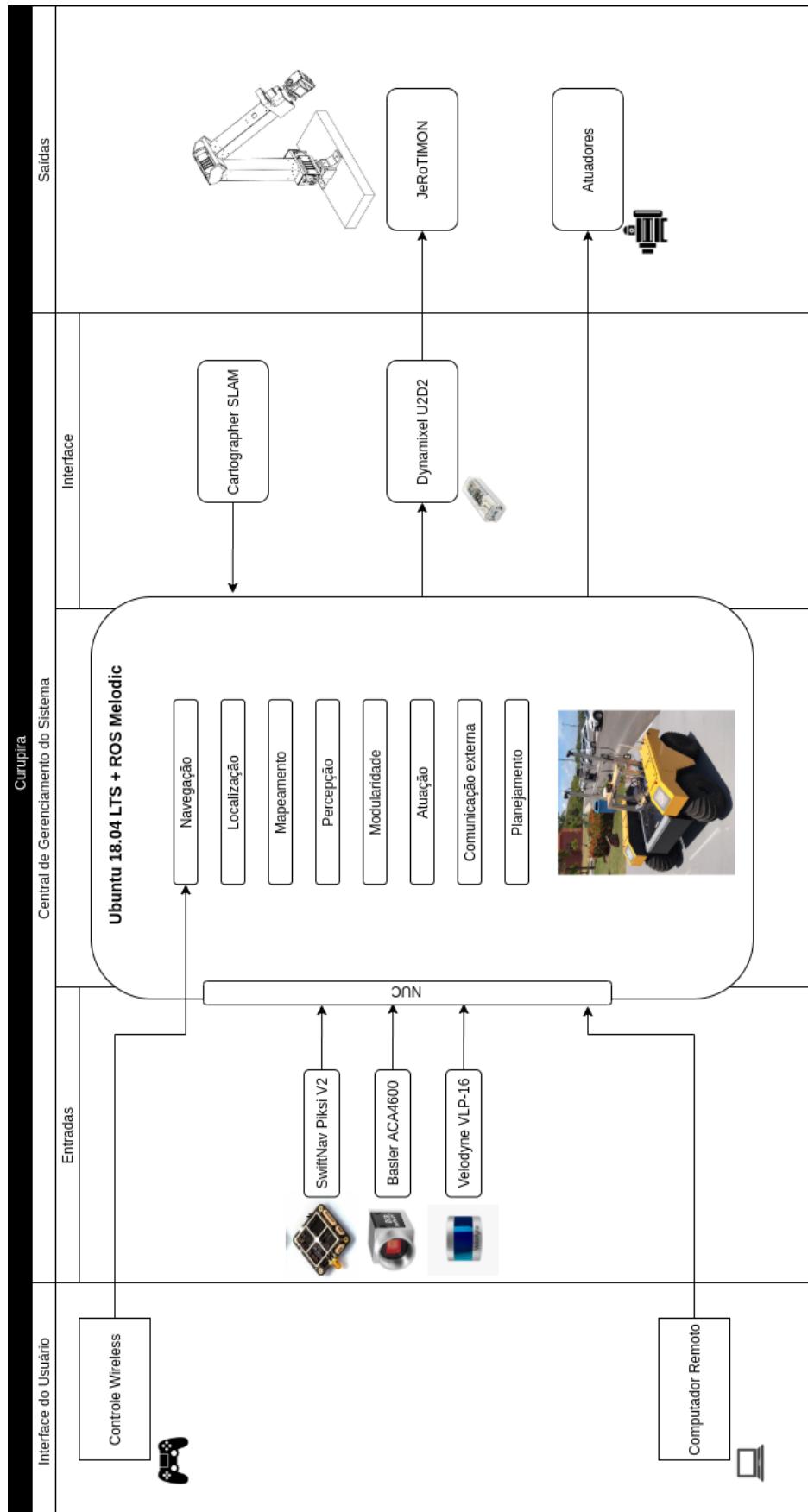
A arquitetura geral é uma representação do sistema de um jeito amplo, destacando as funcionalidades dos componentes de *hardware* e *software* e demonstrando as suas integrações.

Ela é dividida em três grandes grupos: Interface do Usuário, Central de Gerenciamento do Sistema e Saídas. A Interface do Usuário representa o monitoramento remoto do sistema a partir da visualização das atividades via computador remoto ou tomada de decisão do usuário pelo controle *wireless*. A Central de Gerenciamento do Sistema define as entradas e a interface do sistema, ao mesmo tempo que engloba a comunicação entre os computadores e o manipulador. O desenvolvimento de *software* utiliza o sistema operacional *Linux Ubuntu 18.04* e o *framework* de desenvolvimento de robótica *ROS Melodic*.

O sistema que provê os dados para a navegação, localização, planejamento e mapeamento é o *Cartographer SLAM*. A comunicação com o manipulador é feito através de um dispositivo da *ROBOTIS* de nome *U2D2* como mostra o Anexo A.9.

A Figura 1 demonstra as divisões da arquitetura.

Figura 1: Arquitetura Geral Simplificada.



Fonte: Autoria própria.

O sistema é dividido nas seguintes funcionalidades:

- Navegação: Responsável pelo controle dos *drivers* dos motores presentes na plataforma móvel.
- Localização: Responsável por determinar a posição do sistema em relação ao mapa do ambiente de trabalho.
- Mapeamento: Responsável por construir o mapa do ambiente de trabalho, mostrando os obstáculos do mesmo.
- Percepção: Responsável pelo processamento dos dados obtidos pelas câmeras e pelo [LiDAR](#).
- Modularidade: Capacidade do sistema de integrar novos periféricos, como sensores, câmeras, manipuladores e computadores externos.
- Atuação: Responsável por designar as atividades do manipulador.
- Comunicação externa: Responsável pela comunicação do computador central localizado na plataforma móvel e dos computadores remotos que possuem os pacotes do [ROS](#).
- Planejamento: Responsável por designar trajetórias livres de obstáculos para o sistema.

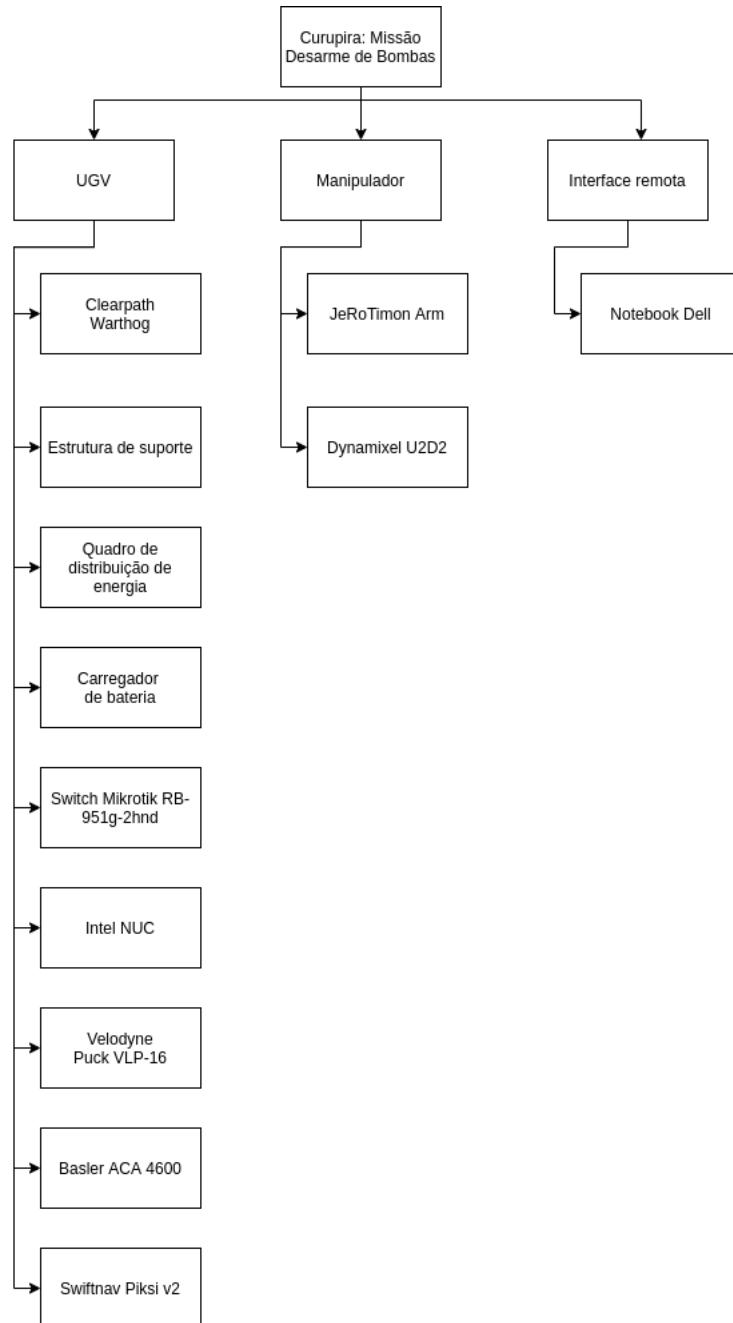


## 3 DESENVOLVIMENTO

### 3.1 Estrutura Analítica do Sistema

A *Estrutura Analítica do Projeto* (EAP), conforme visto na Figura 2, apresenta uma descrição de alto nível dos componentes que compõem o protótipo. Neste equipamento específico, o protótipo é formado fundamentalmente por um *Unmanned Ground Vehicle* (UGV), traduzido para o português como Veículo Terreste não Tripulado, um manipulador robótico e uma interface remota. O UGV representa a plataforma móvel responsável pela locomoção do protótipo, além de servir de base para todos os sensores, computadores e para o manipulador integrado. Dentre os sensores disponíveis, destacam-se a câmera RGB *Basler aca4600* e o LiDAR *Velodyne VLP-16*. A atuação sobre dispositivos externos é realizada pelo manipulador JeRoTIMON sob supervisão da “inteligência” embarcada. Por fim, a interface remota, constituída de um notebook, é responsável por fornecer em tempo real informações provenientes do robô para o usuário.

Figura 2: Estrutura Analítica do Projeto.



Fonte: Autoria própria.

### 3.2 Especificação de componentes

Os componentes principais de *hardware* selecionados para compor o sistema desenvolvido encontram-se relacionados nesta seção, juntamente com uma breve descrição de seus atributos principais.

## Plataforma móvel

A plataforma móvel escolhida para este projeto foi o [UGV](#) anfíbio *Warthog* fabricado e comercializado pela *Clearpath Robotics*, conforme visto na Figura 3. Esta plataforma possui peso bruto igual a 590 kg distribuídos ao longo dos seus 1,52 m de comprimento, 1,38 m de largura e 0,83 m de altura. O *Warthog* é caracterizado pela sua capacidade de movimentação em condições adversas em ambientes secos ou alagados, desde que o mesmo não seja completamente submerso (*IP Rating - 65*). Sua velocidade máxima é 18 km/h em terra e 4 km/h na água. Além disso, quando em operação terrestre, o mesmo suporta carga útil de até 272 kg, podendo movimentar-se em terrenos com inclinação máxima de 35 a 45 °. Suas baterias são do tipo chumbo ácido com tensão nominal de 48 V e capacidade de 105 Ah, o que lhe permite um tempo nominal de operação de 2,5 h.

Além do sistema eletro-mecânico motriz, o *Warthog* conta também com um computador de bordo configurado para operação do ambiente [ROS Kinetic](#) e um conjunto de sensores ([IMU](#), [GPS](#) e *encoders*) e atuadores (E-Stop). Informações disponíveis no Anexo [A.1](#).

Figura 3: UGV *Warthog* da *Clearpath Robotics*

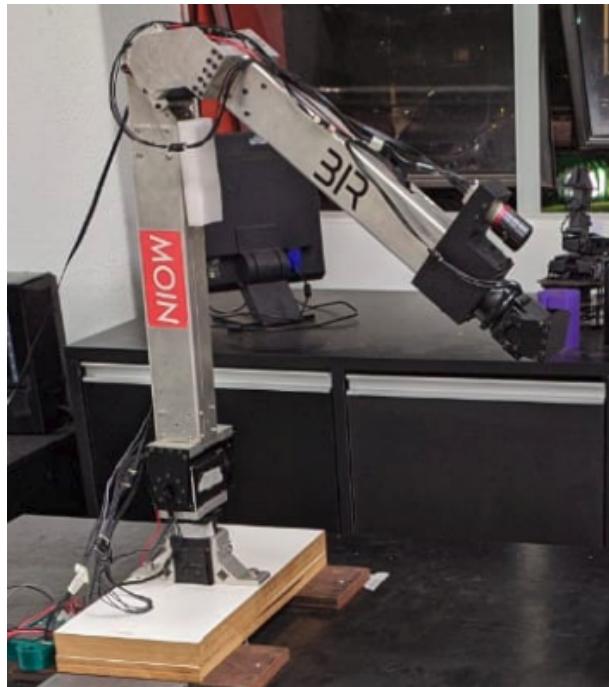


Fonte: ([Clearpath Robotics, 2016](#)).

## Manipulador robótico

Manipuladores robóticos são utilizados como atuadores capazes de interferir no ambiente onde estão inseridos de forma a manipular possíveis objetos de interesse. O manipulador escolhido denominado JeRoTIMON, Figura 4, foi desenvolvido em um projeto de pesquisa pelo grupo [CCRoSA](#). Este equipamento possui 5 graus de liberdade, é capaz de levantar até 2 kg de material e possui alcance máximo de 981 mm. Sua massa total, incluindo a base, é de 10 kg. Informações disponíveis no Anexo [A.8](#).

Figura 4: Manipulador *JeRoTimon*.



Fonte: Centro de Competência em Robótica e Sistemas Autônomos.

## LiDAR

O **LiDAR** é um dispositivo de sensoriamento óptico capaz de aferir a distância entre si e as superfícies no seu entorno, através da emissão de um feixe *laser* e consequente aferição do tempo de retorno da luz emitida. O **LiDAR** utilizado nesta aplicação foi o *Velodyne VLP-16*, Figura 5, o qual é capaz de aferir distâncias de até 100 m com acurácia típica de  $\pm 3$  cm. Este conta com ângulo de visão horizontal de  $360^\circ$  e ângulo de visão vertical de  $-15$  a  $+15^\circ$  subdividido em 16 feixes de *laser* classe 1. A frequência de amostragem do escaneamento pode ser selecionada dentro da faixa de 5 a 20 Hz. Fisicamente, o *Velodyne VLP-16* é construído como um cilindro de 103,3 mm de raio e 71,7 mm de altura com 830 g. Informações disponíveis no Anexo A.4.

Figura 5: *Lidar Velodyne VLP-16*

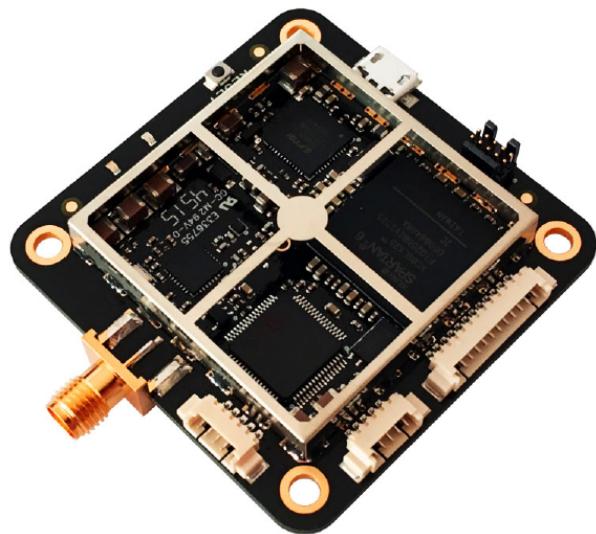


Fonte: *Velodyne Lidar*

## Sistema de georreferenciamento

O sistema de georreferenciamento fornece dados de geolocalização: latitude, longitude e altura da plataforma móvel, utilizados pelo sistema de [SLAM](#). O dispositivo utilizado foi o modelo *Piksi V2* da fabricante *Swiftnav*, Figura 6. Este módulo tem dimensões de 53 por 53 mm, não contando a antena externa utilizada, e é capaz de fornecer medidas com frequência de 10 Hz e acurácia centimétrica quando utilizado no modo [RTK](#). Informações disponíveis no Anexo A.5.

Figura 6: GPS *Piksi V2* da *Swiftnav*



Fonte: *Swiftnav*.

## Câmera

Com o intuito de realizar a detecção da “bomba”, foi utilizada uma câmera **RGB** modelo acA4600-10uc da fabricante *Basler*, como visto na Figura 7. Este dispositivo conta com resolução máxima de 4608 por 3288 *pixels* entregues através de conexão **USB** 3.0. A frequência de captura nominal é de 10 Hz, mas pode ser configurada tendo-se em vista a aplicação, resolução e largura de banda da conexão utilizada. Esta câmera possui largura igual a 29 mm, comprimento 41,3 mm e altura 29 mm, com massa inferior a 80 g. Informações disponíveis no Anexo A.6.

Figura 7: Câmera *Basler* acA4600-10uc.



Fonte: *Basler*.

## Lente

Utilizada em conjunto com a câmera *Basler acA4600* (Secção 3.2), a lente *Kowa LM8HC* permite que as imagens sejam capturadas com ângulo de visão vertical de 44,7° e horizontal de 58,3°. Esta lente está disposta na Figura 8. A distância mínima para foco é de 10 cm e o comprimento focal é de 8 mm com *F Ratio* de 1:1.4 máximo. As dimensões deste dispositivo são respectivamente 66 x 57 x 57 mm (C x L x P), contando com massa igual a 205 g. Informações disponíveis no Anexo A.7.

Figura 8: Câmera *Kowa LM8HC*.



Fonte: *Kowa*.

## Computador central

O computador de bordo escolhido para o processamento das informações provenientes dos sensores e dos algoritmos necessários para o funcionamento da aplicação foi o modelo [NUC](#) 5i5RYK fabricado pela *Intel*, Figura 9. Informações disponíveis no Anexo A.2. Suas principais configurações são:

- Processador Intel Core i5-5250U com frequência de 1.6 GHz (com possibilidade de trabalhar até 2.7 GHz) e 3 MB de memória cache;
- 16 GB de memória RAM DDR3L com frequência de 1333 MHz;
- Placa de vídeo integrada HD Intel 6000;
- 4 portas USB 3.0;
- Saída de vídeo mini HDMI;
- Saída de áudio e entrada de microfone 3.5 mm;
- Placa de rede que suporta cabeamento RJ45 com velocidade de 10/100/1000 Mbps e Wi-Fi;
- Armazenamento interno via SSD SATA *Crucial CT250MX200SSD4* de 256 GB.

Figura 9: Intel NUC5i5RYK.



Fonte: Intel.

## ***Switch***

*Switch* é um equipamento de comunicação capaz de redirecionar pacotes *Ethernet* entre dispositivos nele conectado, possibilitando a integração e comunicação entre esses dispositivos. O modelo de *switch* escolhido é fabricado pela *MikroTik*, sob especificação RB951G, como mostra a Figura 10. Este foi escolhido por apresentar, número de portas e velocidade de comunicação suficiente para a aplicação. Ele possui 5 portas de conexão com velocidades de 10 Mbps, 100 Mbps e 1000 Mbps. Suas dimensões são respectivamente 113 x 138 x 29 mm (C x L x P) com massa igual a 232 g. Informações disponíveis no Anexo A.3.

Figura 10: *Switch* RB951G, *MikroTik*.



Fonte: *MikroTik*.

### 3.3 Funcionalidades

As funcionalidades de um sistema robótico descrevem os subsistemas e a lógica de operação dos mesmos. Para o desenvolvimento da solução proposta pela equipe Curupira, foi levado em consideração oito principais funcionalidades: localização, navegação, percepção, planejamento, mapeamento, modularidade, atuação e comunicação externa.

#### Localização

A funcionalidade de localização representa a capacidade do robô definir a sua posição em um ambiente de trabalho.

A solução definida foi utilizar um sistema de georreferenciamento, tendo em vista que o ambiente de trabalho do sistema é externo. Este sistema de georreferenciamento define as posições de latitude e longitude do robô, que são corrigidas em tempo real à medida que o mesmo se move.

O dispositivo utilizado foi o modelo *Piksi V2* da fabricante *Swiftnav*. A Figura 11 representa a leitura do sensor na área externa do CIMATEC 4.

Figura 11: Leitura do GPS.

Num. sats	8
Lat	-12.937993355
Lng	-38.3884693097
Alt	31.8691921598
Flags	0x00
Mode	SPP (single point position)
Vel. N	0.0140
Vel. E	0.0100
Vel. D	0.0200
PDOP	2.4
GDOP	2.8
TDOP	1.5
HDOP	1.1
VDOP	2.1

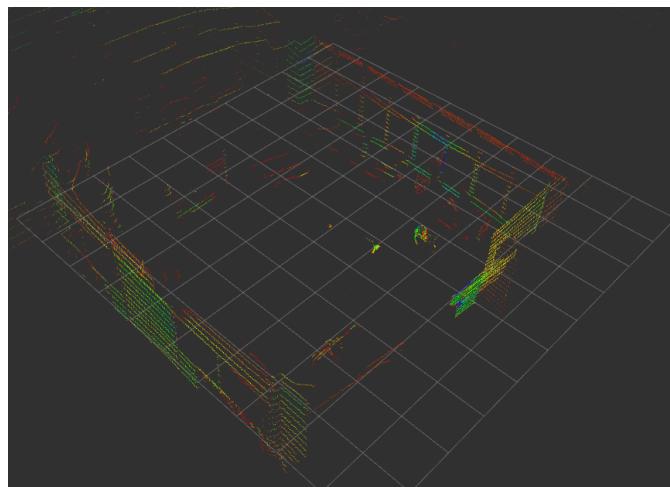
Fonte: Autoria própria. Software: *Swift Console* (*Swift Navigation*, 2020).

## Percepção

A funcionalidade de percepção compreende as saídas geradas pelos sensores integrados ao sistema. Com estas entradas, é possível realizar as tarefas de mapeamento, localização, detecção e navegação autônoma. A qualidade dos sensores é imprescindível para que as tomadas de decisão realizada pelo robô não acarretem falhas.

As principais entradas da funcionalidade são nuvens de pontos para detecção de objetos e obstáculos (Figura 12), georreferenciamento para localização e imagem RGB para identificação do objetivo final (Figura 13).

Figura 12: Leitura do Velodyne VLP-16.



Fonte: Autoria própria.

Figura 13: Identificação da “bomba” utilizando câmera *Basler*.



Fonte: Autoria própria.

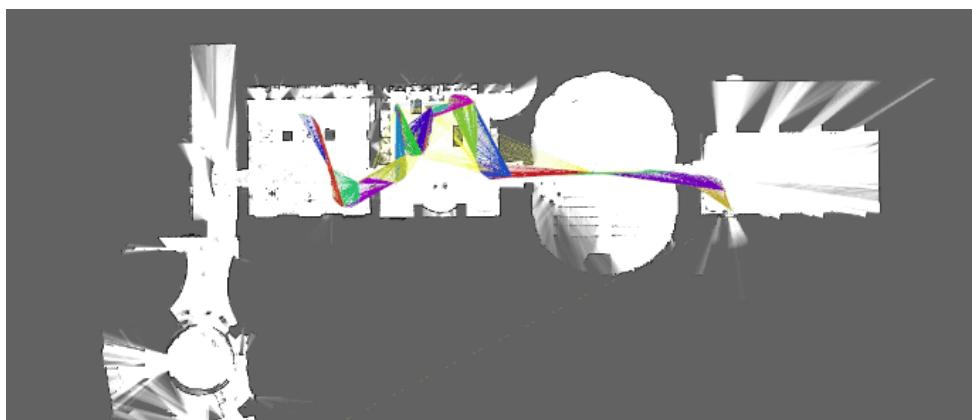
## Mapeamento

A funcionalidade de mapeamento define a capacidade de um robô encontrar os obstáculos no ambiente que está inserido. Essa definição é extremamente importante pois ela influencia diretamente na capacidade de movimentação da plataforma móvel no ambiente.

O *software* que trás as definições para essa funcionalidade é o *Cartographer*. Ele possui um pacote integrado ao *framework ROS* e tem como objetivo a construções de mapas 2D ou 3D utilizando diferentes tipos de sensores perceptivos ( [The Cartographer Authors, 2020](#); [HESS et al., 2016](#)).

O mapeamento teve como principais entradas a nuvem de pontos gerada pelo *LiDAR Velodyne VLP-16* e a localização determinada pelo *GPS Swiftnav Piksi V2*, porém sensores adicionais como *IMU*, odometria e outras nuvens de pontos poderiam melhorar sua performance. Um exemplo de como o algoritmo funciona pode ser visto na Figura 14.

Figura 14: Mapeamento exemplo - *Cartographer*.



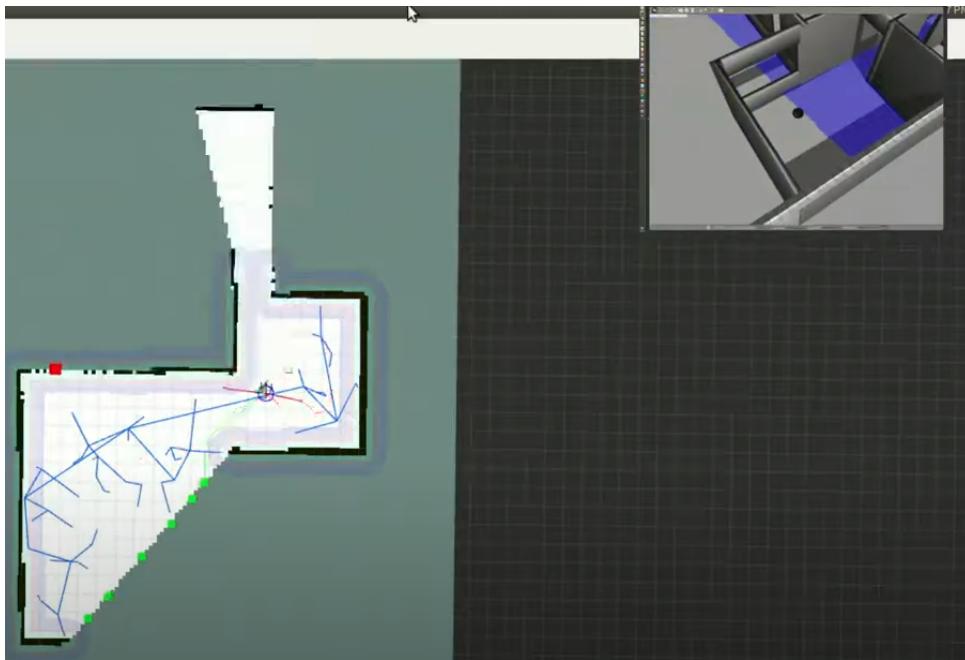
Fonte: ( [The Cartographer Authors, 2020](#)).

## Planejamento

O planejamento é uma funcionalidade que garante que o robô possa fazer uma rota viável. Tendo em vista que o sistema trabalhará em um ambiente com objetos estáticos, sua principal tomada de ação após a detecção de obstáculos é realizar o desvio ou transposição dos mesmos, mantendo o objetivo final.

O *software* por trás desta funcionalidade é o *RRT Exploration* ([UMARI, 2018](#)). Ele é um pactote *ROS* que implementa exploração em robôs móveis. Seu requerimento é a publicação constante de um mapa, que é proveniente da funcionalidade de mapeamento ([UMARI; MUKHOPADHYAY, 2017](#)). Um exemplo de como o algoritmo funciona pode ser visto na Figura 15.

Figura 15: Exemplo de exploração - *RRT*.



Fonte: (UMARI; MUKHOPADHYAY, 2017).

## Navegação

A funcionalidade de navegação é a responsável pela movimentação da base móvel *Clearpath Warthog* ([Clearpath Robotics, 2016](#)). A integração das funcionalidades de percepção, mapeamento e planejamento permite que o robô se mova de forma autônoma pela área externa do CIMATEC 4.

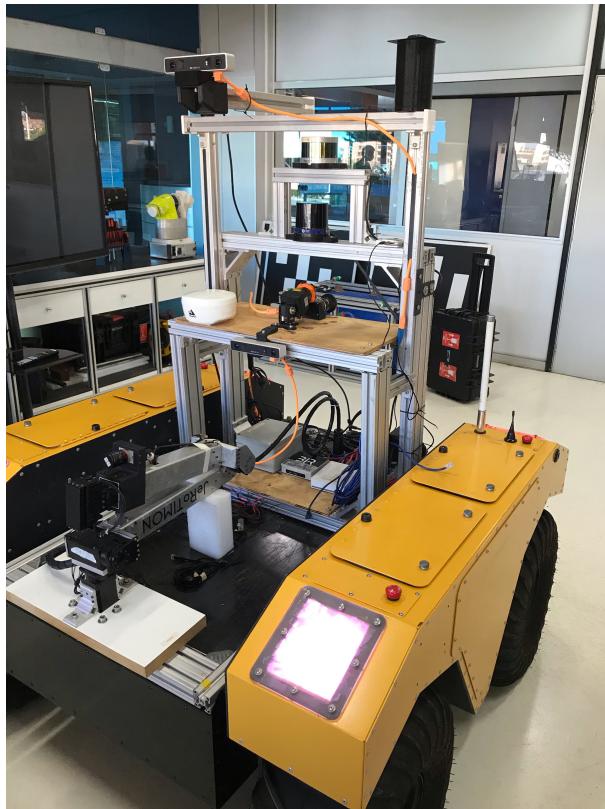
Também existe a possibilidade de movimentação via controle remoto, caso o robô tenha esteja em teste ou tenha cometido alguma falha, necessitando do comando de um operador externo.

## Modularidade

A modularidade define a capacidade do sistema de adicionar novos módulos. Estes novos módulos implicam em novas combinações de sensores e atuadores, podendo melhorar sua tomada de decisão e adicionando novas aplicações para o sistema integrado.

A modularidade é dividida em três subsistemas. O subsistema mecânico engloba a parte estrutural do suporte e a sua conexão com a plataforma móvel, onde os sensores são fixados (como mostra Figura 16 e mais detalhadamente no Apêndice A). O subsistema de controle é definido pela obtenção de dados dos diversos sensores conectados ao computador central via *switch*. Por fim, existe o subsistema de potência, onde os módulos e sensores são energizados dependendo de suas necessidades, como visto no Apêndice B.

Figura 16: Suporte mecânico e sensores integrados.



Fonte: Autoria própria.

## Atuação

Para o objetivo final, é necessário que haja interação entre o manipulador robótico e a “bomba”. A funcionalidade de atuação envia os comandos necessários para que o manipulador JeRoTIMON tenha contato com os fios e retorne para a posição inicial, finalizando o processo.

## Comunicação externa

A funcionalidade de comunicação externa define os dispositivos de controle que estão conectados ao computador central. O computador central que está localizado no *Clearpath Warthog* se comunica com o computador localizado no suporte mecânico, que pode ser conectado na mesma rede por qualquer outro computador via *Secure Socket Shell (SSH)*.



## 4 SIMULAÇÃO E INTEGRAÇÃO

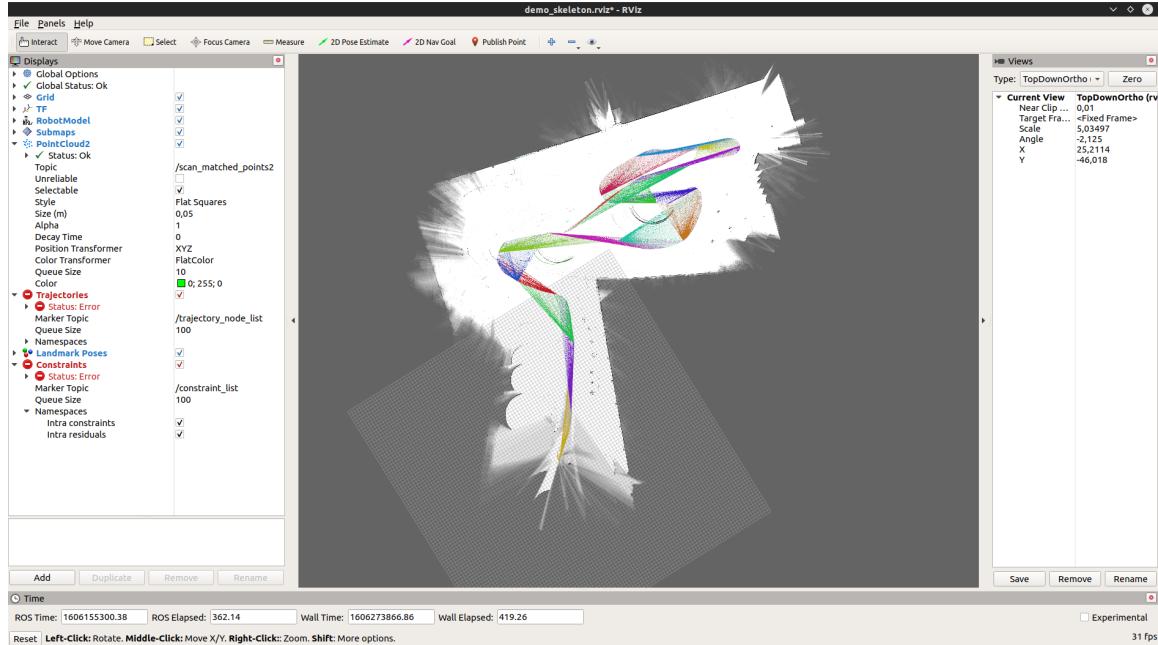
Durante o desenvolvimento deste trabalho, os códigos, *scripts* e arquivos de configuração foram criados, modificados e ajustados em duas fases parcialmente distintas denominadas simulação e integração. Durante o período de simulação, todos os códigos e arquivos de configuração necessários foram executados no *framework ROS Melodic* através do ambiente de simulação Gazebo 9. Em contrapartida, durante a integração dos equipamentos reais, tais códigos e arquivos de configuração, exceto os referentes à detecção da “bomba”, foram executados utilizando um par de computadores conectados em rede. O primeiro destes computadores está localizado dentro do *Warthog* e utiliza o *ROS Kinetic*, sendo este responsável por executar o *ros master node*. Por sua vez, o segundo computador trata-se da Intel NUC (Seção 3.2) e utiliza a versão *Melodic* deste *framework*.

Neste capítulo serão apresentados os resultados e desafios encontrados na aplicação dos componentes anteriormente descritos para a execução das funcionalidades desejadas nos dois períodos do desenvolvimento.

### 4.1 Simulação

A fase de simulação teve como objetivo principal validar a funcionalidade dos algoritmos e sensores escolhidos para a execução da missão de “desarme da bomba”, bem como, iniciar o processo de compreensão do funcionamento dos mesmos e dos ajustes necessários para o funcionamento destes de forma individual e em conjunto. A Figura 17 representa uma série de testes realizados com o intuito de ajustar os parâmetros de configuração do *software Cartographer* de forma a executar o mapeamento e localização da plataforma móvel em ambiente simulado. Neste teste em específico, apenas o *LiDAR* estava ativo, de forma a simular a inexistência efetiva dos sensores *IMU*, *GPS* e odometria da plataforma móvel real devido às dificuldades de utilizá-los com o *Cartographer*. O resultado apresentado nesta figura indica a necessidade da utilização de outros sensores durante o processo de *SLAM*, uma vez que durante rotações da plataforma móvel, pode ocorrer uma falha na comparação entre as *point clouds* capturadas com consequente falha na estimativa correta da localização do robô e na geração do mapa do ambiente.

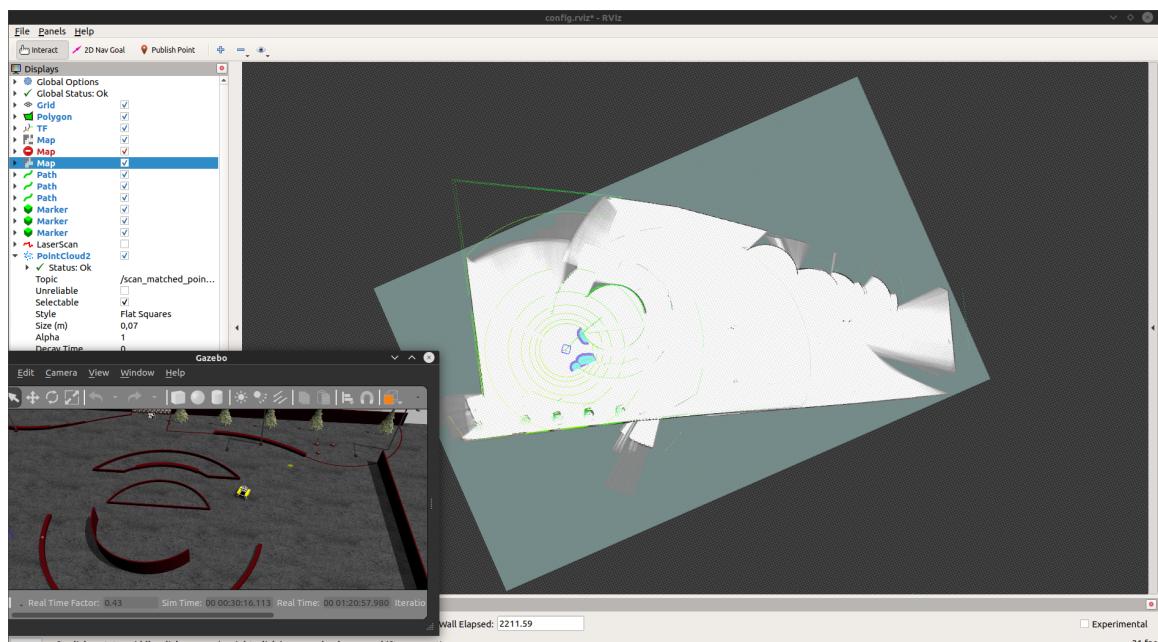
Figura 17: Resultado de mapeamento errôneo com o *Google Cartographer*.



Fonte: Autoria própria.

Desta forma, em outro momento, já com a utilização do pacote *move\_base* configurado para a plataforma, e tendo-se um melhor entendimento dos parâmetros de funcionamento dos algoritmos utilizados, foram simuladas a utilização do **LiDAR**, **GPS**, **IMU** e odometria como fontes de informação simultâneas para o *Cartographer*, como demonstra a Figura 18.

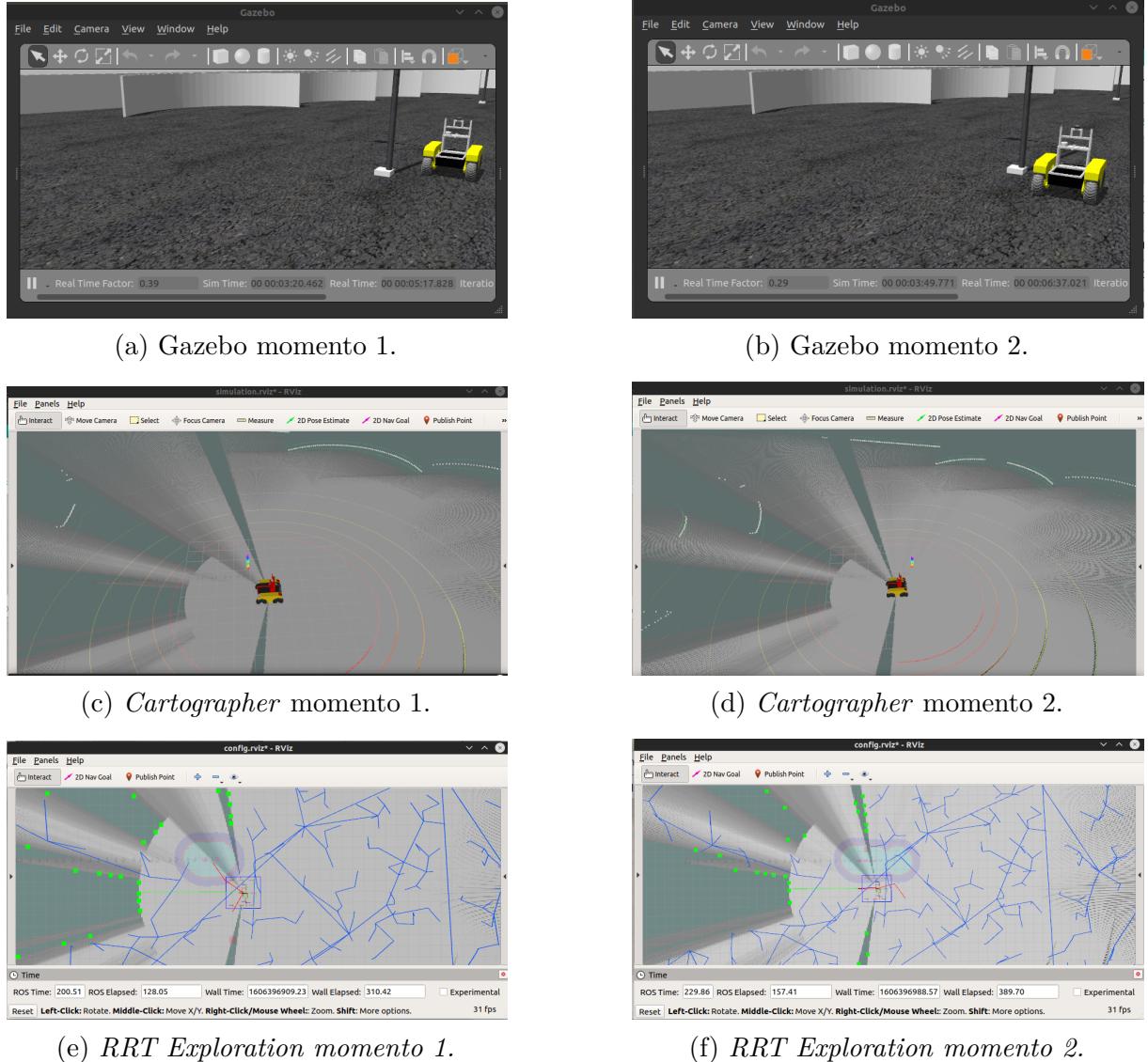
Figura 18: Resultado de mapeamento correto com o *Google Cartographer*.



Fonte: Autoria própria.

A Figura 19 mostra dois momentos distintos de execução do conjunto dos pacotes responsáveis pela autonomia da plataforma móvel: *move\_base*, *Cartographer* e *RRT Exploration*. A análise destas imagens mostra que a plataforma é capaz de desviar dos obstáculos existentes na simulação durante o processo de exploração para o mapeamento do ambiente.

Figura 19: Exploração, localização e mapeamento autônomos.



Fonte: Autoria própria.

Por fim, a última funcionalidade testada em ambiente simulado foi a modificação do algoritmo de controle do manipulador JeRoTIMON, como mostra a Figura 20. Este algoritmo foi atualizado referente ao manipulador ElBorgson, manipulador robótico desenvolvido pelos pesquisadores do CCRoSA. Essa modificação buscou organizar e estruturar o código, além de modificar o serviço de interface para controle do manipulador JeRoTIMON.

Figura 20: Simulação de manipulação.



Fonte: Autoria própria.

## 4.2 Integração

Modificações foram realizadas sobre a plataforma móvel *Warthog*, de forma que a mesma viesse a transportar todos os componentes necessários para o desenvolvimento deste e de outros projetos, visando também o emprego de materiais existentes e a praticidade de sua construção, operação e manutenção. O protótipo utilizado encontra-se disposto na Figura 21.

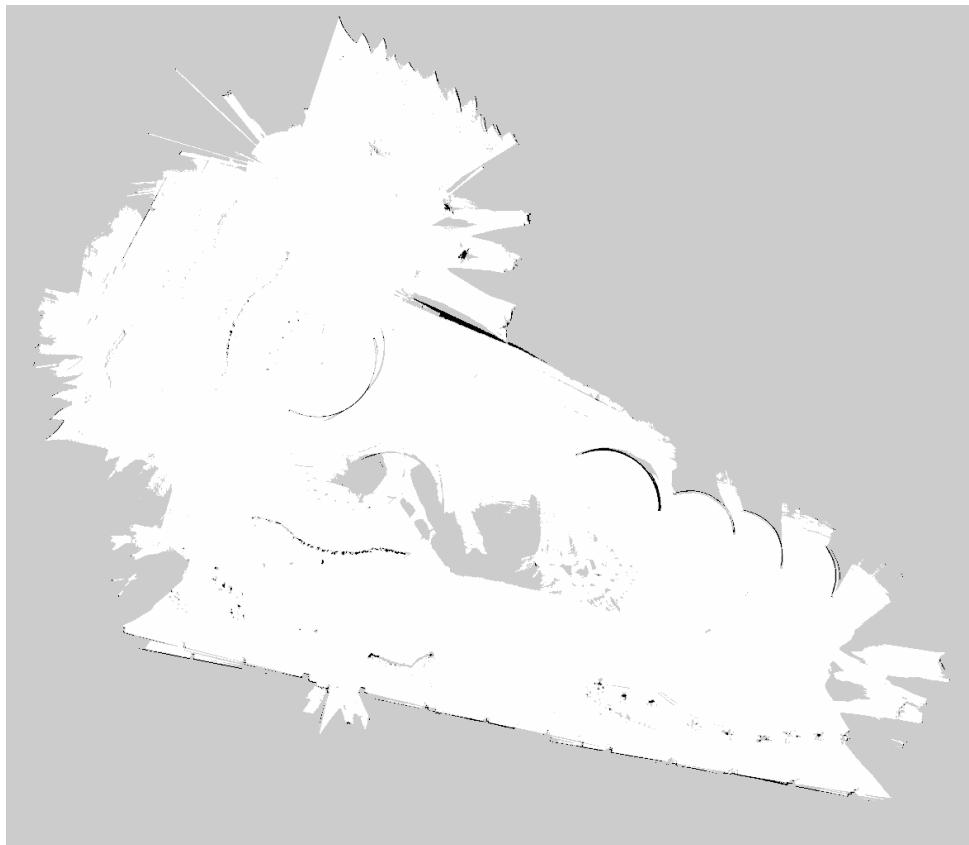
Figura 21: Integração do protótipo.



Fonte: Autoria própria.

O planejamento inicial deste projeto propôs a utilização do *LiDAR Velodyne VLP-16* e da câmera *MyntEye S1030* como sensores principais, com possibilidade de uso da **IMU**, **GPS** e odometria integrados no *Warthog* como auxílio à navegação. Contudo, não foi possível a utilização de nenhum sensor integrado ao *Warthog*, uma vez que a integração de qualquer combinação deles com o **LiDAR** no *Cartographer* produz uma série de erros e falhas de operação deste último. Investigações e tentativas de resolução destes problemas foram executadas sem resultados definitivos. A solução encontrada foi a substituição dos sensores integrados por outros conectados diretamente à **NUC** responsável por executar o *Cartographer*. A Figura 22 apresenta o mapa gerado pelo *Cartographer* através do uso do **LiDAR** *Velodyne VLP-16* em conjunto com o **GPS** *Swiftnav Piksi V2*, sob controle teleoperado. Tentativas de integrar a odometria fornecida por um par de sensores *Swiftnav Piksi V2* configurados para operação **RTK** foram iniciadas, mas não finalizadas devido ao tempo disponível.

Figura 22: Resultado de mapeamento com o *Google Cartographer* em ambiente real.



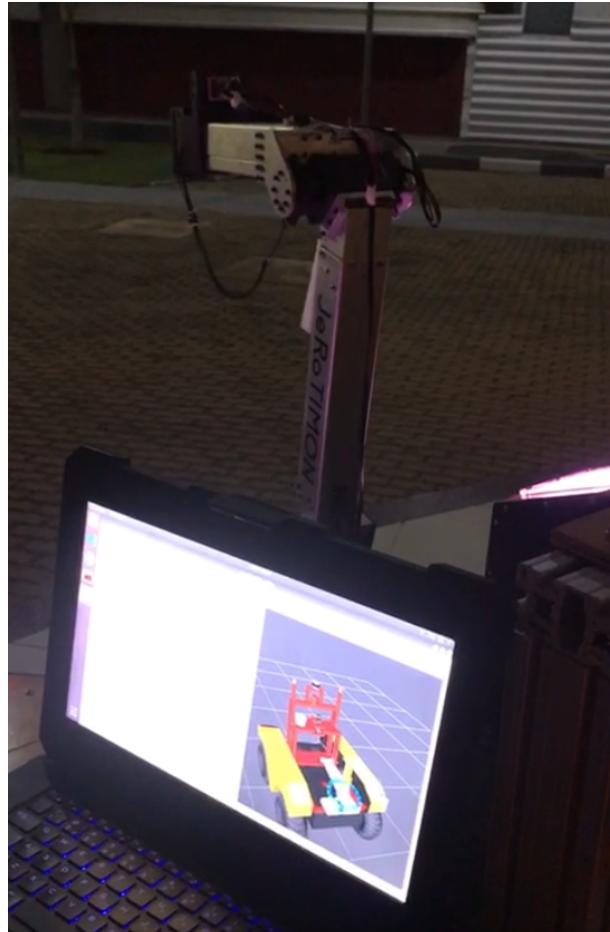
Fonte: Autoria própria.

Os algoritmos utilizados para operação autônoma, *move\_base* e *RRT Exploration*, foram configurados e estão operacionais. Contudo há necessidade da realização de testes mais exaustivos para garantir o seu correto funcionamento.

Por sua vez, a integração do manipulador JeRoTIMON transcorreu de forma tranquila

e com o resultado esperado, disposto na Figura 23. Recomenda-se que sejam realizados mais testes relacionados aos planos de colisão, a fim de se obter uma garantia maior sobre sua segurança operacional.

Figura 23: Integração da manipulação.

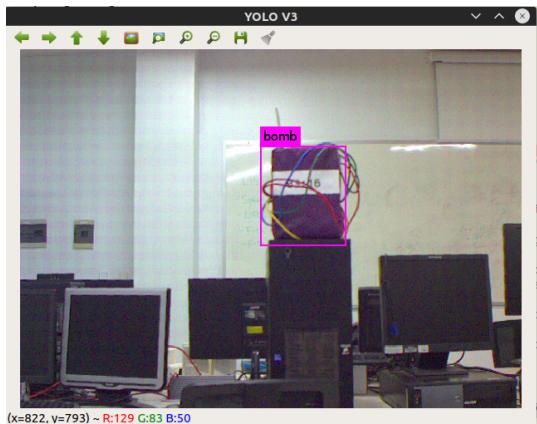


Fonte: Autoria própria.

Finalmente, a última funcionalidade construída foi a detecção da “bomba”. Para tanto, foi treinada uma rede neural tipo YOLOv3-tiny, que é baseada na rede YOLOv3, (REDMON; FARHADI, 2018) através do uso da plataforma *Google Colab PRO* e dos *scripts Python* (theAIGuysCode, 2020). O treinamento foi realizado utilizando 341 imagens adquiridas da “bomba” usando a câmera *Basler aca4600*, que foi escolhida por ter um pacote **ROS** com menor custo de processamento na *Central Processing Unit (CPU)*, em relação ao da câmera *MyntEye S1030*.

Após o treinamento, a rede foi executada utilizando o pacote **ROS darknet\_ros** (BJE-LONIC, 2020). Exemplos dos resultados obtidos podem ser vistos nas Figuras 24a e 24b.

Figura 24: Resultado da detecção da “bomba”.



(a) Ambiente fechado.



(b) Ambiente aberto.

Fonte: Autores.

Cabe ressaltar, que o pacote *darknet\_ros* utiliza o *framework CUDA* para aceleração de sua execução sob presença de *Graphic Processing Unit* (*GPU*)s NVidia. Nos testes realizados, a quantidade de *Frames Per Second* (*FPS*) processados pela rede neural saltou de  $\approx 2$  *FPS* para  $\approx 70$  *FPS* ao utilizar uma *GPU* NVidia Quadro P2000 Mobile em conjunto com a *CPU* Intel Xeon E-2176M original.

Até o presente momento, não foi possível finalizar a integração completa das funcionalidades construídas com o intuito de produzir uma plataforma móvel capas de “desarmar a bomba” de forma completamente autônoma. Trabalhos futuros serão realizados para completar a integração deste projeto.



## 5 GESTÃO DO CONHECIMENTO

O objetivo principal desta metodologia é auxiliar a busca do conhecimento organizacional de forma eficiente. Para isso, é necessário reunir as seguintes funções: organizar, disseminar, avaliar, mensurar e criar os inúmeros conhecimentos organizacionais concebidos durante o desenvolvimento do projeto ([STRAUHS et al., 2012](#)).

O presente capítulo tem por objetivo apresentar as lições aprendidas que surgiram durante cada etapa do projeto, ou seja, partiu da fase inicial do planejamento até a etapa de construção do modelo real. A metodologia utilizada para o desenvolvimento das lições aprendidas foi criar uma comparação entre o que era esperado e o que realmente aconteceu em cada uma das etapas do projeto. Na seção a seguir serão descritas as lições aprendidas durante a concepção da solução do desafio proposto.

### 5.1 Lições aprendidas

Entende-se lições aprendidas como sendo um conjunto de aprendizagens seja ela positiva ou negativa adquiridas durante todo o processo de desenvolvimento do projeto ([PMBOK, 2008](#)). Esta atividade de coletar e analisar as lições aprendidas é considerada uma forma de transferir todo o conhecimento gerado durante a fase de execução de um projeto e com isso torna-se possível auxiliar na elaboração de projetos futuros ([VERONESE, 2014](#)).

Durante o processo de elaboração das lições aprendidas foi de suma importância o envolvimento de todos os participantes envolvidos neste projeto. A Tabela 1 mostra como foi estruturada cada lição aprendida abordando os seguintes aspectos: tema, fase, impacto, o que ocorreu?, como resolveu?, resultados e recomendações para próximos projetos.

Tabela 1: Lições aprendidas.

Tema	Fase	Impacto	O que ocorreu?	Como resolveu?	Resultados	Recomendações para os próximos objetos
Gestão	Planejamento	Negativo	Não conseguiu prever as dificuldade do projeto	Refazendo o cronograma	Melhor divisão das atividades	Discussão de todas as atividades durante a etapa do planejamento
Gestão	Planejamento	Negativo	Não houve tempo hábil para realização das atividades	Aumento na carga horária	Realização das tarefas previstas no cronograma	Prever dificuldades nas atividades e designar tempo extra para realização das mesmas.
Tecnológico	Execução	Negativo	Representação dos sensores na simulação não condizem com a realidade	Diminuir os esforços nas atividades de simulação	Adequação dos sensores para atividade com base nos testes de campo.	Agilizar o processo de verificação da proximidade da simulação com a realidade e agilizar a escolha de prioridades.
Tecnológico	Execução	Negativo	Insistir em soluções falhas	Buscar soluções alternativas	Chegou em uma solução adequada	Dividir o grupo de forma a corrigir a falha e simultaneamente procurar solução alternativa
Gestão	Execução	Positivo	Buscar alcançar o resultado final em detrimento de resultados parciais não estritamente necessários	Investir os esforços no robô físico	Antecipou o desenvolvimento do objetivo final	Análise cautelosa sobre o posicionamento dos sensores antes de instalá-los

Fonte: Autoria própria.



## 6 CONCLUSÃO

O uso de plataformas robóticas móveis tem se mostrado promissor na execução de diversas atividades, principalmente àquelas onde a saúde humana é posta em risco. Um exemplo típico ocorre no manuseio e descarte de materiais explosivos, os quais são realizados atualmente por robôs teleoperados. Avanços tecnológicos são necessários para a correta aplicação de autonomia em tais sistemas. Este projeto visa estudar a aplicação de um robô móvel autônomo empregado no “desarme de bombas”.

Desta forma, o objetivo principal deste projeto é a integração de uma plataforma móvel com um manipulador robótico com o intuito de resolver o problema proposto. Além disso, faz-se necessário acoplar um conjunto de sensores capazes de fornecer informações do ambiente aonde será inserido o robô para que o mesmo funcione de maneira adequada. A solução alcançada pelo grupo apresenta algumas deficiências momentâneas que podem ser resolvidas em desenvolvimentos futuros.

Uma das deficiências existentes é a falta de validação e verificação em longo prazo do correto funcionamento do *software Cartographer SLAM* com os diversos tipos de sensores utilizados. Até o momento, a utilização do *LiDAR* juntamente com o *GPS* obteve os resultados esperados. Contudo, acredita-se que a falta de um dispositivo *IMU* e de informações da odometria diminuam a eficiência e a robustez das funcionalidades de localização e mapeamento. Esta hipótese é derivada dos melhores resultados provenientes do ambiente simulado, aonde se faz uso do *Cartographer* com todos os sensores mencionados anteriormente, o que não ocorre na realidade.

Este projeto funciona como uma prova de conceito de que é possível implementar plataformas móveis de maneira autônoma com as devidas funcionalidades de percepção, localização e mapeamento.

A integração demonstrada nesse relatório continuará sendo testada pela equipe de orientadores e bolsistas do *CCRoSA* e é esperado o cumprimento do objetivo final.



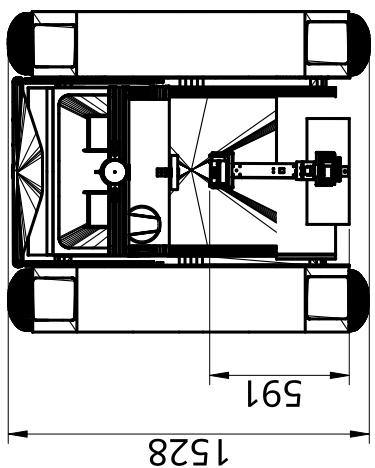
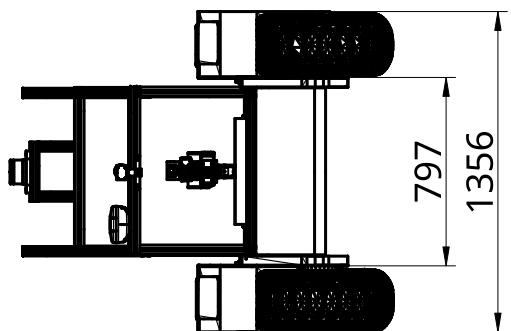
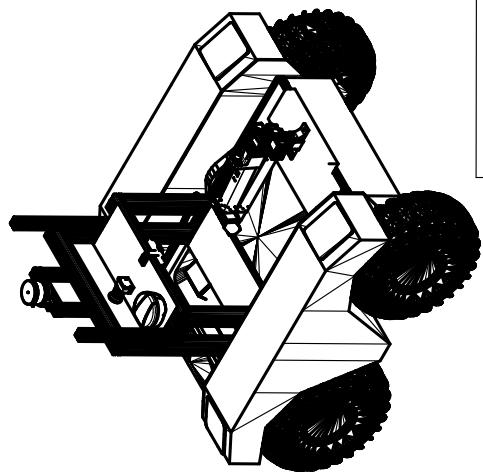
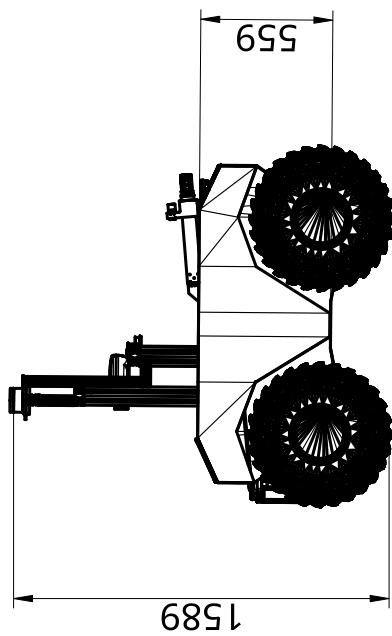
## REFERÊNCIAS

- The Cartographer Authors. *Cartographer ROS Integration*. 2020. <<https:////google-cartographer-ros.readthedocs.io/en/latest/>>. Acessed: 2020-12-08. Citado na página 29.
- BJELOVIC, M. *YOLO ROS: Real-Time Object Detection for ROS*. 2020. <[https://github.com/leggedrobotics/darknet\\_ros](https://github.com/leggedrobotics/darknet_ros)>. Acessed: 2020-12-08. Citado na página 38.
- Clearpath Robotics. *WARTHOG - AMPHIBIOUS UNMANNED GROUND VEHICLE*. 2016. <<https:////clearpathrobotics.com/warthog-unmanned-ground-vehicle-robot/>>. Acessed: 2020-12-08. Citado 2 vezes nas páginas 21 e 30.
- HESS, W. et al. Real-time loop closure in 2d lidar slam. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.: s.n.], 2016. p. 1271–1278. Citado na página 29.
- NUNES, L. *QFD (Quality Function Deployment): o que é e para que serve?* 2019. <<https:////www.voitto.com.br/blog/artigo/qfd>>. Acessed: 2020-12-06. Citado na página 14.
- PMBOK, U. G. d. C. d. *Conhecimentos em Gerenciamentos de Projetos: Guia PMBOK®*. [S.l.]: Pennsylvania: Four Campus Boulevard, 2008. Citado na página 41.
- REDMON, J.; FARHADI, A. Yolov3: An incremental improvement. *arXiv*, 2018. Citado na página 38.
- STRAUHS, F. d. R. et al. *Gestão do conhecimento nas organizações*. [S.l.]: Aymará Educação, 2012. Citado na página 41.
- Swift Navigation. *RESOURCE LIBRARY*. 2020. <<https:////www.swiftnav.com/resource-library>>. Acessed: 2020-12-08. Citado na página 27.
- theAIGuysCode. *YOLOv4 Training Tutorial*. 2020. <[https:////github.com/theAIGuysCode/YOLOv4-Cloud-Tutorial/blob/master/YOLOv4\\_Training\\_Tutorial.ipynb](https:////github.com/theAIGuysCode/YOLOv4-Cloud-Tutorial/blob/master/YOLOv4_Training_Tutorial.ipynb)>. Acessed: 2020-12-08. Citado na página 38.
- UMARI, H. *rrt exploration*. 2018. <<https:////bit.ly/36WksLM>>. Acessed: 2020-12-08. Citado na página 29.
- UMARI, H.; MUKHOPADHYAY, S. Autonomous robotic exploration based on multiple rapidly-exploring randomized trees. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.: s.n.], 2017. p. 1396–1402. Citado 2 vezes nas páginas 29 e 30.
- VERONESE, G. S. Métodos para captura de lições aprendidas: em direção a melhoria contínua na gestão de projetos. *Revista de Gestão e Projetos*, v. 5, n. 1, p. 71–83, 2014. Citado na página 41.

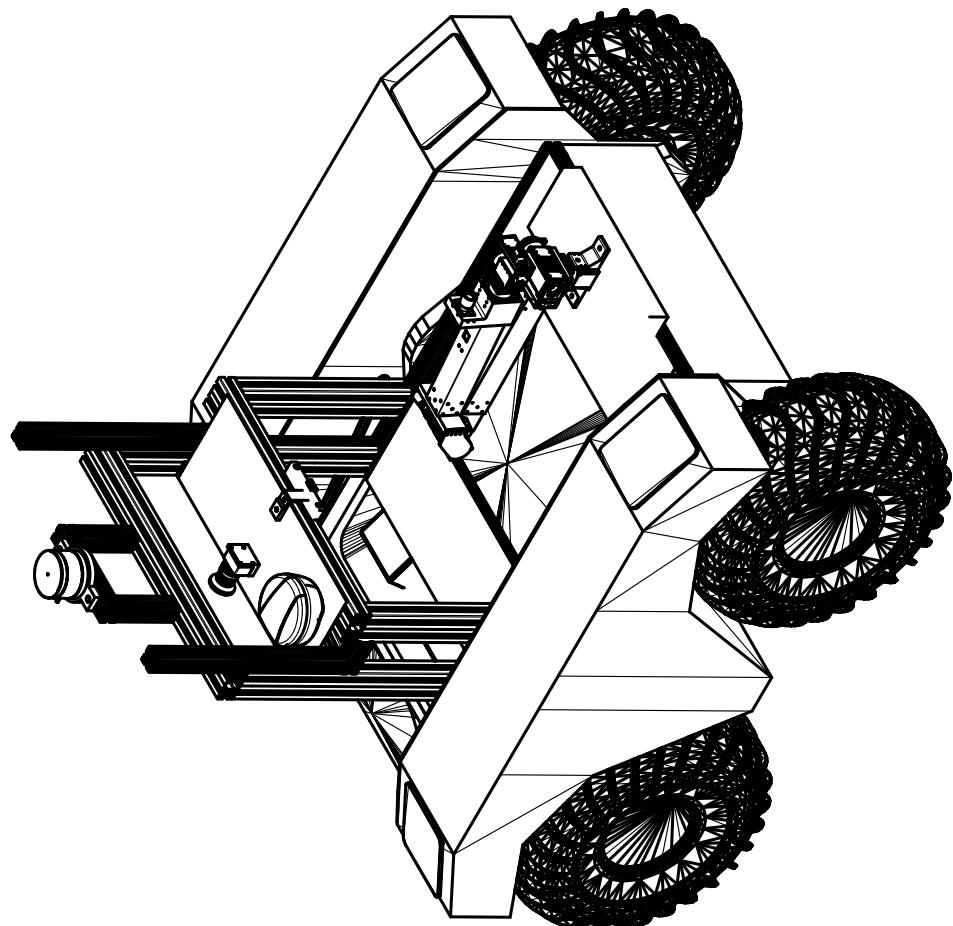


## APÊNDICE A

### Desenho mecânico



<b>Sistema FIEB</b>		<b>SENAI CIMATEC</b>	
PELO FUTURO DA INOVAÇÃO			
<b>Curupira</b>			
TITLE			
NAME	SIGNATURE	DATE	
DRAWN	VINICIUS FELISMINO	2020-11-17	DWG NO.
CHECKED	MARCO REIS	2020-06-29	A4
APPROVED	REBECA TOURNINHO	2020-06-29	1:40
		SCALE	UNIT
		---	mm
			SHEET
			1 of 2



<b>Sistema FIEB</b>		<b>SENAI CIMATEC</b>		PELO FUTURO DA INOVAÇÃO	
TITLE		Curupira			
DRAWN	NAME Vinicio Fellsmirno	SIGNATURE	DATE 2020-11-17	SIZE <b>A4</b>	DWG NO.
CHECKED	MARCO REIS		2020-06-29		1
APPROVED	REBECA TOURINHO		2020-06-29	SCALE 1:16	WEIGHT ---
				UNIT mm	SHEET 2 of 2

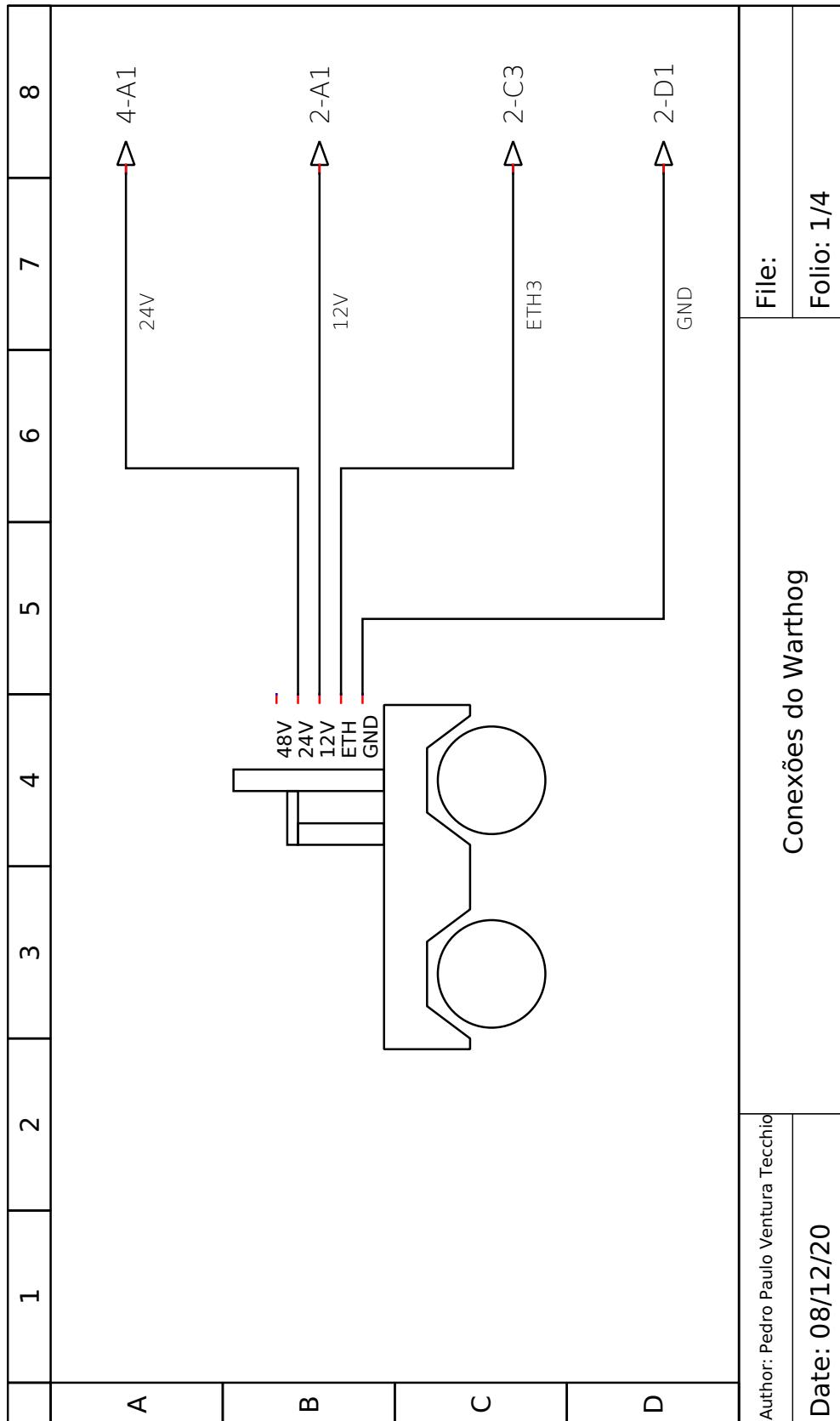


## APÊNDICE B

### Esquema elétrico de potência

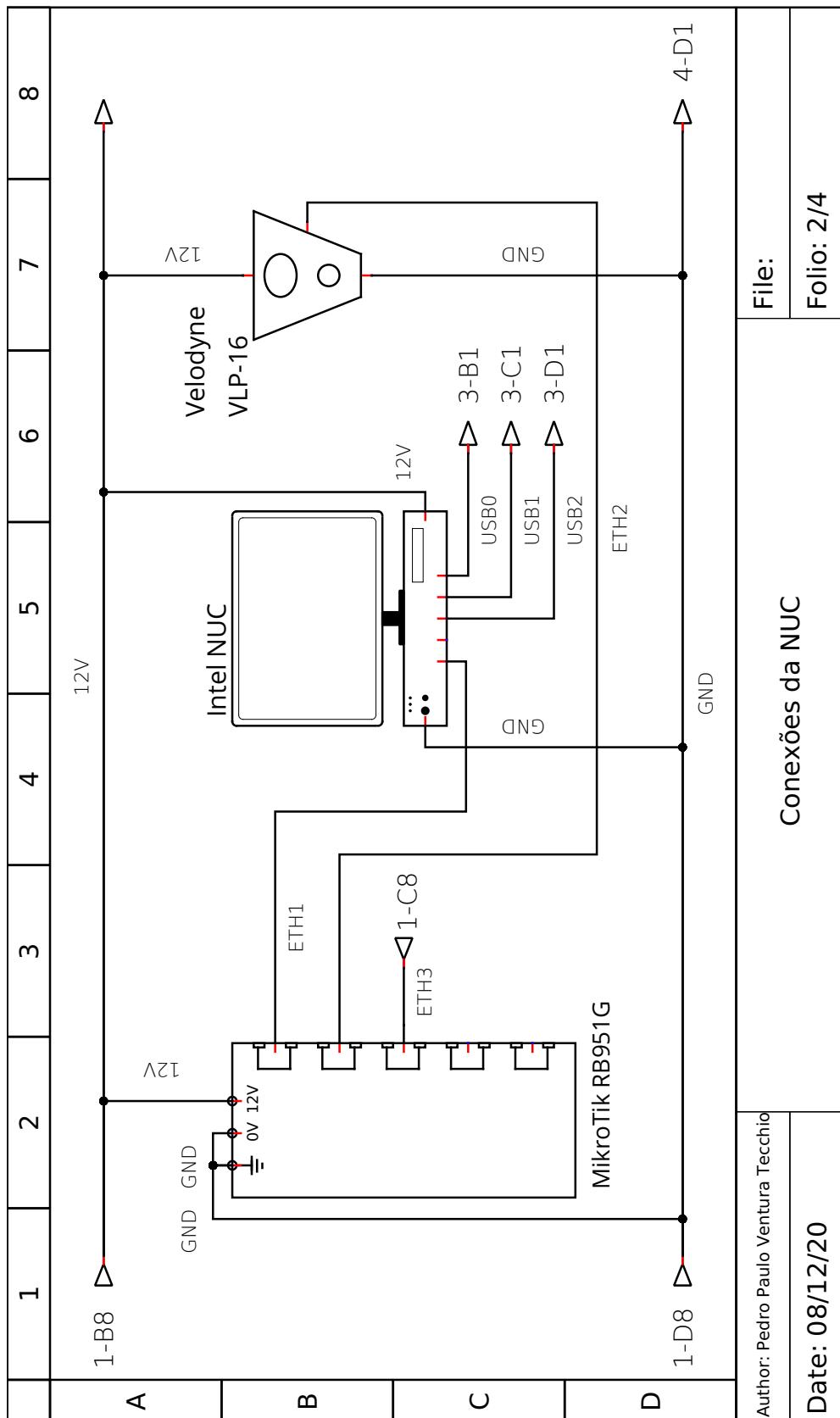
O esquema elétrico do sistema desenvolvido e testado esta disposto nas Figuras 25, 26, 27 e 28 respectivamente.

Figura 25: Esquema elétrico das conexões do Warthog.



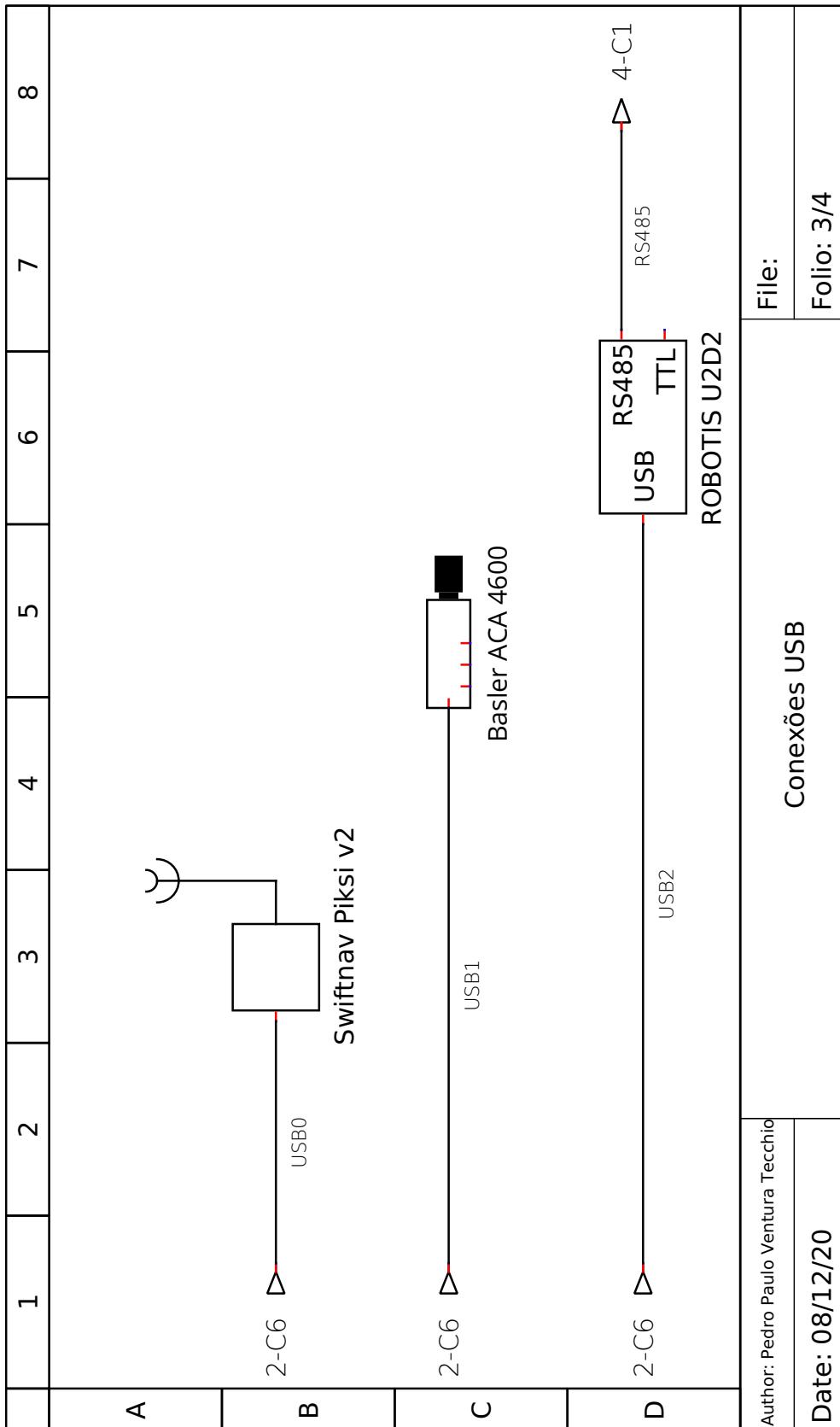
Fonte: Autores.

Figura 26: Esquema elétrico das conexões da NUC.



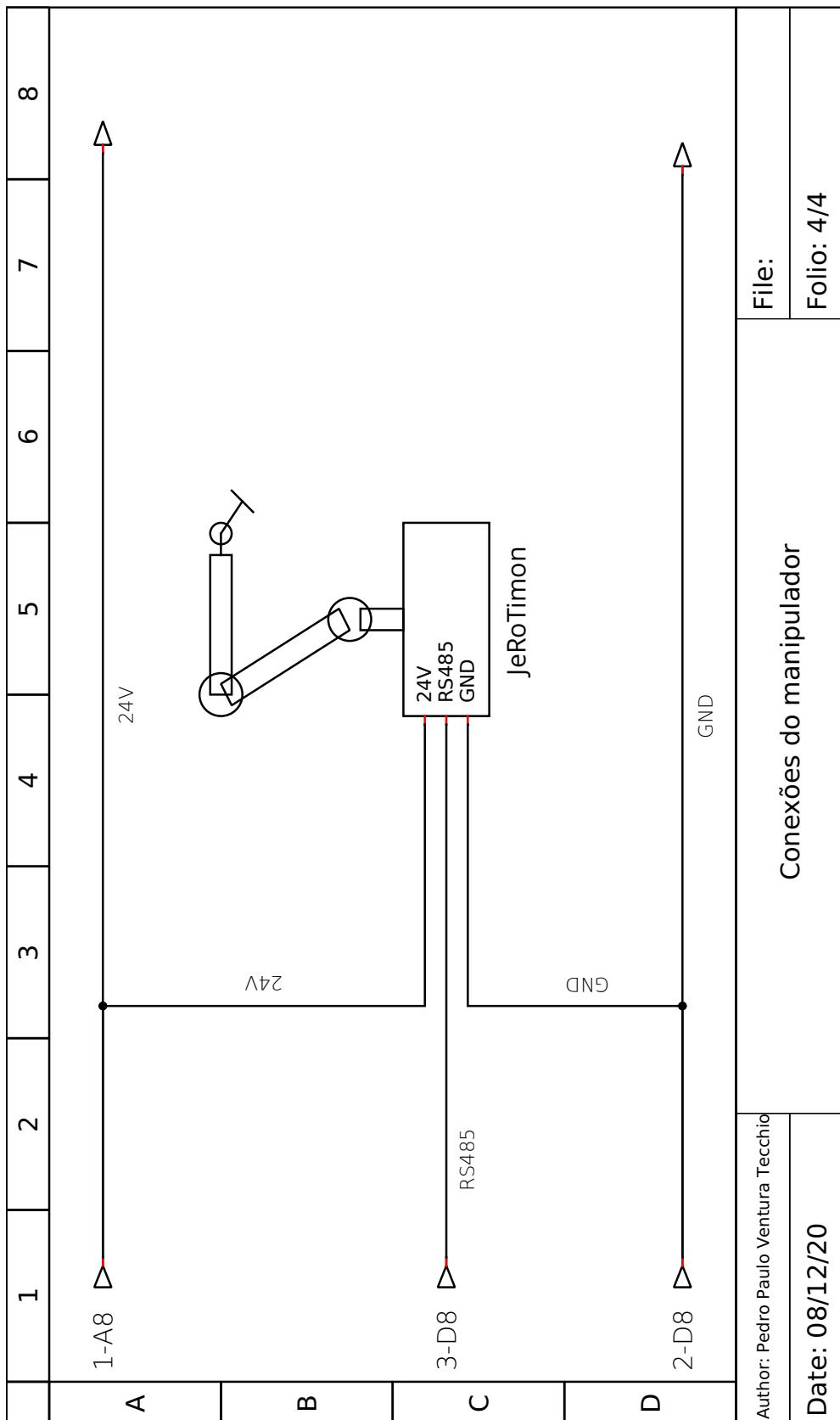
Fonte: Autores.

Figura 27: Esquema elétrico das conexões USB.



Fonte: Autores.

Figura 28: Esquema elétrico das conexões do manipulador.



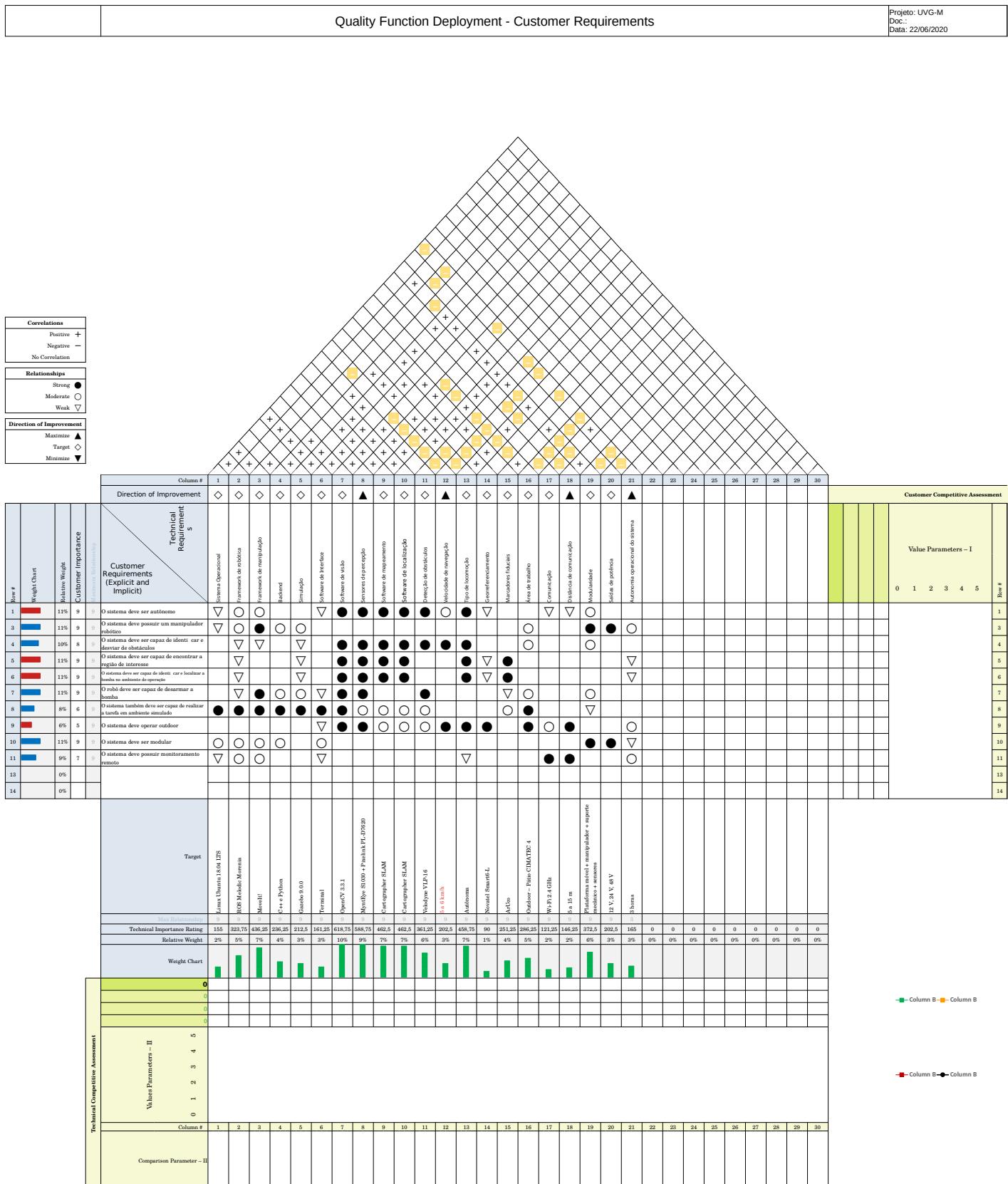
Fonte: Autores.

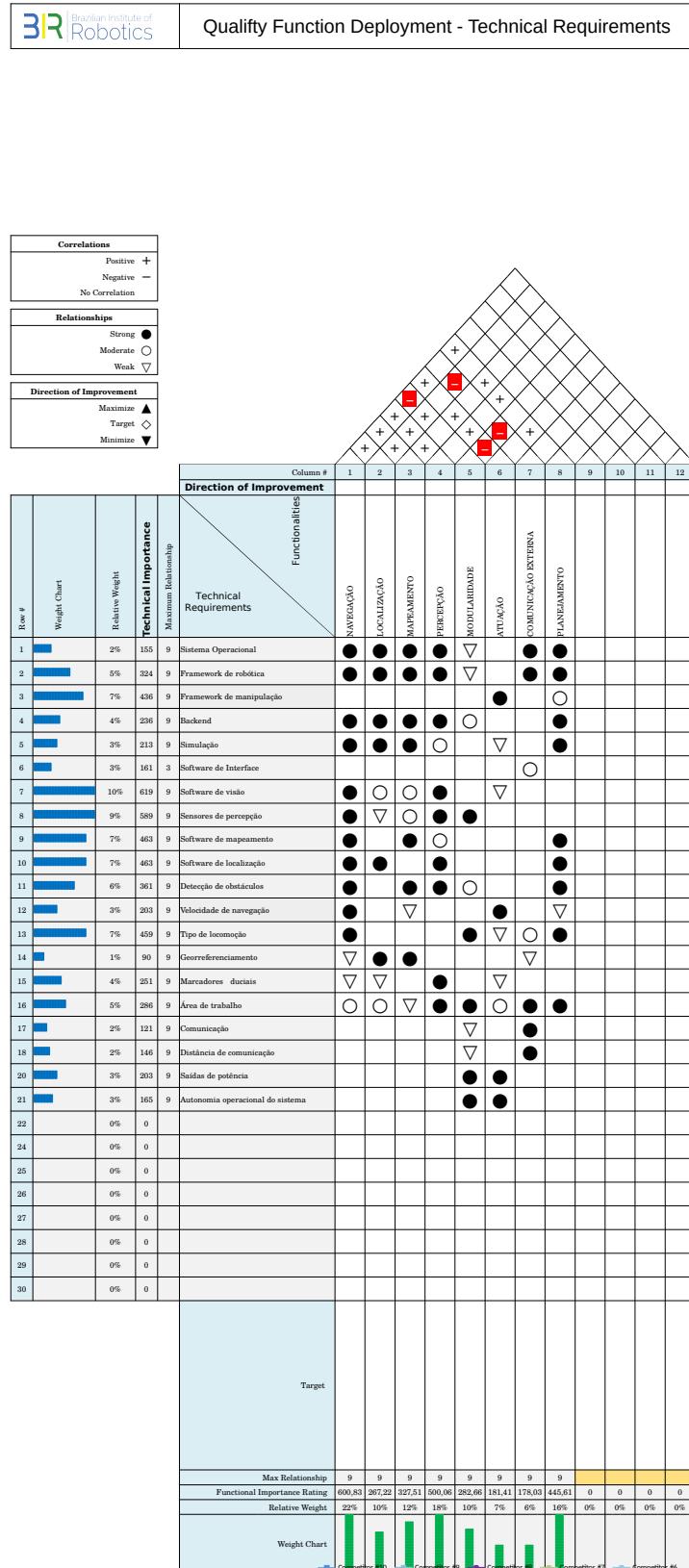


## APÊNDICE C

### Desdobramento da Função Qualidade(QFD)

## Apêndice D







## ANEXO A

### Especificação de componentes utilizados

#### A.1 Plataforma móvel Warthog

A ficha resumo contendo as especificações principais da plataforma móvel *Warthog* da *Clearpath Robotics* está disposta na Tabela 2. Maiores informações e desenho mecânico deste componente estão dispostos no *datasheet* anexado a seguir.

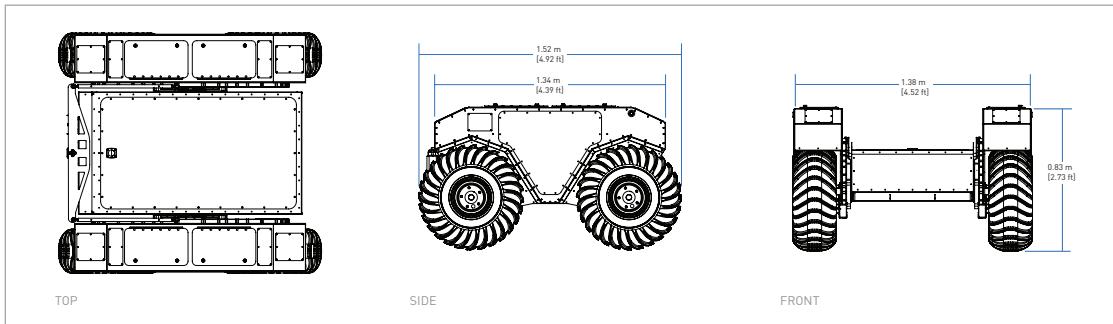
Tabela 2: Especificações da plataforma móvel *Warthog* da *Clearpath Robotics*.

Item	Especificação
Sistema operacional	Ubuntu 16.04 LTS
Versão do ROS	Kinetic
Conexões	Ethernet, USB, Controle Remoto, Wi-Fi
Carga útil	272 [kg]
Tempo de operação nominal	2,5 [h]
Bateria	105 [Ah] @ 48 [V]
Inclinação máxima	35 a 45 °
Velocidade máxima	18 [km/h]
Altura do solo	0,254 [m]
Dimensões	1,52 x 1,38 x 0,83 [m]
Massa bruta	590 [kg]

Fonte: Autoria própria, adaptado de Clearpath Robotics.

# WARTHOG™

## AMPHIBIOUS UNMANNED GROUND VEHICLE



## TECHNICAL SPECIFICATIONS

SIZE AND WEIGHT	
EXTERNAL DIMENSIONS [L x W x H]	1.52 x 1.38 x 0.83 m (4.9 x 4.5 x 2.72 ft)
BASE WEIGHT (includes base battery pack)	280 kg (620 lbs)
GROSS VEHICLE WEIGHT	590 kg (1300 lbs)
GROUND CLEARANCE	254 mm (10 in)
SPEED AND PERFORMANCE	
MAX. PAYLOAD	272 kg (600 lbs)
MAX. INCLINE	35 - 45°
MAX. SPEED	18 km/h (11 mph)
SUSPENSION	Geometric Passive Articulation
TRACTION	24" Argo tire (24" Turf tire or 12" wide Quad Track System optional)
BATTERY AND POWER SYSTEM	
BATTERY CHEMISTRY	AGM sealed lead acid (Li-ion optional)
CAPACITY	105 Ah at 48 V, expandable to 110Ah with Li-ion option
CHARGE TIME	4 hrs
NOMINAL RUN TIME	Lead acid: 2.5 hrs      Li-ion: 3 hrs
USER POWER	5 V, 12 V Fused (24 V, 48 V optional)
INTERFACING AND COMMUNICATION	
CONTROL MODES	Remote control, Computer controlled velocity commands ( $v, \theta$ ), Indoor/outdoor autonomy packages
FEEDBACK	Battery voltage, motor currents, wheel odometry, control system status, temperature, safety status
COMMUNICATION	Ethernet, USB, Remote Control, Wi-Fi
DRIVERS AND APIs	Packaged with ROS Kinetic (includes RViz, Gazebo support), Matlab API available
INCLUDED HARDWARE	IMU, encoders, Onboard computer, E-Stop (hardware loop), E-Stop (software loop), removable mounting plates, bilge pumps, brakes
ENVIRONMENTAL	
OPERATING AMBIENT TEMPERATURE	-20 to 40 °C (-4 to 104 °F)
STORAGE TEMPERATURE	-40 to 50 °C (-40 to 122 °F)
IP RATING	IP65 - Vehicle is designed to float and should not be fully submerged
AMPHIBIOUS	Fully amphibious, 4 km/h (2.4 mph) maximum water speed*

Contact us today for pricing and a free 30 minute technical assessment: 1-800-301-3863

© 2016 Clearpath Robotics, Inc. All Rights Reserved. Clearpath Robotics, Warthog, and clearpathrobotics.com are trademarks of Clearpath Robotics. All other product and company names listed are trademarks or trade names of their respective companies. \* Warthog is not amphibious with Quad Track System configuration.

## A.2 Computador Intel NUC

A ficha resumo contendo as principais especificações do computador Intel NUC5i5RYK utilizado está disposta na Tabela 3. Maiores informações deste componente estão dispostos no *databrief* anexado a seguir.

Tabela 3: Especificações do computador Intel NUC 5i5RYK.

Item	Especificação
Coleção	Intel NUC Kit 5º geração
Sistema operacional	Ubuntu 18.04 LTS
Modelo da placa	NUC5i5RYK
Formato da placa	UCFF (4 x 4 [polegadas])
Processador	Intel Core i5-5250U
Processador gráfico	Intel HD Graphics 6000
Memória de sistema	2x8GB DDR4-2133 1.2V SO-DIMM
Disco de armazenamento de dados	Crucial SSD CT250MX200SSD4 250GB
Periféricos	Intel Gigabit LAN
	4 USB 3.0
	2 USB 2.0
	Intel Dual Band Wireless-AC 8260
	Bluetooth 4.1
Tensão de operação	19 [V]
Temperatura de operação	0 ~40 [°C]
Temperatura de armazenagem	-20 ~60 [°C]
Dimensões (L x A x P)	115 x 34,5 x 111 [mm]
Potência especificada da fonte	65 [W]

Fonte: Autoria própria, adaptado de Intel.

### Power, Capabilities, and Performance in Four Inches Square

#### HIGHLIGHTED FEATURES

- 1 5th generation Intel® Core™ i5-5250U processor
- 2 Two DDR3L SO-DIMM sockets  
(up to 16 GB, 1333/1600 MHz)
- 3 M.2 slot with flexible support for a 42, 60, or 80 mm SATA or PCIe® SSD
- 4 Intel® Dual Band Wireless-AC and Bluetooth 4.0
- 5 Kensington lock support
- 6 Backpanel DC power connector (12V - 19V)
- 7 One Mini DisplayPort® version 1.2 supporting 8 channel digital audio (7.1 surround sound)
- 8 Intel® Gigabit LAN
- 9 2x USB 3.0 ports on the back panel
- 10 Mini HDMI® port supporting HDMI 1.4a and 7.1 surround sound
- 11 Support for user-replaceable third-party lids
- 12 2x USB 3.0 ports on the front panel (including one charging port)
- 13 Intel® HD Audio® via Headphone/Microphone jack
- 14 Consumer infrared sensor



NUC5i5RYK PRODUCT BRIEF 3

### Intel® NUC NUC5i5RYK TECHNICAL SPECIFICATIONS

#### PROCESSOR

- 5th generation Intel® Core™ i5-5250U processor (1.6 GHz up to 2.7 GHz Turbo Dual Core, 3 MB Cache, 15W TDP)
- Supports Intel® Hyper-Threading Technology<sup>1</sup>
- Supports Intel® 64 architecture<sup>2</sup>

#### GRAPHICS

- Intel® HD Graphics 6000
- One Mini DisplayPort<sup>3</sup> version 1.2 supporting ultra-high definition 4K displays and multiple monitor functionality
- One Mini HDMI<sup>4</sup> 1.4a port

#### SYSTEM MEMORY

- Two DDR3L SO-DIMM sockets (up to 16 GB, 1333/1600 MHz) in dual channel configuration, 1.35V

#### STORAGE CAPABILITIES

- One M.2 Type M connector supporting 22x42, 22x60, and 22x80 SATA or PCIe<sup>5</sup> SSDs

#### PERIPHERAL CONNECTIVITY

- Integrated Intel® Gigabit LAN
- Four Super Hi-Speed USB 3.0 ports (two back panel ports and two front ports including one charging port)
- Two additional Hi-Speed USB 2.0 ports via internal header

<sup>1</sup> Requires an Intel® Wireless Display enabled system, compatible adapter and TV. 1080p and Blu-Ray® or other protected content playback only available on select Intel® processor-based systems with built-in visuals enabled, a compatible adapter and media player, and supporting Intel® WDI software and graphics driver. Details can be found on your PC manufacturer. For more information, see [www.intel.com/go/widi](http://intel.com/go/widi).

<sup>2</sup> Requires a system with Intel® Turbo Boost Technology. Intel Turbo Boost Technology and Intel Turbo Boost Technology 2.0 are only available on select Intel® processors. Consult your system manufacturer. Performance varies depending on hardware, software, and system configuration. For more information, visit <http://www.intel.com/go/turbo>

<sup>3</sup> Requires a select Intel® processor, enabled chipset and a Solid-State Drive (SSD) or hybrid drive. Contact your system manufacturer for more information.

<sup>4</sup> Requires an Intel® Ready Mode Technology-enabled system or motherboard, a genuine Intel® processor, and Windows® 7 or Windows 8 OS. Results dependent upon hardware, applications installed, Internet connectivity, setup, and configuration.

Copyright 2015 © Intel Corporation. All rights reserved. Intel, the Intel logo, Look Inside, the Look Inside logo, Intel Inside, and Intel Core are trademarks of Intel Corporation in the U.S. and/or other countries.

<sup>5</sup> Other names and brands may be claimed as the property of others.

- Intel® Dual Band Wireless-AC 7265, 802.11ac, 2x2, up to 867 Mbps
- Dual Mode Bluetooth 4.0

#### SYSTEM BIOS

- 64 Mb Flash EEPROM with Intel® Platform Innovation Framework for EFI Plug and Play
- Advanced configuration and power interface V3.0b, SMBIOS2.5
- Intel® Visual BIOS
- Intel® Express BIOS update support

#### HARDWARE MANAGEMENT FEATURES

- Processor fan speed control
- Voltage and temperature sensing
- Fan sensor inputs used to monitor fan activity
- ACPI-compliant power management control

#### EXPANSION CAPABILITIES

- One NFC header
- 2x Internal USB 2.0 ports via 1x8 header (for replaceable lid support)
- One AUX\_PWR header

#### AUDIO

- Intel® HD Audio<sup>6</sup> via Mini HDMI 1.4a and Mini DisplayPort version 1.2 supporting 8 channel digital audio (7.1 surround sound)
- Intel HD Audio via front panel analog audio jack (supporting headset, speakers, headphones, microphone)

#### FRONT-PANEL CONNECTORS

- Reset, HDD LED, Power LEDs, power on/off

#### MECHANICAL CHASSIS SIZE

- 4.53" x 4.37" x 1.36"
- 115mm x 111mm x 34.5mm

#### BASEBOARD POWER REQUIREMENTS

- 19V, 65W wall-mount AC-DC power adapter
- Multi-country AC adapter (IEC plug types A, C, G, and I)

#### ENVIRONMENT OPERATING TEMPERATURE

- 0° C to +40° C

#### STORAGE TEMPERATURE

- -20° C to +60° C

#### PRODUCT SAFETY REGULATIONS AND STANDARDS

- IEC 60950-1
- UL 60950-1
- EN 60950-1
- CAN/CSA-C22.2 No. 60950-1

#### EMC REGULATIONS AND STANDARDS (CLASS B)

- CISPR 22
- FCC CFR Title 47, Chapter I, Part 15, Subparts A, B
- ICES-003
- EN 55022
- EN 55024
- VCCI V-3, V-4
- KN-22
- KN-24
- CNS 13438

#### ENVIRONMENTAL REGULATIONS

- RoHS Directive 2011/65/EU
- WEEE Directive 2002/96/EC
- China RoHS MII Order #39

Look for Intel® NUC with Intel Inside® at [www.intel.com/NUC](http://www.intel.com/NUC)

Intel products are not intended for use in medical, life-saving, or life-sustaining applications. Intel may make changes to specifications and product descriptions at any time, without notice.

All products, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice. Availability in different channels may vary.

Actual Intel® NUC kit may differ from the image shown.

<sup>6</sup> System resources and hardware (such as PCI and PCI Express<sup>7</sup>) require physical memory address locations that can reduce available addressable system memory. This could result in a reduction of as much as 1 GB or more of physical addressable memory being available to the operating system and applications, depending on the system configuration and operating system.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT, EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS. INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO THE SALE AND/OR USE OF INTEL PRODUCTS. INTEL DOES NOT WARRANT THAT THE USE OF INTEL PRODUCTS WILL BE UNINTERRUPTED OR ERROR-FREE, OR THAT DEFECTS WILL BE CORRECTED FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT.

Printed in USA

0215BCP|HBD/PDF

Please Recycle

331508-001US



### A.3 Switch MikroTik RB951G

A ficha resumo contendo as especificações principais do *switch* MikroTik RB951G está disposta na Tabela 4. Maiores informações e desenho mecânico deste componente estão dispostos no *datasheet* anexado a seguir.

Tabela 4: Especificações do *switch* MikroTik RB951G-2HnD.

Item	Especificação
CPU	Atheros AR9344 600MHz
Memória	128 MB DDR2
Ethernet	5 portas Gigabit
Antenas	2x2 MIMO PIF
Potência de operação	<7 [W]
Tensão de alimentação	8 - 30 [V]
Dimensões (L x A x P)	113 x 138 x 29 [mm]

Fonte: Autoria própria, adaptado de MikroTik.

]

## RB951G-2HnD

The RB951G-2HnD is a wireless SOHO Gigabit AP with a new generation Atheros CPU and more processing power. It has five Gigabit Ethernet ports, one USB 2.0 port and a high power 2.4GHz 802.11b/g/n wireless AP with antennas built in.

In comparison with previous model RB751G-2HnD, it has more powerful 600Mhz CPU (instead of 400Mhz), more RAM – 128MB instead of 64MB, same form factor and price. The device is very small and will look good in any home or office, wall mounting anchor holes are provided.

Package contains RouterBOARD 951G-2HnD in a plastic case and power adapter. Specific frequency range may be limited by country regulations.



Features	RB951G-2HnD (Gigabit, USB, 2GHz, 802.11n, dual chain)
CPU	Atheros AR9344 600MHz CPU
Memory	128MB DDR2 onboard memory
Ethernet	Five independent 10/100/1000 Gigabit Ethernet ports
LEDs	Power, NAND activity, 5 Ethernet LEDs, wireless activity LED
Power options	PoE: 8-30V DC on Ether1 (Non 802.3af). Jack: 8-30V DC
Dimensions	113x138x29mm
Weight	Without packaging and PSU: 232g, full weight in package: 420g
Power consumption	Up to 7W
Operating Temp	-20C .. +50C
Operating System	MikroTik RouterOS, Level4 license
Package contains	RouterBOARD in a plastic case, power adapter
Antennas	2x2 MIMO PIF antennas, max gain 2.5dBi
RX sensitivity	802.11g: -96dBm @ 6Mbps to -80dBm @ 54Mbps 802.11n: -96dBm @ MCS0 to -78dBm @ MCS7
TX power	802.11g: 30dBm @ 6Mbps to 25dBm @ 54 Mbps 802.11n: 30dBm @ MCS0 to 23dBm @ MCS7
Modulations	OFDM: BPSK, QPSK, 16 QAM, 64QAM DSSS: DBPSK, DQPSK, CCK

## A.4 Lidar Velodyne VLP-16

A ficha resumo contendo as especificações principais do *lidar* Velodyne VLP-16 está disposta na Tabela 5. Maiores informações e desenho mecânico deste componente estão dispostos no *datasheet* anexado a seguir.

Tabela 5: Especificações do *lidar* Velodyne VLP-16.

Item	Especificação
Quantidade de canais	16
Distância máxima	100 [m]
Acurácia	$\pm 3$ [cm]
Ângulo de visão vertical	+15 a -15 [°]
Resolução angular	2 [°]
Frequência de giro	5 a 20 [Hz]
Laser	903 [nm]
Conexão	Ethernet 100 Mbps
Potência de operação	8 [W]
Tensão de alimentação	9 - 18 [V]
Massa	830 [g]
Dimensões (L x A x P)	103,3 x 103,3 x 71,7 [mm]

Fonte: Autoria própria, adaptado de Velodyne Lidar.



### Puck



Velodyne Lidar's Puck is a small and compact lidar that is performance and power optimized for usage across a variety of applications ranging from automotive, mapping, robotics, security, smart cities and more. The Puck is attractively priced and built on the foundations of Velodyne's leadership in lidar, enabling real-time, surround view, 3D distance and calibrated reflectivity measurements.

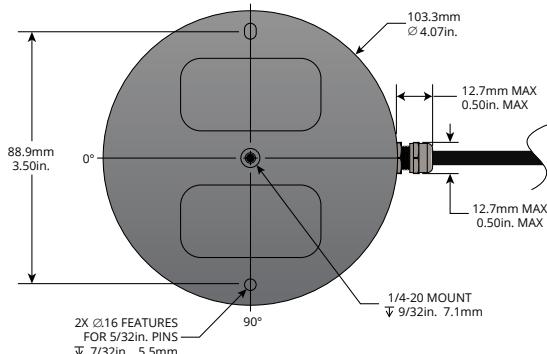
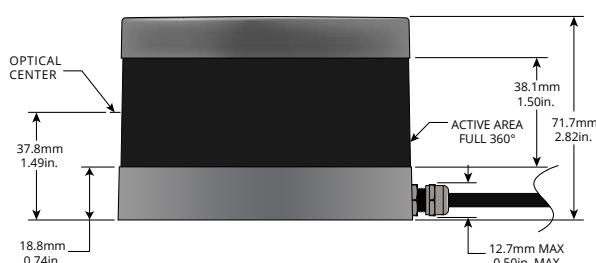
The Puck has a range of 100 m and generates up to ~600,000 points/second, across a 360° horizontal field of view and a 30° vertical field of view. It uses proven, Class 1 eye-safe 905 nm technology with substantial autonomous fleet validation, making the Puck a sensor of choice for lower speed autonomous vehicle (AV) applications. The Puck has best-in-class power, which enables operation over a wide temperature range. Its use of off-the-shelf components enables enhanced scalability and attractive volume pricing. Like other Velodyne sensors, the Puck has world-class technical support available across North America, Europe & Asia from the world's leading lidar company.



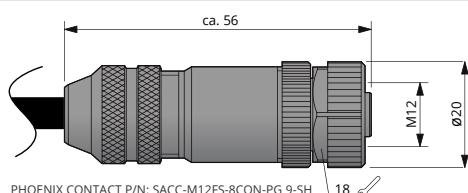
### Puck at a glance

- 100 m range with compact form factor
- Proven 905 nm tech, with largest install base
- Top-of-the-line field-of-view
- Best-in-class accuracy and calibrated intensity
- Best-in-class power & temperature range
- Sensor-to-sensor interference mitigation feature
- Optional, enhanced short range detection
- Versatile, with attractive volume pricing

### DIMENSIONS (Subject to change)



### M12 CONNECTOR OPTION



For other connector options contact  
**Velodyne Sales (sales@velodyne.com)**

[velodynelidar.com](http://velodynelidar.com)

## Puck

### Real-Time Lidar Sensor

The Puck provides high definition 3-dimensional information about the surrounding environment.



Specifications:	
<b>Sensor:</b>	<ul style="list-style-type: none"> <li>• 16 Channels</li> <li>• Measurement Range: 100 m</li> <li>• Range Accuracy: Up to ±3 cm (Typical)<sup>1</sup></li> <li>• Field of View (Vertical): +15.0° to -15.0° (30°)</li> <li>• Angular Resolution (Vertical): 2.0°</li> <li>• Field of View (Horizontal): 360°</li> <li>• Angular Resolution (Horizontal/Azimuth): 0.1° – 0.4°</li> <li>• Rotation Rate: 5 Hz – 20 Hz</li> <li>• Integrated Web Server for Easy Monitoring and Configuration</li> </ul>
<b>Laser:</b>	<ul style="list-style-type: none"> <li>• Laser Product Classification: Class 1 Eye-safe per IEC 60825-1:2007 &amp; 2014</li> <li>• Wavelength: 903 nm</li> </ul>
<b>Mechanical/ Electrical/ Operational</b>	<ul style="list-style-type: none"> <li>• Power Consumption: 8 W (Typical)<sup>2</sup></li> <li>• Operating Voltage: 9 V – 18 V (with Interface Box and Regulated Power Supply)</li> <li>• Weight: ~830 g (without Cabling and Interface Box)</li> <li>• Dimensions: See diagram on previous page</li> <li>• Environmental Protection: IP67</li> <li>• Operating Temperature: -10°C to +60°C<sup>3</sup></li> <li>• Storage Temperature: -40°C to +105°C</li> </ul>
<b>Output:</b>	<ul style="list-style-type: none"> <li>• 3D Lidar Data Points Generated:             <ul style="list-style-type: none"> <li>- Single Return Mode: ~300,000 points per second</li> <li>- Dual Return Mode: ~600,000 points per second</li> </ul> </li> <li>• 100 Mbps Ethernet Connection</li> <li>• UDP Packets Contain:             <ul style="list-style-type: none"> <li>- Time of Flight Distance Measurement</li> <li>- Calibrated Reflectivity Measurement</li> <li>- Rotation Angles</li> <li>- Synchronized Time Stamps (μs resolution)</li> </ul> </li> <li>• GPS: \$GPRMC and \$GPGGA NMEA Sentences from GPS Receiver (GPS not included)</li> </ul>

63-9229 Rev-K      VLP-16

For more details and ordering information, contact Velodyne Sales ([sales@velodyne.com](mailto:sales@velodyne.com))

1. Typical accuracy refers to ambient wall test performance across most channels and may vary based on factors including but not limited to range, temperature and target reflectivity.

2. Operating power may be affected by factors including but not limited to range, reflectivity and environmental conditions.

3. Operating temperature may be affected by factors including but not limited to air flow and sun load.



CLASS 1 LASER PRODUCT

Copyright ©2019 Velodyne Lidar, Inc. Specifications are subject to change. Other trademarks or registered trademarks are property of their respective owners.

**Velodyne Lidar, Inc.** 5521 Hellyer Ave, San Jose, CA 95138 / [lidar@velodyne.com](mailto:lidar@velodyne.com) / 408.465.2800

**velodynelidar.com**

## A.5 GPS Swiftnav Piksi V2

A ficha resumo contendo as especificações principais do GPS Swiftnav Piksi V2 está disposta na Tabela 6. Maiores informações e desenho mecânico deste componente estão dispostos no *datasheet* anexado a seguir.

Tabela 6: Especificações do GPS Swiftnav Piksi V2.

Item	Especificação
Acurácia (RTK)	~1 [cm]
Frequência de atualização	10 [Hz]
Conexões	USB e UART
Potência de operação	500 [mW]
Tensão de alimentação	3,5 - 5,5 [V]
Massa	32 [g]
Dimensões (L x A x P)	53 x 62,5 x 9,1 [mm]

Fonte: Autoria própria, adaptado de Swiftnav.



## Piksi Datasheet

Flexible, high-performance GPS receiver platform running open-source software

---

### Features

- Centimeter-accurate relative positioning (Carrier phase RTK)
- 10 Hz position/velocity/time solutions
- Open-source software and board design
- Low power consumption - 500mW typical
- Small form factor - 53x53mm
- USB and dual UART connectivity
- External antenna input
- Full-rate raw sample pass-through over USB

### Applications

- Autonomous Vehicle Guidance
- GPS/GNSS Research
- Surveying Systems
- Precision Agriculture
- Unmanned Aerial Vehicles
- Robotics
- Space Applications

### Overview

Piksi™ is a low-cost, high-performance GPS receiver with Real Time Kinematics (RTK) functionality for centimeter-level relative positioning accuracy.

Its small form factor, fast position solution update rate and low power consumption make Piksi ideal for integration into autonomous vehicles and portable surveying equipment.

Piksi's open source firmware allows it to be easily customized to the particular demands of end users' applications, easing system integration and reducing host system overhead.

In addition, Piksi's use of the same open source GNSS libraries as Peregrine, Swift Navigation's GNSS post-processing software, make the combination of the two a powerful toolset for GNSS research, experimentation and prototyping at every level from raw samples to position solutions.

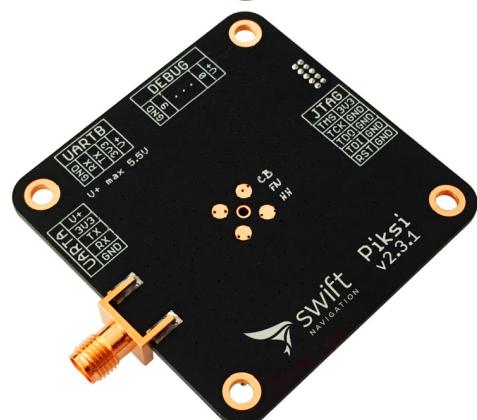
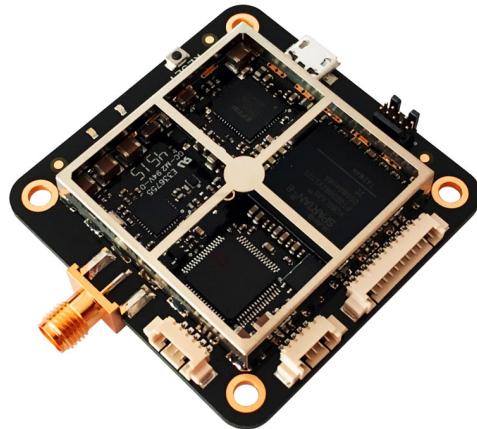


Figure 1: Piksi front and back view

With these tools, developers can quickly move from prototyping software on a desktop to running it standalone on the Piksi hardware.

A high-performance DSP on-board and our flexible Swift-NAP correlation accelerator provide Piksi with ample computing resources with which advanced receiver techniques, such as multipath mitigation, spoofing detection and carrier phase tracking can be implemented.

## System Architecture

The Piksi receiver architecture consists of three main components. The RF front-end downconverts and digitizes the radio frequency signal from the antenna. The digitized signal is passed into the SwiftNAP which performs basic filtering and correlation operations on the signal stream. The SwiftNAP is controlled by a microcontroller which programs the correlation operations, collects the results and processes them all the way to position/velocity/time (PVT) solutions.

### Front-end

The RF front-end consists of a Maxim MAX2769 integrated down-converter and 3-bit analog-to-digital converter operating at 16.368 MS/s. This front-end is capable of covering the L1 GPS signal bands.

### SwiftNAP

The SwiftNAP consists of a Xilinx Spartan-6 FPGA that comes pre-programmed with Swift Navigation's SwiftNAP

firmware. The SwiftNAP contains correlators specialized for satellite signal tracking and acquisition. The correlators are flexible and fully programmable via a high-speed SPI register interface and are used as simple building blocks for implementing tracking loops and acquisition algorithms on the microcontroller.

While the SwiftNAP HDL is not open-source at this time, the Piksi has no restrictions against loading one's own firmware onto the on-board Spartan-6 FPGA.

### Microcontroller

The on-board microcontroller is a STM32F4 with an ARM Cortex-M4 DSP core running at up to 168 MHz. This powerful processor performs all functions above the correlator level including tracking loop filters, acquisition management and navigation processing and is able to calculate PVT solutions at over 10 Hz in our default software configuration. All software running on the microcontroller is supplied open-source.

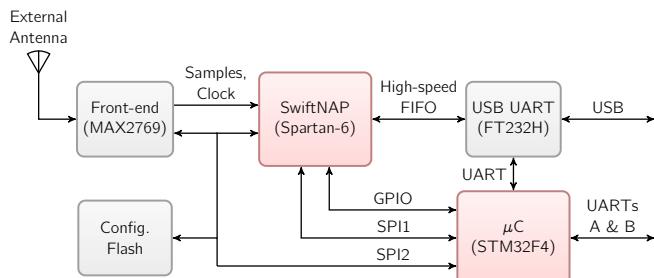


Figure 2: Piksi Block Diagram

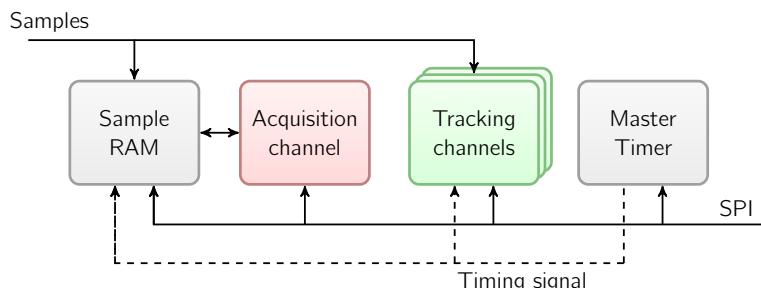
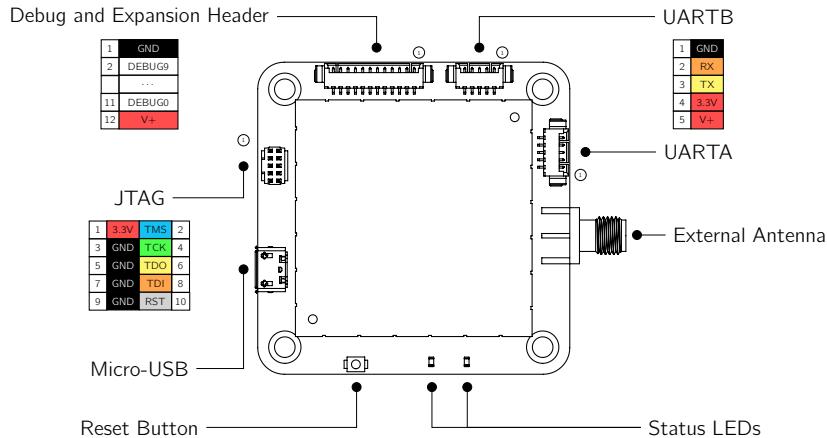


Figure 3: SwiftNAP Block Diagram

## Connections



### USB

A Micro-USB socket provides USB connectivity to the host. This can be configured as a USB-Serial bridge to the microcontroller (the default) or as a high-speed FIFO interface to the SwiftNAP for streaming full-rate raw IF data samples to or from the host.

This allows capture of raw IF data for processing on the host or running the Piksi from pre-recorded data or simulator output for hardware-in-the-loop testing.

### UARTs A & B

Two UARTs provide high-speed 3.3V LVTTL level asynchronous serial interfaces which can be configured to transmit NMEA-0183 messages or binary navigation solution data, system status and debugging information and receive commands or differential corrections from the host or another Piksi board.

When configured in USB-Serial bridge mode, the USB interface functions identically to the two dedicated UARTs.

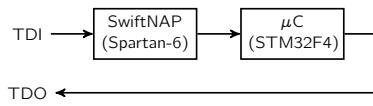
### External Antenna

An external active antenna input is provided on an SMA connector and features a software switchable 3.3V bias.

### JTAG

No JTAG adapter is required to develop for the Piksi as the board is supplied with a built-in bootloader.

For advanced debugging, a 0.05" pitch micro JTAG header compatible with the ARM Cortex-M standard pinout<sup>1</sup> is provided on the board. This allows access to the Spartan-6 FPGA and STM32F4 microcontroller JTAG interfaces.



### Debug and Expansion Header

Access is provided to debugging signals from the SwiftNAP and I/O for future expansion boards and accessories. Assignment of these signals varies depending on the SwiftNAP firmware configuration.

### Power

Power may be supplied to the board either over USB or through the V+ pins on the UART connectors. A 3.3V output from the on-board switching regulator is provided to power any external peripherals.

<sup>1</sup>ARM Cortex-M Debug Connector specification, <http://bit.ly/ICb6W6>

## Electrical Specifications

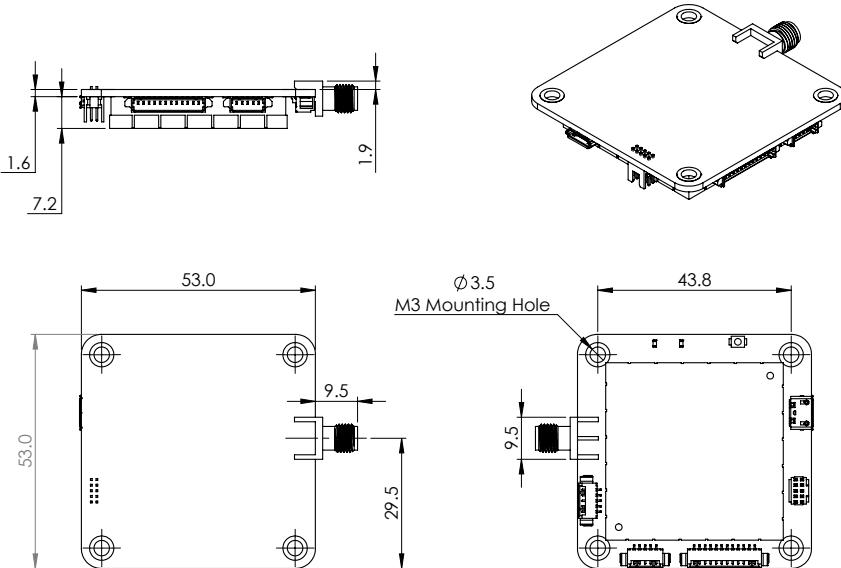
Supply voltage .....	3.5 – 5.5 V	Active antenna input impedance .....	50 $\Omega$
Maximum voltage rating.....	5.5 V <sup>(1)</sup>	Active antenna bias voltage .....	3.3 V <sup>(3)</sup>
Power consumption .....	500 mW <sup>(2)</sup>	Max. antenna bias current draw .....	57 mA
Max. 3.3V output current draw .....	500 mA		

<sup>(1)</sup>Piksi provides no overvoltage protection and even momentary overvoltage can permanently damage the device.

<sup>(2)</sup>Typical, dependant on firmware configuration.

<sup>(3)</sup>Switchable in software

## Mechanical Drawing



All dimensions are in millimeters. Drawing not to scale.

### Notes

1. Mass 32g.
2. M3 mounting holes are plated through and connected internally to ground.
3. 3D CAD models are available from our website, <http://www.swiftnav.com>.

## A.6 Câmera Basler ace acA4600

A ficha resumo contendo as especificações principais da câmera colorida de alta resolução Basler ace acA4600 está disposta na Tabela 7. Maiores informações e desenho mecânico deste componente estão dispostos no *datasheet* anexado a seguir.

Tabela 7: Especificações da câmera Basler acA4600-10uc.

Item	Especificação
Especificação de câmera	acA4600
Resolução	4608 × 3288 [px]
Modelo do sensor	On-Semi MT9F002
Taxa de captura (padrão)	10 [fps]
Tamanho de pixel	1,4 × 1,4 $\mu$ m
Alcance dinâmico	62,1 [dB]
Círculo máximo de imagem	Formato óptico 1/2,3"
Encaixe de lente	C-Mount
Protocolo de comunicação	USB 3.0 (5 [Gbit/s])
Porta de comunicação	USB
Tensão de operação	5 [V]
Temperatura de operação	0 ~ 50 [°C]
Massa	< 80 [g]
Dimensões (L x A x P)	29,3 x 29 x 29 [mm]
Potência necessária	~ 2,5 W a 5 V <sub>DC</sub> , máximo de 2,8 W

Fonte: Autoria própria, adaptado de Basler.

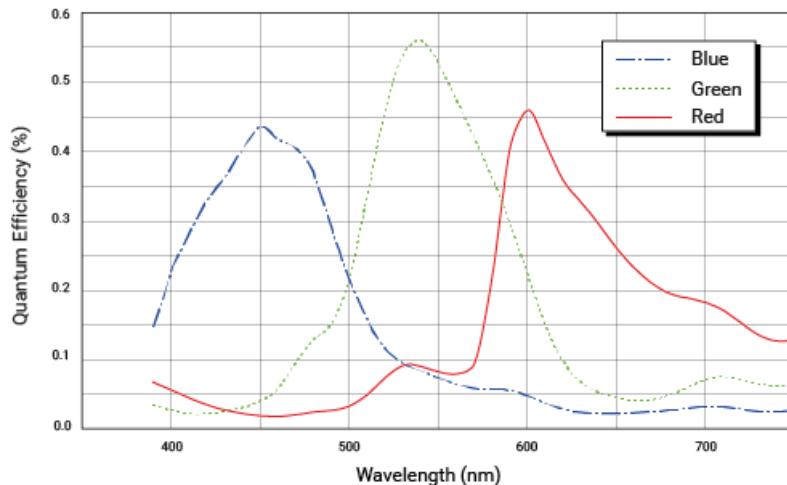
## Specifications

### General Specifications

Specification	acA4600-10uc
Resolution (H x V Pixels)	4608 x 3288
Sensor Type	ON Semiconductor MT9F002 Progressive scan CMOS Rolling shutter
Optical Size	1/2.3"
Effective Sensor Diagonal	8.0 mm
Pixel Size (H x V)	1.4 µm x 1.4 µm
Frame Rate (at Default Settings)	10 fps
Product Line	 ace classic
Mono / Color	Color
Image Data Interface	USB 3.0, nominal max. 5 Gbit/s (SuperSpeed)
Pixel Formats	See <a href="#">Pixel Format</a> .
Synchronization	Via hardware trigger Via software trigger Via free run
Exposure Time Control	Programmable via the camera API
Camera Power Requirements	Nominal 5 VDC supplied via the camera's USB 3.0 port ≈2.5 W (typical) @ 5 VDC ≈2.8 W (max.)
I/O Lines	1 opto-coupled input line

	1 opto-coupled output line 2 general purpose I/O (GPIO) lines
Lens Mount	C-mount
Size (L x W x H)	29.3 mm x 29 mm x 29 mm (without lens mount or connectors) 48.2 mm x 29 mm x 29 mm (with lens mount and connectors)
Weight	<80 g
Conformity	CE (includes RoHS), UL Listed, FCC, GenICam 2.x (including PFNC 2.x and SFNC 2.x), IP30, USB3 Vision, REACH The EU Declaration of Conformity is available on the <a href="#">Basler website</a> .
Software	<a href="#">Basler pylon Camera Software Suite</a> (version 4.0 or higher) Available for Windows, Linux x86, Linux ARM, and OS X
Accessories	<a href="#">Cables for your camera model</a> <a href="#">Lenses for your camera model</a> <a href="#">Additional accessories for your camera model</a>

## Spectral Response



The spectral response curve excludes lens characteristics, light source characteristics, and IR cut filter characteristics.

## IR Cut Filter

Color cameras are equipped with an IR cut filter. The filter is mounted in a filter holder inside the lens mount.

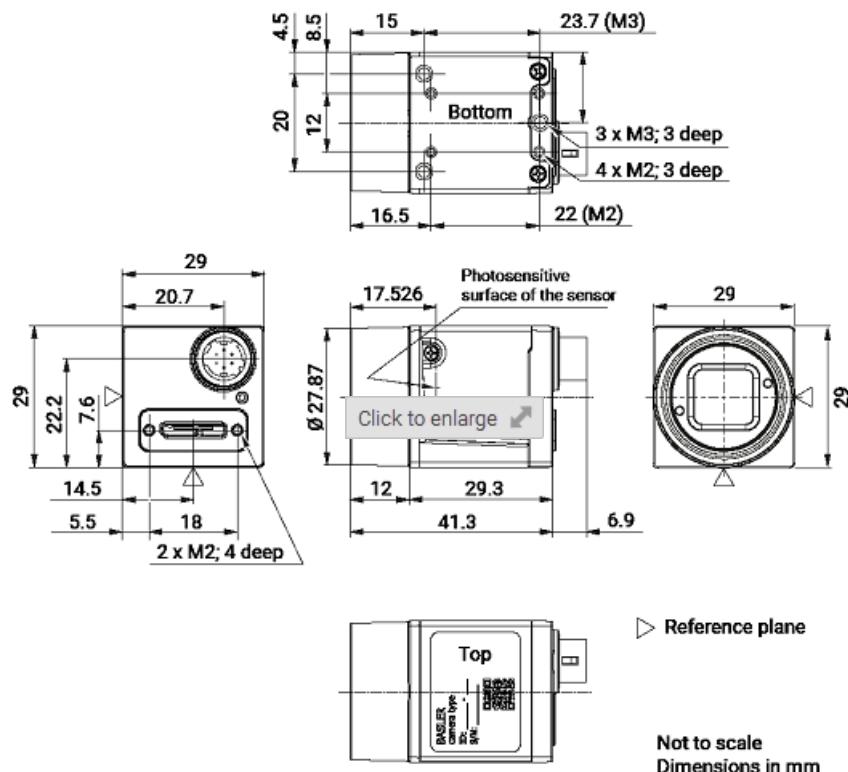
The IR cut filter has the following spectral characteristics:

Wavelength [nm]	Transmittance
450–610	$T_{\min} > 90 \%$
450–620	$T_{\text{avg}} > 93 \%$
$645 \pm 10$	$T = 50 \%$
700–1070	$T_{\max} < 4 \%$
690–1070	$T_{\text{avg}} < 1 \%$

The filter holder can be removed. For more information, see the [ACE IR Cut Filter Holder Removal Procedure](#) application note.

## Mechanical Specifications

### Camera Dimensions and Mounting Points



### Maximum Allowed Lens Intrusion

→ See [Maximum Allowed Lens Intrusion](#).

## Mounting Instructions

→ See [Mounting Instructions](#).

## Stress Test Results

→ See [Stress Test Results](#).

## Requirements

---

### Environmental Requirements

#### Temperature and Humidity

Housing temperature during operation	0–50 °C (32–122 °F)
Humidity during operation	20–80 %, relative, non-condensing
Storage temperature	-20–80 °C (-4–176 °F)
Storage humidity	20–80 %, relative, non-condensing
Housing temperature according to UL 60950-1	max. 70 °C (158 °F)
Ambient temperature according to UL 60950-1	max. 30 °C (86 °F)

UL 60950-1 test conditions: no lens attached to camera; no heat dissipation measures; ambient temperature kept at 30 °C (86 °F).

#### Heat Dissipation

→ See [Providing Heat Dissipation](#).

## Electrical Requirements

<b>DANGER</b>
Electric Shock Hazard
<b>WARNING</b>
Fire Hazard
<b>NOTICE</b>
Incorrect voltage can damage the camera.

#### Camera Power

You must supply camera power that complies with the Universal Serial Bus 3.0 specification.

The camera's nominal operating voltage is 5 VDC, effective on the camera's connector.

### Opto-Coupled I/O Input Line

Voltage	Description
30 VDC	Absolute maximum. This voltage must never be exceeded. Doing so may damage the camera and voids the warranty.
0–24 VDC	Safe operating range.
0–1.4 VDC	Indicates a logical 0 (with inverter disabled).
>1.4–2.2 VDC	Region where the logic level transition occurs; the logical status is not defined in this region.
>2.2 VDC	Indicates a logical 1 (with inverter disabled).
<ul style="list-style-type: none"> <li>• <b>Minimum current:</b> 5 mA</li> <li>• <b>Current draw:</b> 5–15 mA</li> </ul>	

### Opto-Coupled I/O Output Line

Voltage	Description
30 VDC	Absolute maximum. This voltage must never be exceeded. Doing so may damage the camera and voids the warranty.
3.3–24 VDC	Safe operating range.
<3.3 VDC	Unreliable I/O output.
<ul style="list-style-type: none"> <li>• <b>Leakage current:</b> &lt;60 <math>\mu</math>A. Actual leakage depends on operating temperature and production spread of electronic components.</li> <li>• <b>Maximum load current:</b> 50 mA</li> <li>• <b>Minimum load current:</b> Not specified. Consider the following: <ul style="list-style-type: none"> <li>• Leakage current will have a stronger effect when load currents are low.</li> <li>• Propagation delay of the output increases as load currents decrease.</li> <li>• Higher-impedance circuits tend to be more susceptible to EMI.</li> <li>• Higher currents cause higher voltage drops in long cables.</li> </ul> </li> </ul>	

### General Purpose I/O Lines

#### NOTICE

Applying incorrect electrical signals to the camera's GPIO line can severely damage the camera.

### Operation as Input

Voltage	Description
30 VDC	Absolute maximum. This voltage must never be exceeded. Doing so may damage the camera and voids the warranty.
0–5 VDC	Safe operating range. The minimum external pull-up voltage is 3.3 VDC.
0–0.8 VDC	Indicates a logical 0 (with inverter disabled).
>0.8–2.0 VDC	Region where the logic level transition occurs; the logical status is not defined in this region.
>2.0 VDC	Indicates a logical 1 (with inverter disabled).
<ul style="list-style-type: none"> <li><b>Current draw (high-level):</b> &lt;100 µA</li> <li><b>Sink current:</b> Your application must be able to accept 2 mA sink current from the GPIO input line without exceeding 0.8 VDC.</li> </ul>	

### Operation as Output

Voltage	Description
30 VDC	Absolute maximum. This voltage must never be exceeded. Doing so may damage the camera and voids the warranty.
3.3–24 VDC	Safe operating range.
<3.3 VDC	Unreliable GPIO output.
<ul style="list-style-type: none"> <li><b>Internal pull-up resistor:</b> <math>\approx 2\text{ k}\Omega</math>, with open collector. Many applications will have to provide an additional pull-up resistor.</li> <li><b>Residual voltage ("on" state):</b> <math>\approx 0.4\text{ V}</math> at 50 mA and <math>25^\circ\text{C}</math> (<math>77^\circ\text{F}</math>) housing temperature. Actual residual voltage depends on operating temperature, load current, and production spread of electronic components.</li> <li><b>Leakage current:</b> &lt;60 µA. Actual leakage depends on operating temperature and production spread of electronic components.</li> <li><b>Maximum load current:</b> 50 mA</li> <li><b>Minimum load current:</b> Not specified. However, consider the following: <ul style="list-style-type: none"> <li>Leakage current will have a stronger effect when load currents are low.</li> <li>Propagation delay of the output increases as load currents decrease.</li> <li>Higher-impedance circuits tend to be more susceptible to EMI.</li> </ul> </li> </ul>	

- Higher currents cause higher voltage drops in long cables.

## Circuit Diagrams

→ See [Circuit Diagrams for Basler ace Cameras](#).

## Cable Requirements

### USB 3.0 Cable

- Use a high-quality USB 3.0 cable with a Micro-B plug.
- To [avoid EMI](#), cables must be shielded, as specified in the USB 3.0 standard.
- Basler recommends using USB 3.0 cables from the [Basler Vision Components](#) range.

For more information about recommended USB 3.0 cables, see the [Recommended Accessories for Basler USB 3.0 Cameras](#) document.

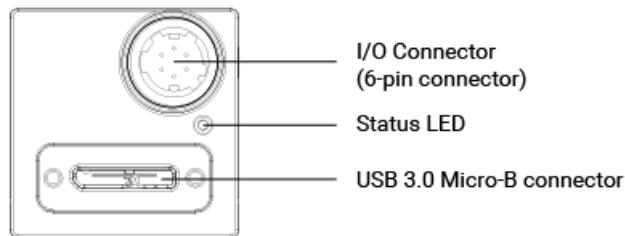
### I/O Cable

- The I/O cable must be shielded.
- The I/O cable must have a cross-section at least  $0.14 \text{ mm}^2$  (close to AWG26).
- Use twisted pair wire cables.
- Maximum recommended cable length: 10 m
- Camera-side connector: Hirose micro plug (part number HR10A-7P-6S) or equivalent
- Close proximity to strong magnetic fields should be avoided.
- Basler recommends using I/O cables from the [Basler Vision Components](#) range:
  - [GPIO cable, 10 m](#) (yellow cable): For use with the [GPIO lines](#) of your camera. To avoid interferences due to crosstalk, the opto-coupled I/O lines are not connected.
  - [Opto-I/O cable, 10 m](#) (blue cable): For use with the [opto-coupled I/O lines](#) of your camera. To avoid interferences due to crosstalk, the GPIO lines are not connected.
  - [Opto-GPIO Y-cable, 2 x 10 m](#) (yellow-blue cable): Allows you to use the [GPIO lines](#) and the [opto-coupled I/O lines](#) simultaneously without interferences due to crosstalk. There are two separate wires to split both I/O types.

## Physical Interface

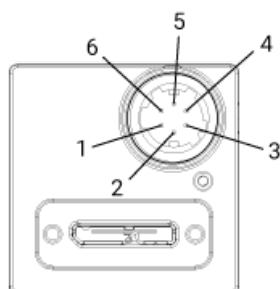
---

### Camera Connectors and Status LED



6-pin connector	Hirose micro receptacle (part number HR10A-7R-6PB) Recommended mating connector: Hirose micro plug (part number HR10A-7P-6S)
USB 3.0 Micro-B connector	Standard USB 3.0 Micro-B connector with screw lock Recommended mating connector: Standard connector with screws
Status LED	Indicates camera operation (LED lit = camera operating).

## Connector Pin Numbering and Assignments



Pin	Line	Function
1	Line 3	General purpose I/O (GPIO) line
2	Line 1	Opto-coupled I/O input line
3	Line 4	General purpose I/O (GPIO) line
4	Line 2	Opto-coupled I/O output line
5	-	Ground for opto-coupled I/O lines
6	-	Ground for General Purpose I/O (GPIO) lines

## Precautions

---

→ See [Safety Instructions for Basler ace Cameras](#).

## Installation

---

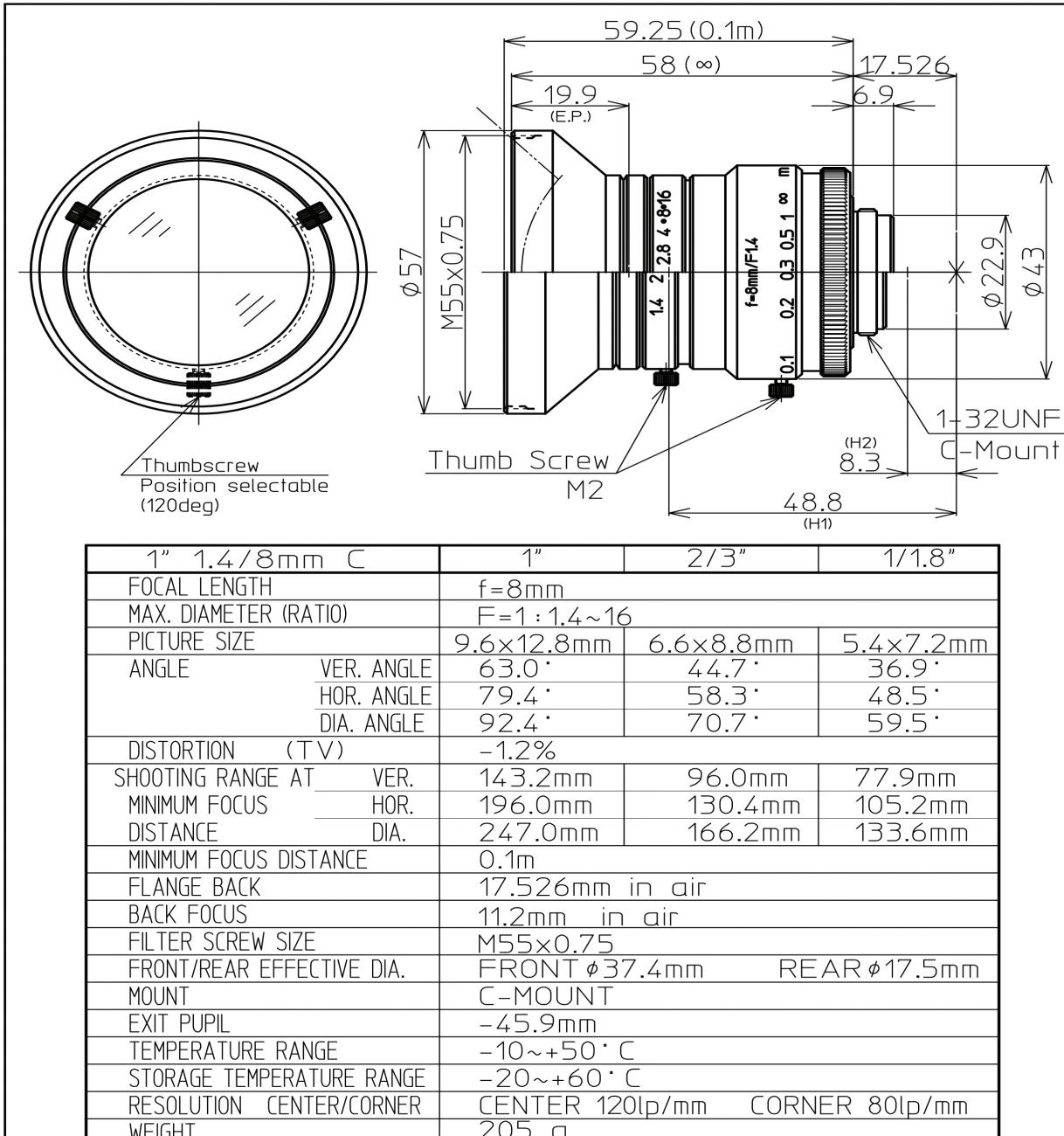
## A.7 Lente Kowa LM8HC

A ficha resumo contendo as principais especificações da lente Kowa LM8HC utilizada em conjunto com a câmera fotográfica está disposta na Tabela 8. Maiores informações e desenho mecânico deste componente estão dispostos no *datasheet* anexado a seguir.

Tabela 8: Especificações da lente Kowa LM8HC para sensor de 1 polegada de diâmetro.

Item	Especificação
Comprimento focal	8 [mm]
Razão de diâmetro máxima	$F = 1 : 1,4 \sim 1,6$
Tamanho da imagem	9,6 x 12,8 [mm]
Ângulo de visão vertical	63,0 [°]
Ângulo de visão horizontal	79,4 [°]
Ângulo de visão diagonal	92,4 [°]
Distância focal mínima	0,1 [mm]
Tamanho do filtro	M55 x 0.75
Encaixe de lente	C-Mount
Temperatura de operação	-10 ~ 50 [°C]
Temperatura de armazenagem	-20 ~ 60 [°C]
Massa	205 [g]
Dimensões (L x A x P)	66,15 x 57 x 57 [mm]

Fonte: Autores, adaptado de Kowa American Corp.



NOTE: Specifications and availability are subject to change without notice

The table contains the following information:

	1/1	HR975NCN-3H
2015.7.30	1" F1.4 f=8mm	
2014.12.25		
2013.4.15	LM8HC	

## A.8 Manipulador JeRoTimon

A ficha resumo contendo as especificações principais do Manipulador JeRoTimon está disposta na Tabela 9. Maiores informações e desenho mecânico deste componente estão dispostos no *datasheet* anexado a seguir.

Tabela 9: Especificações do manipulador JeRoTimon.

Item	Especificação
DOF	5
Carga útil	2 [kg]
Alcance	981 [mm]
Ângulos de operação	Junta 1: -45 a 45 [°] Junta 2: -90 a 90 [°] Junta 3: -43 a 173 [°] Junta 4: -90 a 90 [°] Junta 5: -90 a 90 [°]
Comunicação	RS-485 57.600 [bps]
Tensão de alimentação	24 [V]
Massa	6,4 [kg]

Fonte: Autoria própria.

[Brazilian-Institute-of-Robotics / timon\\_hm\\_manipulator](#) Private

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [...](#)

[master](#) [...](#)

**timon\_hm\_manipulator / README.md**

 **migueelnery** Update README.md [History](#)

 3 contributors

[Raw](#) [Blame](#) [!\[\]\(2d0455478c16d2c6a675ff987f9a7030\_img.jpg\)](#) [!\[\]\(e2e2617290b6350df77eba50388e4441\_img.jpg\)](#)

254 lines (203 sloc) 8.32 KB

 **JeRoTIMON**

JeRoTIMON is a manipulator that is being designed, simulated and built in order to meet demands related to the recognition of visual markers and in response to actuate switches, keys or buttons. These buttons and markers can be viewed horizontally or vertically.



**Open Source software and packages related**

## Software

- ROS Melodic
- MoveIt

## Packages

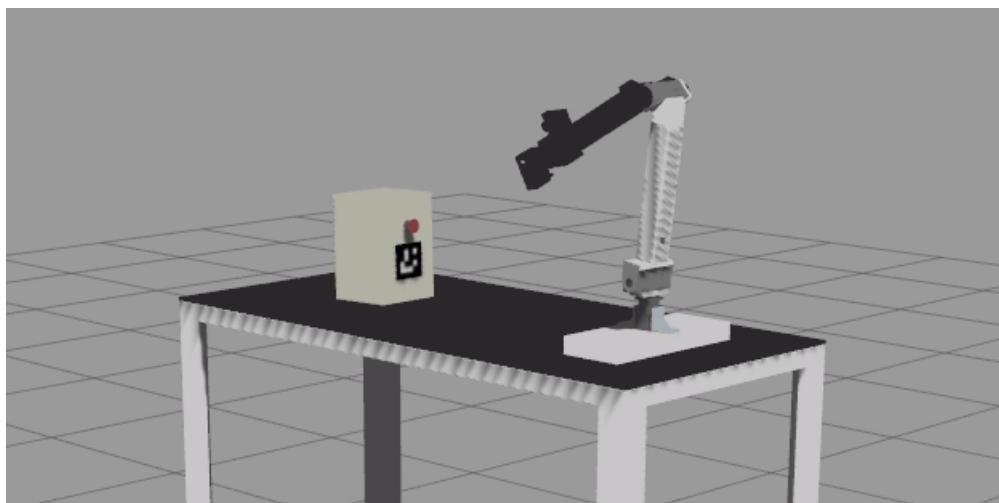
- BIR Marker
- def\_cam\_teledyne\_nano
- OpenCV 3

# User manual

---

## Simulate JeRoTIMON

---



### Install softwares and libraries:

- Install ROS Melodic on Ubuntu 18.04
- Intall OpenCV 3.3.1
- Install MoveIt

### Install ROS melodic packages

```
$ sudo apt install ros-melodic-ros-control ros-melodic-gazebo-ros-control ros-melodic-controller-manager ros-melodic-joint-trajectory-controller ros-melodic-joint-state-controller ros-melodic-position-controllers ros-melodic-trac-ik-kinematics-plugin

$ sudo apt-get install ros-melodic-moveit-visual-tools
```

### Setup workspace

```
$ mkdir -p catkin_ws/src  
$ cd catkin_ws/src  
  
For simulation:  
$ git clone -b feature/simulation https://github.com/Brazilian-Institute-of-  
Robotics/timon_hm_manipulator.git  
For real manipulator:  
$ git clone https://github.com/Brazilian-Institute-of-  
Robotics/timon_hm_manipulator.git  
$ git clone -b final_settings https://github.com/Brazilian-Institute-of-  
Robotics/bir_marker_localization.git  
$ git clone https://github.com/ROBOTIS-GIT/open_manipulator_msgs.git  
  
$ cd ..  
  
$ catkin_make
```

## Run application

### Gazebo simulation

```
$ roslaunch manipulator_gazebo gazebo.launch
```

### Moveit

```
$ roslaunch manipulator_gazebo moveit_demo.launch
```

### Marker localization

```
$ roslaunch timon_demo bir_marker_localization.launch
```

### Execute mission

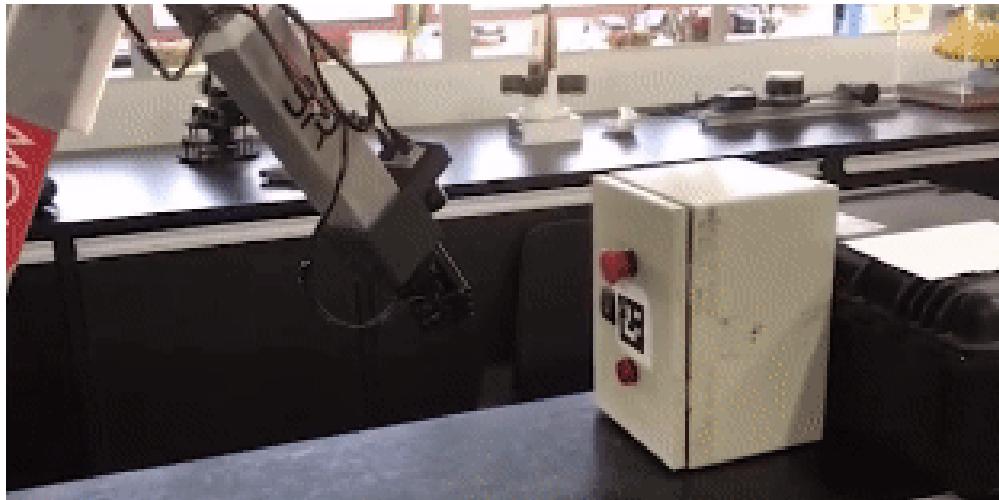
```
$ roslaunch timon_demo push_button_simulation.launch
```

## How to change a box orientation

For change the box orientation, go to *manipulator\_gazebo/launch/spawn\_box.launch* and change *box\_vertical.urdf.xacro* for *box\_horizontal.urdf.xacro*.

## Operate JeRoTIMON manipulator

---



Install the same software and packages than simulation and download and install the DALSA framework for Ubuntu. You need to sign up on their website to download it.

## Setup workspace

```
$ mkdir -p catkin_ws/src  
$ cd catkin_ws/src  
  
$ git clone https://github.com/Brazilian-Institute-of-  
Robotics/timon_hm_manipulator.git  
$ git clone -b final_settings https://github.com/Brazilian-Institute-of-  
Robotics/bir_marker_localization.git  
$ git clone https://github.com/ROBOTIS-GIT/open_manipulator_msgs.git  
$ git clone https://github.com/ROBOTIS-GIT/dynamixel-workbench.git  
$ git clone -b refactor_code https://github.com/Brazilian-Institute-of-  
Robotics/def_cam_teledyne_nano.git
```

Copy [this package](#) to dynamixel-workbench folder.

```
$ cd ..  
$ catkin_make
```

## Run application

### Controllers

```
$ roslaunch timon_arm_controller dxl_controllers.launch  
  
$ roslaunch timon_arm_controller moveit.launch  
  
$ roslaunch timon_arm_controller dxl_moveit_bridge.launch
```

### Camera

```
$ roslaunch def_cam_teledyne_nano camera_example.launch
```

#### Marker localization

```
$ roslaunch timon_demo bir_marker_localization.launch
```

#### Execute mission

```
$ roslaunch timon_demo push_button_real.launch
```

## JeRoTIMON specifications

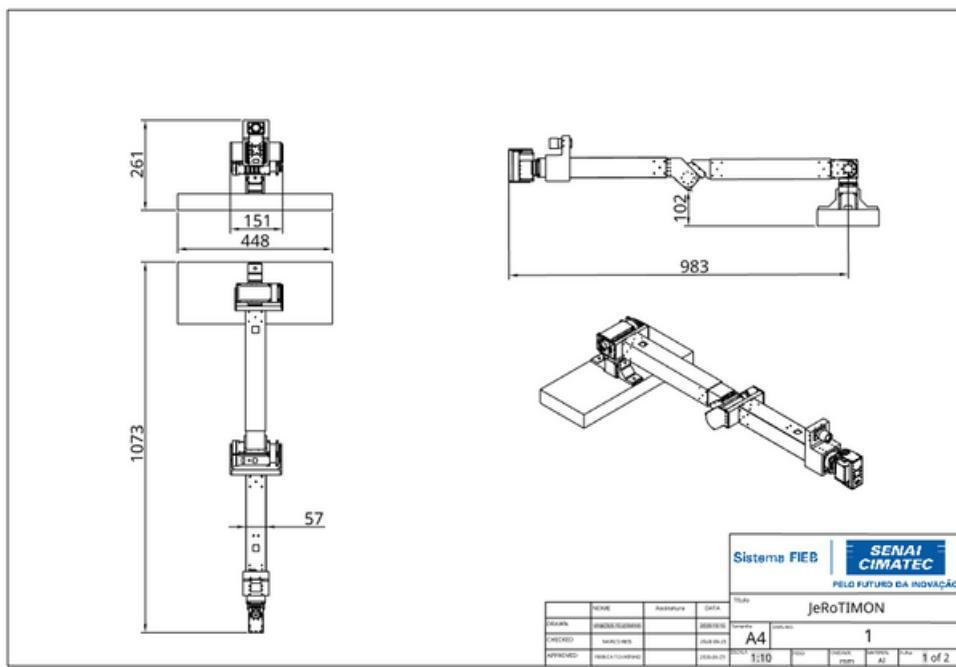
---

### Specifications table

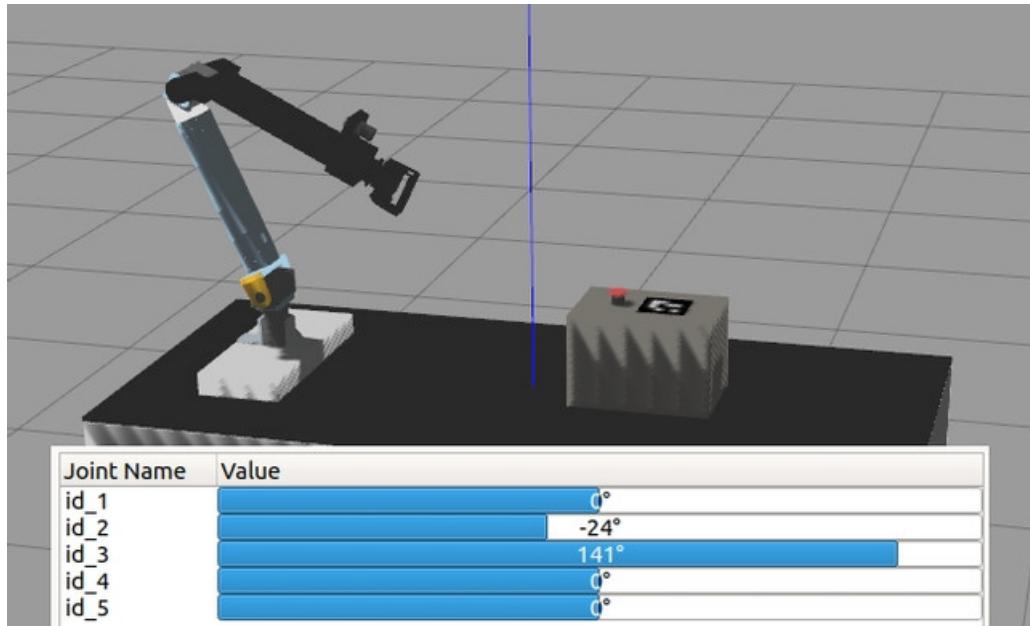
Item	JeRoTIMON
DOF	5
Payload	2 kg
Reach	981 mm
Weight (with base)	10 kg
Weight (without base)	6.4 kg
Operating voltage	24 V
Resolution	Joint 1, Joint 2, Joint 3, Joint 4: 1,003,846 pulse/rev Joint 5: 607,500 pulse/rev
Motors	Joint 1, Joint 2, Joint 3 : PH54-200-S500-R (200W) Joint 4: PH54-100-S500-R (100 W) Joint 5: PH42-020-S300-R (20 W)
Operating Range	Joint 1: -45° ~ 45° Joint 2: -90° ~ 90° Joint 3: -43° ~ 173° Joint 4: -90° ~ 90° Joint 5: -90° ~ 90°
Camera	Teledyne Genie Nano C290
Position sensor type	Homing: Absolute Encoder Control: Incremental Encoder
Communication	USB

Item	JeRoTIMON
Electrical pattern	RS485
Communication Baudrate	57,600 bps

## Dimension



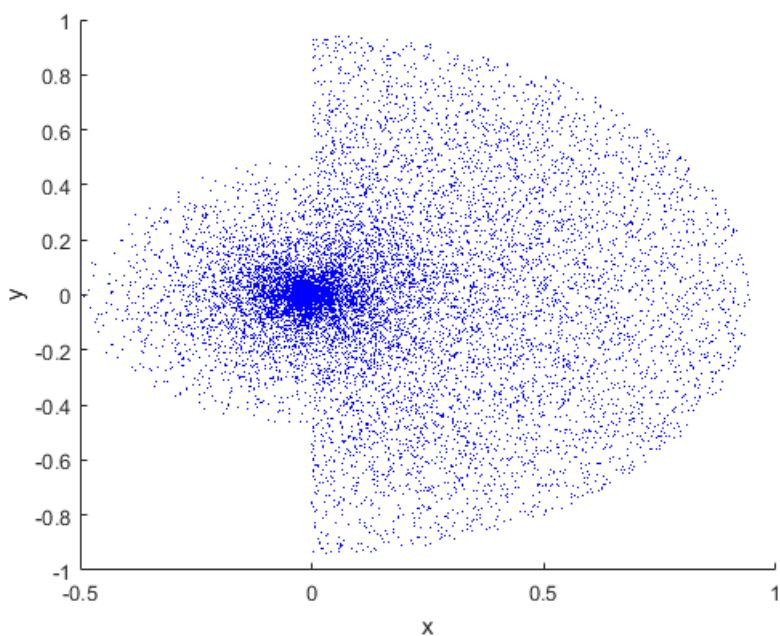
## Home Position



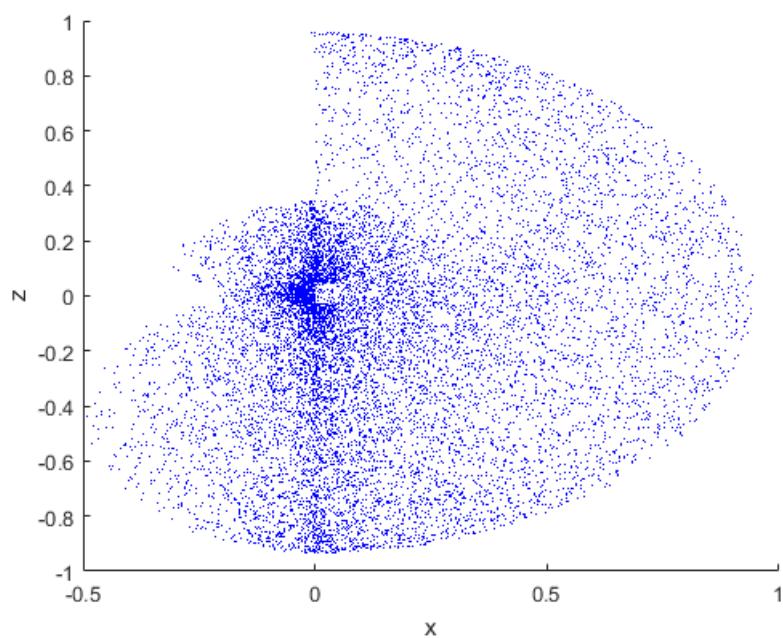
## Workspace

---

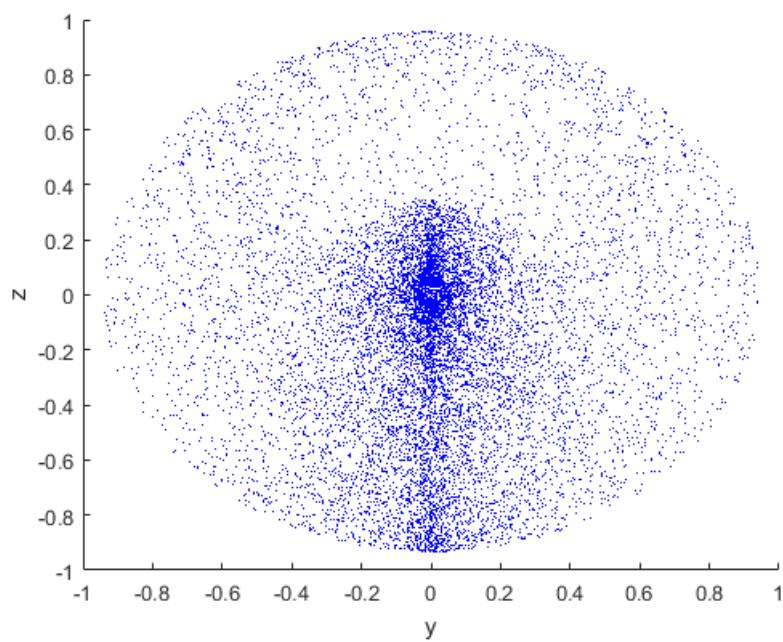
XY



XZ

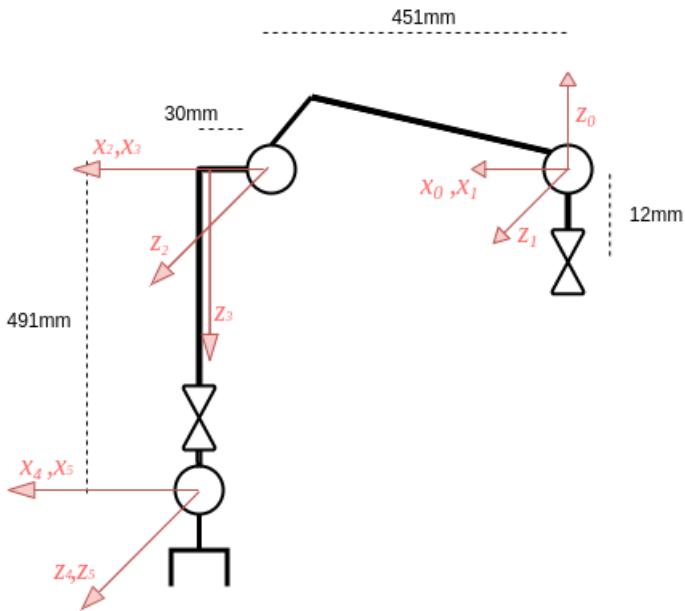


YZ



DH Configuration

---

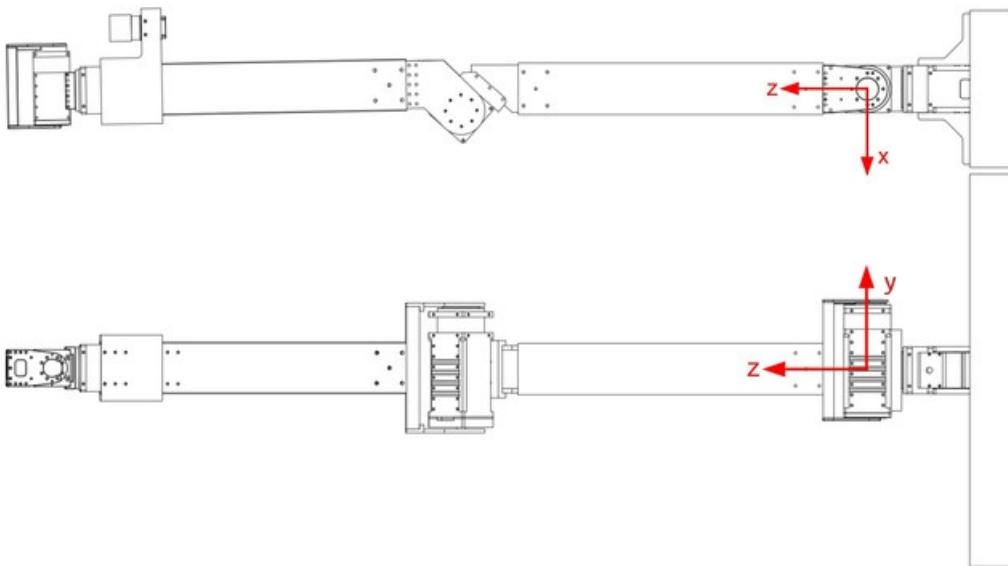


### DH parameters

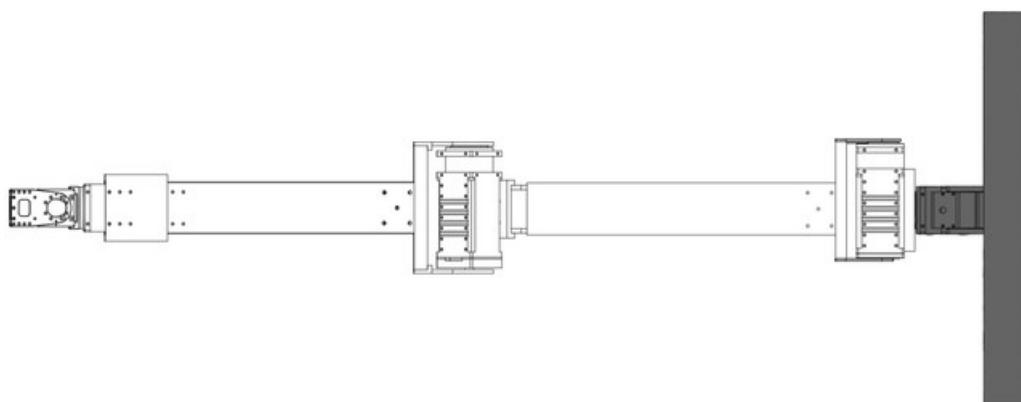
LINK	Link Length (mm)	Link Twist (rad)	Joint Offset (mm)	DXL Angle (rad)
1	0	$\pi/2$	12	0
2	451		0	$\frac{\pi}{2} - \tan^{-1}\left(\frac{30}{451}\right)$
3	30	$-\pi/4$	0	$\frac{\pi}{4} + \tan^{-1}\left(\frac{30}{451}\right)$
4	0	$\pi/2$	491	0
5	0	0	0	0

### Mass Property

#### Coordinate

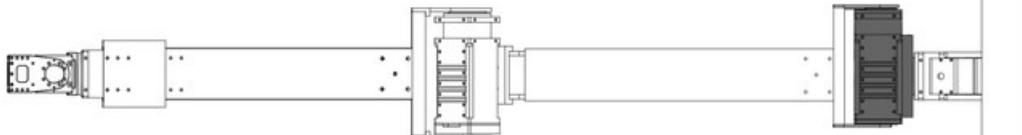


### Link 0



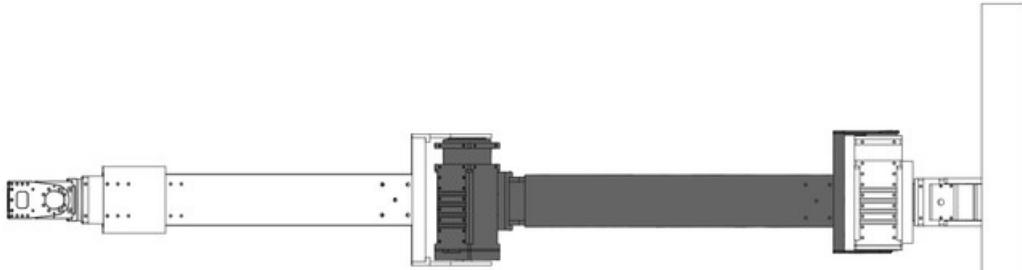
- **Mass:** 3.72384654 kg
- **Volume:** 0.00413037 m<sup>3</sup>
- **Surface area:** 0.28622123 m<sup>2</sup>
- **Center of mass:**
  - X: -0.00000147 m
  - Y: 0.00000000 m
  - Z: 0.03504069 m
- **Moments of inertia:** kg m<sup>2</sup>
  - L<sub>xx</sub>: 0.01072644 L<sub>xy</sub>: -9.465e-9 L<sub>xz</sub>: -3.247e-8
  - L<sub>yx</sub>: -9.465e-9 L<sub>yy</sub>: 0.04865651 L<sub>yz</sub>: 6.830e-13
  - L<sub>zx</sub>: -3.247e-8 L<sub>zy</sub>: 6.830e-13 L<sub>zz</sub>: 0.05388213

### Link 1



- **Mass:** 1.03781084 kg
- **Volume:** 0.00040169 m<sup>3</sup>
- **Surface area:** 0.05853078 m<sup>2</sup>
- **Center of mass:**
  - X: 0.00814457 m
  - Y: 4.45597047e - 8 m
  - Z: 0.16022275 m
- **Moments of inertia: kg m<sup>2</sup>**
  - L<sub>xx</sub>: 0.0005687 L<sub>xy</sub>: 2.602e-10 L<sub>xz</sub>: -0.00004027
  - L<sub>yx</sub>: 2.602e-10 L<sub>yy</sub>: 0.00166759 L<sub>yz</sub>: -2.222e-10
  - L<sub>zx</sub>: -0.00004027 L<sub>zy</sub>: -2.222e-10 L<sub>zz</sub>: 0.00155695

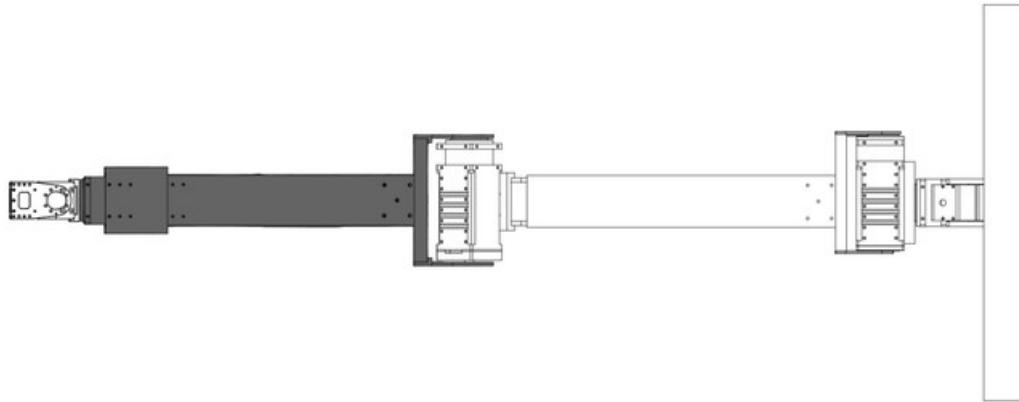
## Link 2



- **Mass:** 2.05026959 kg
- **Volume:** 0.00077667 m<sup>3</sup>
- **Surface area:** 0.29790369 m<sup>2</sup>
- **Center of mass:**
  - X: 0.00294129 m
  - Y: -0.01058166 m
  - Z: 0.48960322 m

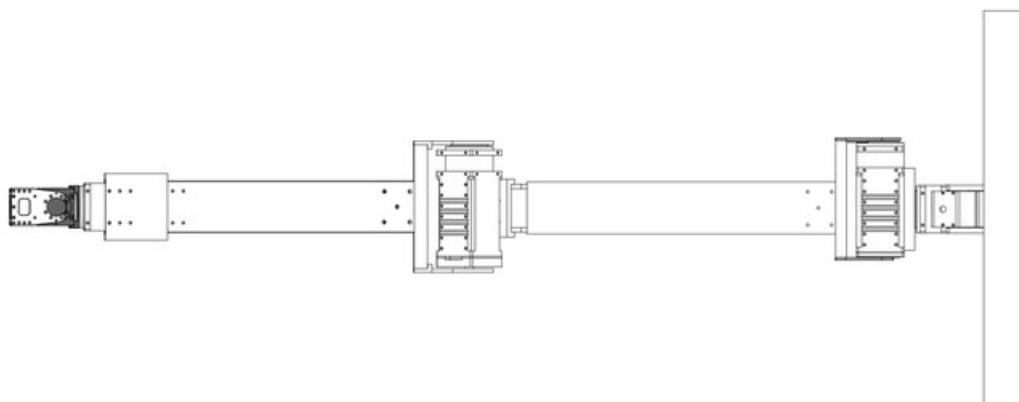
- **Moments of inertia: kg m<sup>2</sup>**
  - **L<sub>xx</sub>**: 0.06174738 **L<sub>xy</sub>**: -0.00003476 **L<sub>xz</sub>**: 0.00098626
  - **L<sub>yx</sub>**: -0.00003476 **L<sub>yy</sub>**: 0.06319525 **L<sub>yz</sub>**: 0.00311701
  - **L<sub>zx</sub>**: 0.00098626 **L<sub>zy</sub>**: 0.00311701 **L<sub>zz</sub>**: 0.0034029

### Link 3



- **Mass:** 1.83580505 kg
- **Volume:** 0.00081683 m<sup>3</sup>
- **Surface area:** 0.32055793 m<sup>2</sup>
- **Center of mass:**
  - X: 0.00257384 m
  - Y: 0.00207951 m
  - Z: 0.91287054 m
- **Moments of inertia: kg m<sup>2</sup>**
  - **L<sub>xx</sub>**: 0.03439469 **L<sub>xy</sub>**: -0.00000261 **L<sub>xz</sub>**: -0.00001194
  - **L<sub>yx</sub>**: -0.00000261 **L<sub>yy</sub>**: 0.03489631 **L<sub>yz</sub>**: -0.0001288
  - **L<sub>zx</sub>**: -0.00001194 **L<sub>zy</sub>**: -0.0001288 **L<sub>zz</sub>**: 0.0023687

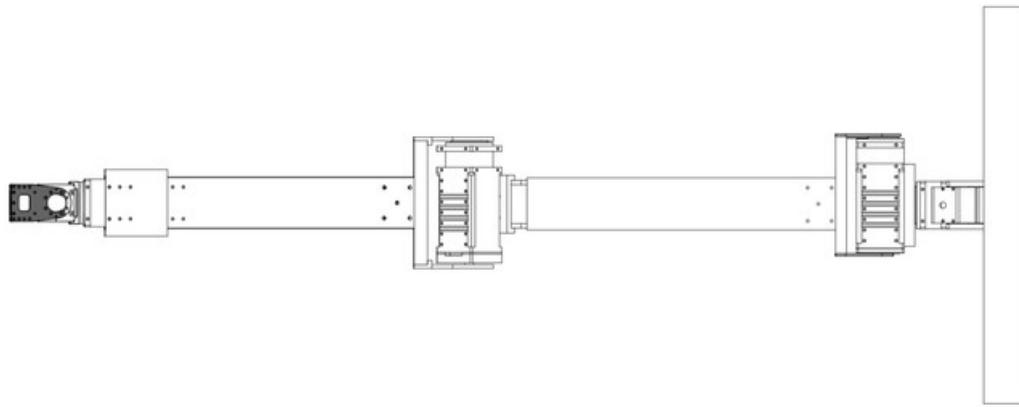
### Link 4



- **Mass:** 0.39425655 kg

- **Volume:** 0.00014602 m<sup>3</sup>
- **Surface area:** 0.03018022 m<sup>2</sup>
- **Center of mass:**
  - X: 0.00257156 m
  - Y: 0.00133073 m
  - Z: 1.09673426 m
- **Moments of inertia: kg m<sup>2</sup>**
  - L<sub>xx</sub>: 0.00028795 L<sub>xy</sub>: 2.010e-10 L<sub>xz</sub>: -8.166e-13
  - L<sub>yx</sub>: 2.010e-10 L<sub>yy</sub>: 0.00011981 L<sub>yz</sub>: -0.00000446
  - L<sub>zx</sub>: -8.166e-13 L<sub>zy</sub>: -0.00000446 L<sub>zz</sub>: 0.0002842

## Link 5



- **Mass:** 0.08694786 kg
- **Volume:** 0.0000322 m<sup>3</sup>
- **Surface area:** 0.02435944 m<sup>2</sup>
- **Center of mass:**
  - X: 0.00257155 m
  - Y: 0.00684212 m
  - Z: 1.13578562 m
- **Moments of inertia: kg m<sup>2</sup>**
  - L<sub>xx</sub>: 0.00012963 L<sub>xy</sub>: -5.235e-13 L<sub>xz</sub>: -9.996e-14
  - L<sub>yx</sub>: -5.235e-13 L<sub>yy</sub>: 0.00003703 L<sub>yz</sub>: 0.00000247
  - L<sub>zx</sub>: -9.996e-14 L<sub>zy</sub>: 0.00000247 L<sub>zz</sub>: 0.00012613

## A.9 Conversor ROBOTIS U2D2 (RS485 - USB)

A ficha resumo contendo as principais especificações do conversor de sinais ROBOTIS U2D2 está disposta na Tabela 10. Maiores informações sobre sua utilização podem ser encontradas no *datasheet* anexado a seguir. O desenho mecânico deste conversor esta disposto na Figura 29.

Tabela 10: Especificações do conversor ROBOTIS U2D2 (RS485-USB).

Item	Especificação
Massa	9 [g]
Dimensões (L x A x P)	48 x 18 x 14.6 [mm]
Portas de comunicação com computador	USB
Portas de comunicação com periféricos	3 Pinos nível TTL 4 Pinos RS-485 4 Pinos UART
Taxa de transmissão	Máximo 6 [Mbps]

Fonte: Autores, adaptado de ROBOTIS.

**ROBOTIS e-Manual**

[Edit on GitHub](#)

[G](#)

ToC

TOP

## 1. Introduction



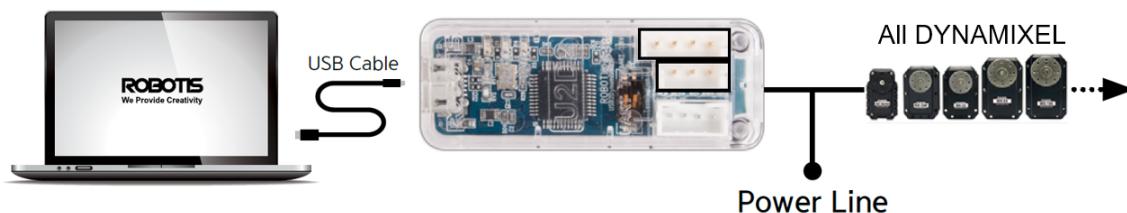
**U2D2**

U2D2 is a small size USB communication converter that enables to control and operate DYNAMIXEL with PC.

## 2. How to Use

### 2. 1. PC to DYNAMIXEL

U2D2 can be connected to the USB port of the PC with the enclosed USB cable. It supports both 3Pin TTL connector and 4Pin RS-485 connector to link up with various DYNAMIXEL's. U2D2 does not supply power to DYNAMIXEL, therefore, an external power supply should provide power to DYNAMIXEL. Be careful as recommended voltage for each DYNAMIXEL could be vary by model. Please refer to e-Manual for recommended supply voltage and power connection with various power supplies.



```
graph LR; Laptop[ROBOTIS] --- USB[USB Cable]; USB --- U2D2[U2D2]; U2D2 --- PowerLine((Power Line)); PowerLine --- AllDynamixels[All DYNAMIXEL]
```

### 2. 2. PC to UART

U2D2 is able to connect ROBOTIS controllers that support 4Pin UART communication such as OpenCM9.04, CM-150 and CM-200 to the PC with the enclosed USB cable.



4Pin UART Communication  
Supported Controllers



ToC

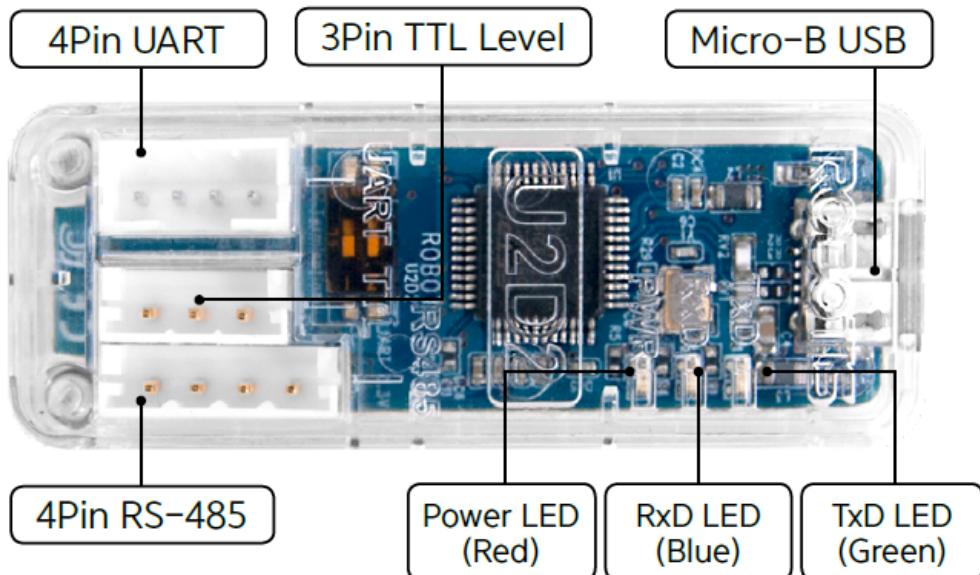
▲  
TOP

### 3. Specifications



Item	Description
Weight	9g
Dimensions	48mm x 18mm x 14.6mm
Ports	3Pin TTL Level(TTL Communication supported DYNAMIXEL) 4Pin RS-485(RS-485 Communication supported DYNAMIXEL) 4Pin UART(Controllers that support 4Pin UART(such as CM-150 and CM-200))
Baudrate	Maximum 6Mbps

### 4. Layout



G  
ToC  
▲ TOP

Ports	Description
4Pin UART	Convert USB and UART
3Pin TTL Level	Connect to DYNAMIXEL with 3Pin TTL Level Communication
4Pin RS-485	Connect to DYNAMIXEL with 4Pin RS-485 Communication
Status LED	Display status of Power supply, TxD(Data write) and RxD(Data Read)
Micro-B USB	Connect to the PC with USB cable

## 5. Baudrate

Reference BPS	Actual BPS	Error(%)
9,600	9,600	0.00
57,600	57,588.4823	-0.02
115,200	115,246.0984	0.04
1,000,000	1,000,000	0.00
2,000,000	2,000,000	0.00
3,000,000	3,000,000	0.00
4,000,000	4,000,000	0.00
4,500,000	4,571,428.571	1.56
6,000,000	6,000,000	0.00

**NOTE :** Less than 3% of the baud rate error will not affect to the UART communication.

**NOTE :** For the stable communication with higher baudrate, configure USB Latency value to the lower.  
[USB Latency Setting](#)

## 6. Supply Power

U2D2 does not supply power to DYNAMIXEL, therefore, an external power supply should provide power to DYNAMIXEL as below. Before connecting DYNAMIXEL to an external power supply, please check the recommended voltage for DYNAMIXEL.



ToC



TOP

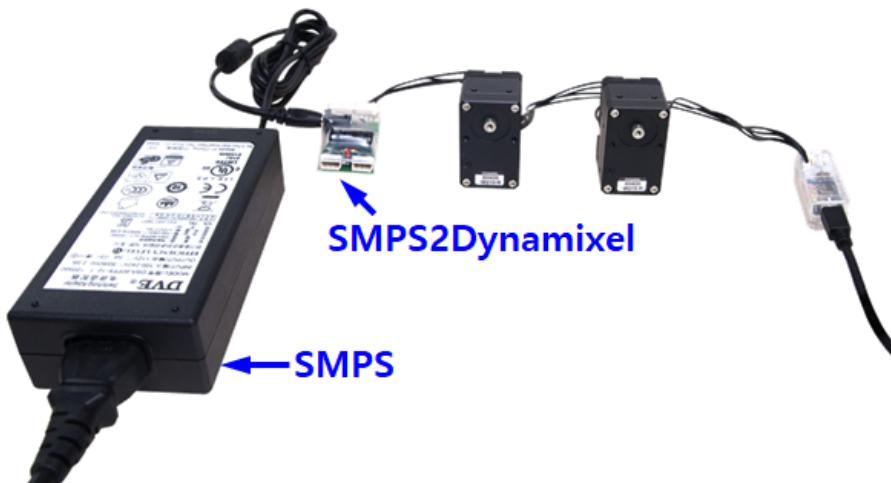
Caution for Power Supply

- For a stable power supply, it is recommended to use ROBOTIS controller or SMPS2Dynamixel.
- When connecting cable to DYNAMIXEL, please turn off the power.

### 6. 1. Using SMPS2Dynamixel

In order to operate DYNAMIXEL with SMPS2Dynamixel, please connect DYNAMIXEL to SMPS2Dynamixel, then connect SMPS to SMPS2Dynamixel as shown below image.

Please compare operating voltage of DYNAMIXEL with 12V 5A ROBOTIS SMPS before supplying power.



### 6. 2. Using ROBOTIS Controller

In order to operate DYNAMIXEL with OpenCM 485 Expansion board, please connect DYNAMIXEL to OpenCM485 Exp board, then connect battery or SMPS to the expansion board as shown below image.

Please compare operating voltage of DYNAMIXEL with battery or 12V 5A ROBOTIS SMPS before supplying power.

OpenCM485 EXP board will bypass the input power source to output power.



### 6. 3. Using Power Supply

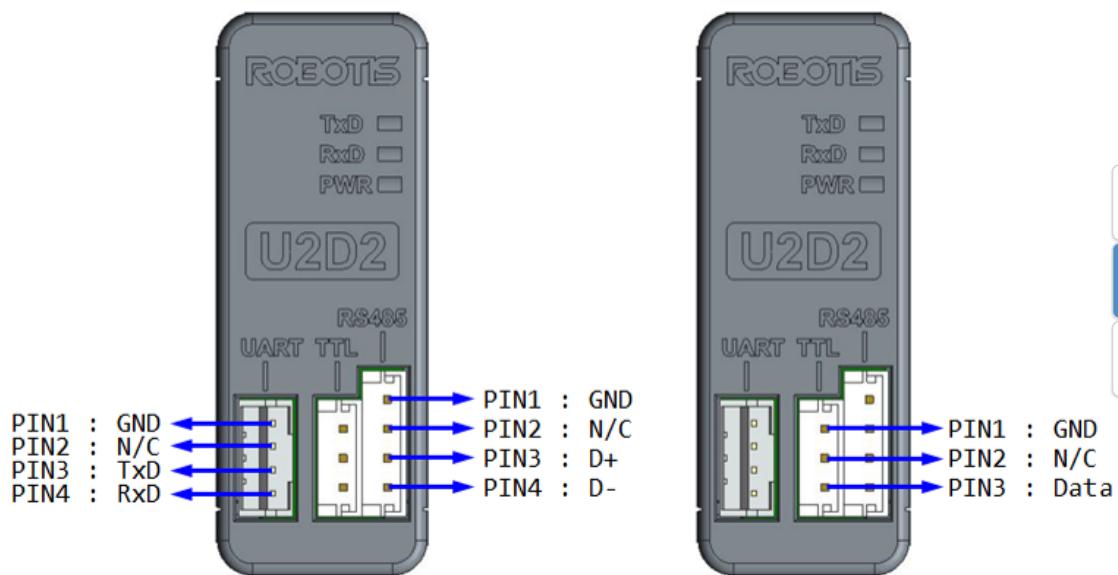
In order to operate DYNAMIXEL with Power Supply, please configure the power supply for DYNAMIXEL before connecting cable.

Please check the pinout of the cable to avoid connecting to wrong pins.

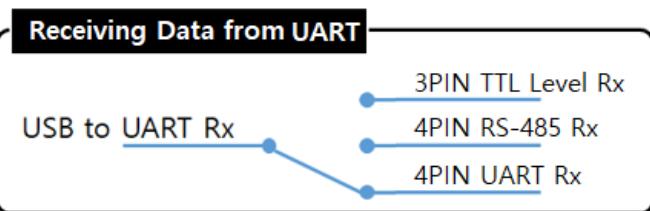
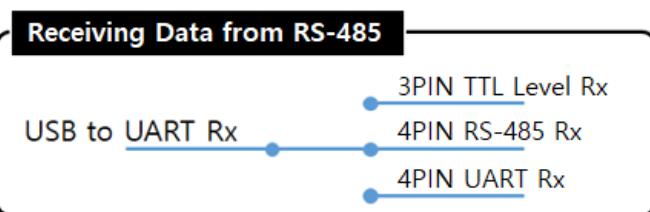
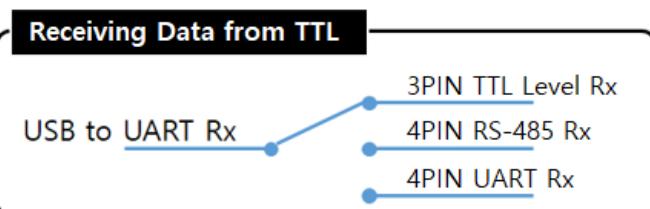
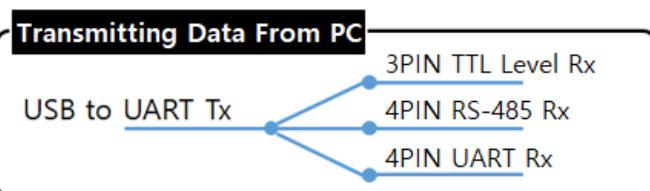
How to check whether the USB downloader(LN-101) driver is installed correctly.



### 7. Pinouts



## 8. Data Flow Control





ToC

TOP

When data is transmitted from the USB port in PC to peripheral devices(Tx), all connected device will receive an identical data.

On the other hand, when data is received from one of the peripheral devices(Rx), it will be automatically converted as USB communication and sent to the PC. During this process, received data will not be transmitted to other peripheral devices. If data from more than two peripheral devices are received, the received data can be corrupted.

Therefore, when connecting 3Pin TTL Level DYNAMIXEL and 4Pin RS-485 DYNAMIXEL at the same time, all connected DYNAMIXEL should have its unique ID in order to prevent receiving corrupt data. In addition, when using Bulk Read and Sync Read instructions that requires to receive data from multiple DYNAMIXEL's, relevant DYNAMIXEL's should be connected to the same communication channel or else the instruction will not work properly.

## 9. Softwares

### 1. RoboPlus

RoboPlus supports Windows environment to manage controllers and DYNAMIXEL's.

- R+ Task 2.0 : R+ Task 2.0 can be used when downloading Task Code to the controller.
- R+ Motion 2.0 : R+ Motion 2.0 can be used when downloading Motion file to the controller.
- R+ Manager 2.0 : R+ Manager 2.0 is used to connect controllers or DYNAMIXEL's and update firmware or test the control table..
- [ROBOTIS Homepage](#)

### 2. DYNMAIXEL Workbench

DYNMAIXEL Workbench supports Linux environment to test control table of DYNAMIXEL.

- [ROBOTIS GitHub](#)
- [DYNMAIXEL Workbench Manual](#)

### 3. DYNAMIXEL SDK

DYNAMIXEL SDK supports various programming languages such as C, C++, C#, MATLAB, LabVIEW, Python and Java in order to help developing DYNAMIXEL control software. DYNAMIXEL SDK provides a bundle of functions/methods that can be used for operating DYNAMIXEL with protocols in the form of digital packet.

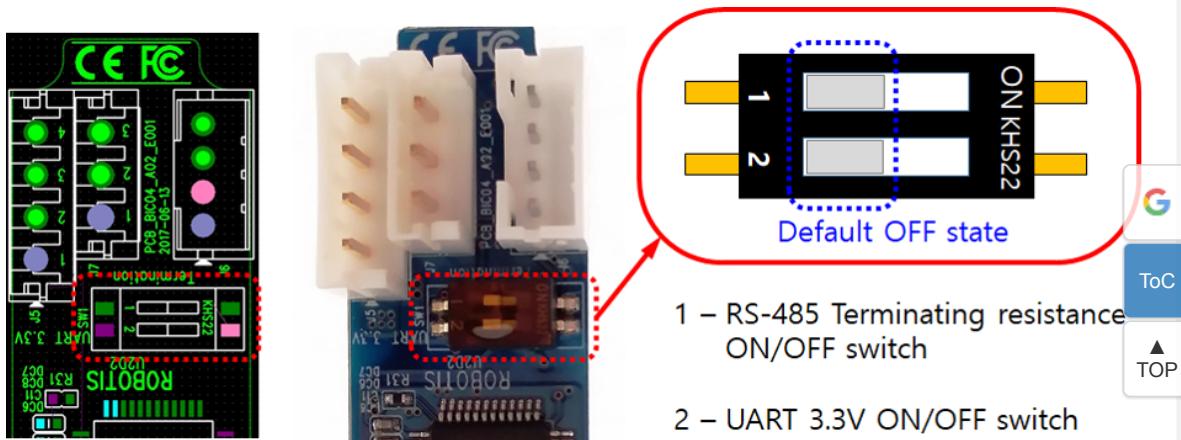
DYNAMIXEL SDK is free to download from GitHub.

- [ROBOTIS GitHub](#)
- [DYNAMIXEL SDK Manual](#)

## 10. Additional Features

### 10. 1. Internal DIP Switch

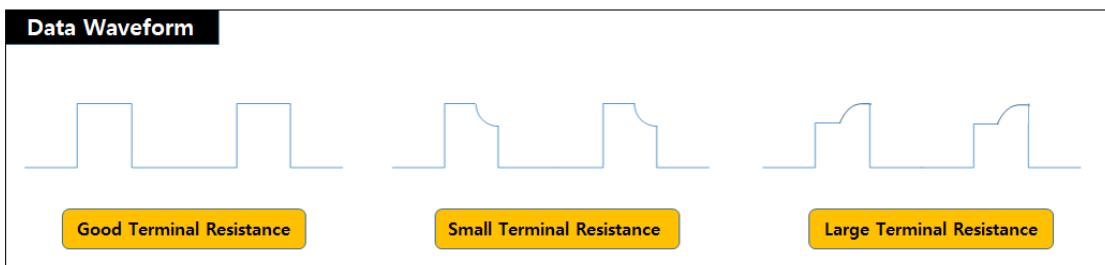
There is a two-pole switch inside of the plastic cover. This switch is set to OFF as a default. Under normal circumstances this default setting is recommended, therefore, the switch cannot be accessed without opening the case. Please open the case and set RS-485 termination resistor and UART 3.3V power switch only after thoroughly understand its usage.



### 10. 1. 1. RS-485 Termination Resistor( $120\Omega$ ) Switch

Transmission devices are designed to have certain impedances based on design standard and signal reflection or interference can occur when impedance do not match over the transmission lines. The termination resistors at the end of transmission lines suppress such effect by adjusting impedance and allowing constant current flow. However, the termination resistor is not a mandatory, especially in a short RS-485 network with lower communication baud rate.

If communication is suffering from frequent errors, the termination resistor might be required. The regular  $120\Omega$  termination resistor will be applied on the transmission line by shifting the switch to ON. If termination resistor doesn't resolve the communication issue, the resistance value might need to be adjusted.



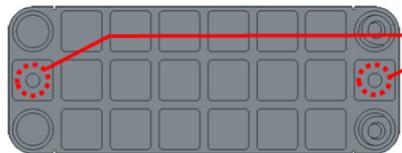
### 10. 1. 2. UART 3.3V Power Supply Switch

The peripheral device such as controllers that can be connected to the 4Pin UART usually have an external power source, therefore additional 3.3V power from the UART port #2 pin is not necessary.

However, when connecting BT-410 that does not have an external power source needs to be powered by the 4Pin UART port #2 pin. In this case, the UART 3.3V power switch need to be turned on.

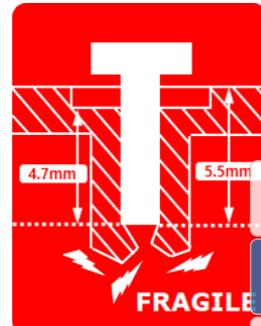
If the switch is turned on, second pin of the 4Pin UART connector will be able to supply 3.3V.

## 10. 2. Caution for Assembly



Mount Holes

The maximum depth of the available tap is 4.7mm. When tightening the bolts to a depth of more than 4.7mm may cause damage.



- Bolt for Mount Hole : M2 Tap bolt
- Mount Hole Depth : Valid depth 4.7mm, Depth from the surface 5.5mm

## 11. References

### 11.1. Certifications

Please inquire us for information regarding unlisted certifications.

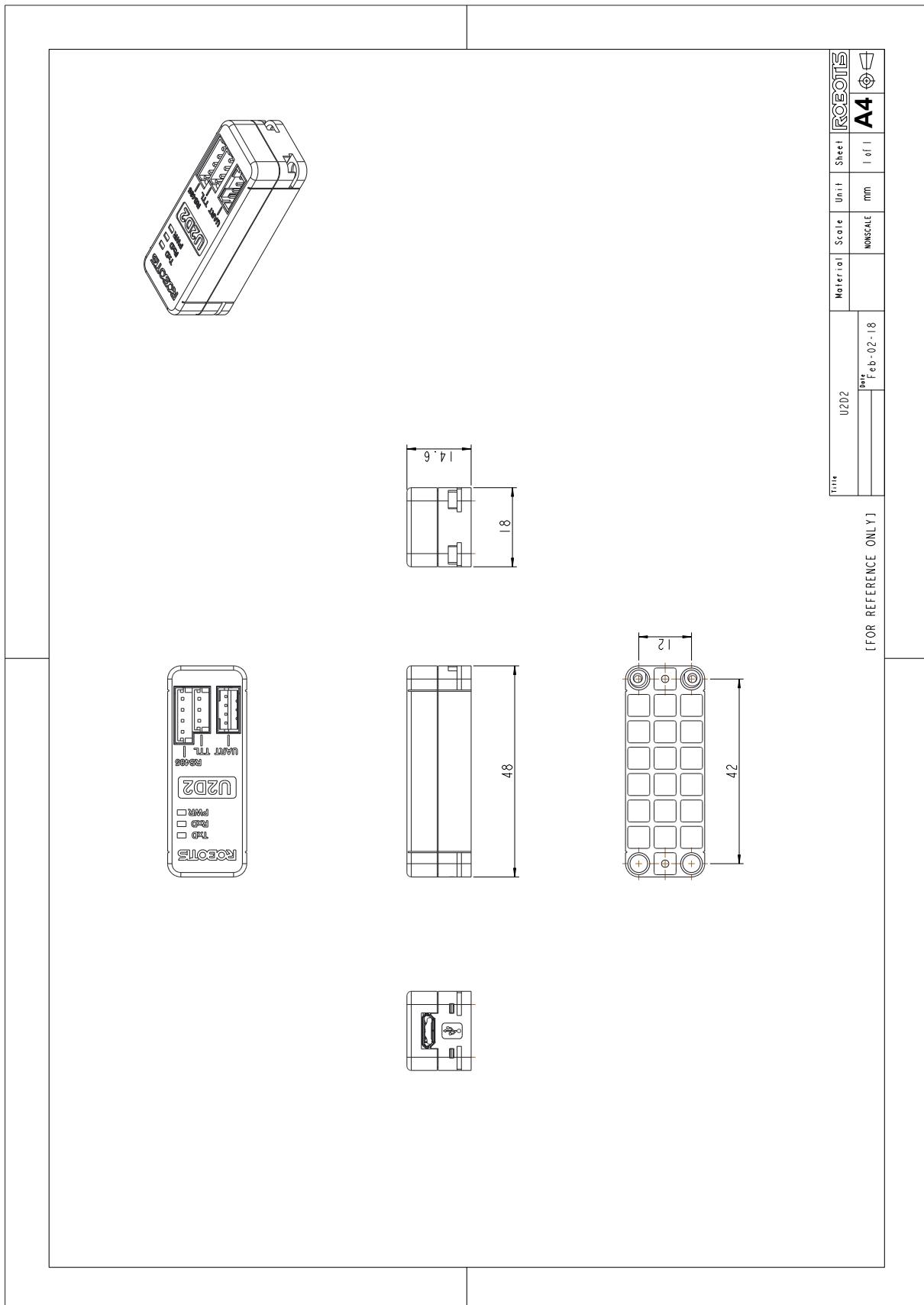
#### 11.1.1. FCC

**Note:** This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

#### WARNING

Any changes or modifications not expressly approved by the manufacturer could void the user's authority to operate the equipment.

Figura 29: Desenho mecânico do conversor ROBOTIS U2D2 (RS485 - USB).



Fonte: ROBOTIS.

---

## **Estudo estatístico - Análise de Regressão**

---

# Estudo estatístico Análise de Regressão

Anderson Queiroz  
Rodrigo Formiga  
Jean Silva  
Aziel Freitas

Senai CIMATEC. CCRoSA - Centro de Competência em Robótica e  
Sistemas Autônomos  
anderson.vale@fbter.org.br  
rodrigo.farias@fbter.org.br  
jean.silva@fbter.org.br  
aziel.freitas@fbter.org.br

22 de setembro de 2020

# 1 Introdução

O experimento apresentado neste relatório consistiu em avaliar a influência de dois parâmetros básicos no tempo de queda de alguns modelos de “helicóptero” feito em dobraduras de papel.

## 1.1 Six Sigma

A filosofia *Six Sigma* (ou Seis Sigma) é uma estratégia em que você define rotinas e desenvolve trabalhos de melhoria de processos em sua organização [COUTINHO, 2019]. Este método foi originado pelo programa de melhoria da qualidade da Motorola em 1987 com o objetivo de se aproximar do zero defeito. Com o sucesso alcançado pela metodologia na empresa, o processo foi difundido [CORRÊA, 2007]. *Six Sigma* trabalha com três grandes objetivos: redução de custos, otimização de processos e satisfação do cliente [COUTINHO, 2019].

Tendo em vista essa informação, existe uma escala que determina a qualidade do processo:

Nível da qualidade	Defeitos por milhão	% de conformidade
1 Sigma	691.463	30,85%
1,5 Sigma	500.000	50%
2 Sigma	308.537	69,15%
3 Sigma	66.807	93,32%
4 Sigma	6.210	99,38%
5 Sigma	233	99,97%
<b>6 Sigma</b>	<b>3,4</b>	<b>99,99966%</b>

Figura 1: Tabela *Six Sigma* [COUTINHO, 2019]

A partir da tabela, podemos perceber o espectro de algumas empresas, que trabalham em produção de milhões de unidades, então obter somente 99% de confiabilidade não é o bastante. O processo é definido como 6 Sigma ao atingir uma proporção de 3 ou 4 erros a cada 1 milhão de unidades produzidas. Algumas empresas aceitam trabalhar como 4 Sigma em alguns dos seus processos.

## 1.2 Otimização de Processos

A otimização de respostas em um processo é fundamental para o *Six Sigma* na fase de melhoria. Uma das ferramentas mais importantes que representa o conceito principal dessa filosofia é a Planejamento de Experimentos (**DoE**). **DoE** busca combinar fatores de um processo com o objetivo de otimizar a resposta desejada extraíndo o máximo possível de informações com um número mínimo de experimentos. [TOLEDO, 2003]

O planejamento e a condução desses testes também fazem parte do **DoE**, visto que a falta de planejamento ou a má condução pode culminar em resultados incoerentes, obrigando a repetição de todo o processo de análise, ou mesmo falsos-positivos. Os testes devem ser projetados de modo que a aleatoriedade esteja sempre presente, em detrimento a qualquer eventual manipulação dos dados, pois ela é uma peça fundamental na normalidade dos experimentos.

## 2 Revisão literária

Serão apresentados na presente seção alguns pontos que, segundo a literatura, nortearão o estudo estatístico do trabalho em questão, abordando alguns pontos importantes para realização do planejamento e análise dos experimentos. Logo em seguida, serão aplicados os conceitos aqui apresentados.

### 2.1 Planejamento e análise de experimentos

Técnicas de planejamento e análises de experimentos são utilizáveis em empresa, processo de fabricação, desenvolvimento de um novo produto e até em fabricação de commodities. Geralmente a maioria destes processos possuem diversas variáveis controláveis, como temperatura, velocidade, dimensões entre outras. A variação destas variáveis, de modo isolado ou combinado, pode trazer melhorias ou prejuízos a qualidade do produto ou serviço. Com isso, faz-se necessário um estudo analítico de como cada variável pode influenciar o processo [MONTGOMERY, 2012].

O planejamento e análise de experimentos (*Design of Experiments, DoE*) é uma técnica de planejamento de experimentos, ou seja, como os experimentos devem proceder para garantir conclusões assertivas. A validade das conclusões obtidas após análise dos experimentos, são diretamente ligadas ao modo como os experimentos foram realizados, logo, o planejamento do experimento torna-se o ponto mais importante na hora de se analisar algum processo produtivo, bens ou serviços prestados. O planejamento corretamente executado também pode evitar desperdícios, além de isolar e determinar relações das variáveis [MONTGOMERY, 2012].

A seguir serão apresentados alguns pontos importantes para análise estatística realizada neste presente trabalho, como: Princípios do planejamento, que irão nortear como deve ser condizido o planejamento de experimentos, bem como etapas para realização dos experimentos e análise dos mesmos [TAHARA, 2008].

#### 2.1.1 Princípios do planejamento de experimentos

O planejamento de experimentos possui três princípios, que são: replicação, aleatoriedade e blocagem [TAHARA, 2008].

##### Replicação

A replicação consiste na obtenção de mais de uma unidade experimental, ou seja, replicar o teste para cada ponto experimental e assim permite que obtenha-se uma estimativa mais precisa, diminuindo a influência de variações indesejadas ou inevitáveis [TAHARA, 2008].

## Aleatoriedade

Os métodos estatísticos requerem que as observações sejam variáveis aleatórias, de modo a garantir a distribuição igual dos fatores não esperados. Com isso garantem-se estimativas não tendenciosas dos efeitos e erros experimentais, bem como evitar influência sistemática de fatores não controláveis [TAHARA, 2008].

## Blocagem

A blocagem é uma técnica extremamente importante, utilizada com o objetivo de aumentar a precisão de um experimento. Este princípio pode ser aplicado em alguns casos onde deseja-se isolar algum fator conhecido, analisando de forma individual sua influência sobre o processo. A mudança de pessoas no processo experimental ou a mudança de lote de um produto pode ser visto a princípio como um fator não desejado de ser analisado, logo, o princípio de blocagem permite que o mesmo seja isolado, porém, não ignorado[TAHARA, 2008].

### 2.1.2 Etapas para o desenvolvimento de experimentos

Coleman e Montgomery (1993) propõem as seguintes etapas para o desenvolvimento de um planejamento de experimentos [COLEMAN, 1993]:

1. **Caracterização do problema:** A definição do problema que está sendo analisado é uma etapa essencial para se entender o estudo analítico. Para isso se faz necessário conhecimento sobre todo o processo para assim definir de forma clara o objetivo e relatar de forma específica o problema analisado.
2. **Escolha dos fatores de influência e níveis:** Para conduzir o experimento deve-se escolher os fatores variáveis, os intervalos sobre os quais esses fatores variarão e os níveis específicos. Para tanto é necessário conhecimento do processo, experiências práticas e teóricas para determinar tais fatores mais relevantes para análise experimental.
3. **Seleção das variáveis de resposta:** É também necessário ao planejar um experimento que se tenha clareza de qual variável-resposta será obtida como parâmetro para qualificar os experimentos. Variável de resposta que não atenda a necessidade do experimento pode levar a conclusões equivocadas.
4. **Determinação de um modelo de planejamento de experimento:** A escolha do planejamento envolve consideração sobre o tamanho da amostra (número de replicações) e determinar se há formação de blocos ou outras restrições de aleatorização.

**5. Condução do experimento:** A condução do experimento deve seguir todo o planejamento, evitando variações de ambiente, metodologia do experimento ou inserção de novas variações não previamente estabelecidas. O responsável pelo experimento e os equipamentos utilizados devem ser mantidos do início ao fim, exceto em casos onde estas variações sejam desejáveis.

**6. Análise dos dados:**

Após todas as demais etapas anteriores, os resultados podem ser obtidos com uso de ferramentas e pacotes estatísticos para auxiliar na visualização de gráficos e dados. A verificação da validade do modelo é também um ponto importante para analisar.

**7. Conclusões e recomendações:**

Após a obtenção dos resultados, o experimento deve apresentar conclusões práticas para proporcionar recomendações de ações em cima do processo analisado. As recomendações são frutos das etapas decorridas no planejamento do experimento, ou seja, planejamentos com falhas irão gerar ações equivocadas no processo.

## 2.2 Análise de regressão

Um método muito utilizado nos estudos estatísticos é a análise de regressão. Este método permite examinar a relação entre duas ou mais variáveis que se deseja-se estudar. O método permite por meio de modelos matemáticos avaliar quais as relações entre as variáveis, quais delas são importantes para o processo e quais apresentam pouca relevância [Action, sd].

As variáveis utilizadas podem ser divididas em duas: Variável Dependente, que é a variável que está sendo estudada, como uma saída do processo (a influência das demais variáveis será analisada através das variáveis dependentes); e Variável Independente, que são as entradas do estudo, são variáveis que supostamente causam impactos ou certa influência na variável dependente [Action, sd].

A análise de regressão tem por finalidade chegar a algumas conclusões e direcionamentos sobre o processo estudado, as finalidades deste estudo são [Action, sd]:

- Predição dos dados;
- Seleção de variáveis influenciáveis no processo;
- Estimação de parâmetros;
- Realizar inferências sobre os parâmetros, como: testes de hipóteses e intervalos de confiança.

### 2.2.1 Tipos de regressão

A análise de regressão é dividida em alguns tipos, tendo cada uma sua própria especificidade, logo, todo analista deve saber qual forma usar, variando sua escolha pelo tipo de dado e sua distribuição. Os tipos mais comuns são: Regressão Linear; Regressão Polinomial; Regressão de Poisson; Regressão de Ridge; Regressão Logística e Mínimos quadrados parciais (PLS) [MARIA, 2018]. Para fins deste estudo será abordado de forma mais aprofundada a regressão linear.

#### Regressão linear

A regressão Linear é um modelo matemático que tem por objetivo observar a relação entre duas ou mais variáveis por meio de uma reta, e utilizar o resultado da função dessa reta para estimar valores e encontrar relações. Quando existe apenas uma variável independente e uma variável dependente, ela é chamada de **regressão linear simples**. Quando existe mais de uma variável independente, é chamado de **regressão linear múltipla**. Na equação de regressão linear múltipla, (equação 1), o 'Y' é a variável dependente, 'X' são as variáveis independentes,  $\beta_i$  são os coeficientes de regressão e  $\epsilon$  é o termo de erro [MARIA, 2018].

$$Y = \beta_1 + \beta_2 X_2 + \beta_3 X_3 + \dots + \beta_i X_i + \epsilon \quad (1)$$

Como exemplo de uma análise de regressão linear no R (figura 2), será analisado os dados a seguir e entender as informações contidas no estudo.

Figura 2: Exemplo de regressão linear no R [MARIA, 2018]

```
(Intercept)          Agriculture      Examination     Education      Catholic
66.9151817       -0.1721140      -0.2580082     -0.8709401    0.1041153
Infant.Mortality  1.0770481
> summary(model)

Call:
lm(formula = Fertility ~ ., data = swiss)

Residuals:
    Min      1Q   Median      3Q      Max 
-15.2743 -5.2617  0.5032  4.1198 15.3213 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 66.91518 10.70604  6.250 1.91e-07 ***
Agriculture -0.17211  0.07030 -2.448 0.01873 *  
Examination -0.25801  0.25388 -1.016 0.31546    
Education   -0.87094  0.18303 -4.758 2.43e-05 ***
Catholic    0.10412  0.03526  2.953 0.00519 ** 
Infant.Mortality 1.07705  0.38172  2.822 0.00734 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.165 on 41 degrees of freedom
Multiple R-squared:  0.7067, Adjusted R-squared:  0.671 
F-statistic: 19.76 on 5 and 41 DF,  p-value: 5.594e-10
```

### R múltiplo

Esse dado serve para medir a relação linear entre as variáveis dependente e independentes, ou seja, mede o quanto elas estão correlacionadas. O valor desejável é próximo de 1 (100%) [MARIA, 2018].

### R-Quadrado

O R-quadrado é uma medida estatística de quanto os dados estão próximos em linha de regressão. Ele também é conhecido como coeficiente de determinação múltipla para a regressão múltipla. Este valor vai de 0-1, onde o 0 (zero) indica que o modelo não serve para explicar a variabilidade dos dados, enquanto 1 (um) indica que o modelo consegue explicar toda variabilidade dos dados de resposta ao redor da média [MARIA, 2018].

### Coeficientes

São os valores que serão multiplicados pelas variáveis independentes para obter o valor esperado da variável dependente. Valores próximos de 1 (um) indicam que a variável independente analisada interfere fortemente na variável de saída [MARIA, 2018].

### *P-value*

Servem para avaliar a hipótese nula. Em valores inferiores a 0,05 é rejeitado a hipótese nula e a variável não deve ser descartada ou ignorada para o estudo. No caso da figura 2, a variável “Education” apresenta *p-value* muito pequeno e menor que 0,05, logo, esta variável influencia fortemente a variável de saída [MARIA, 2018].

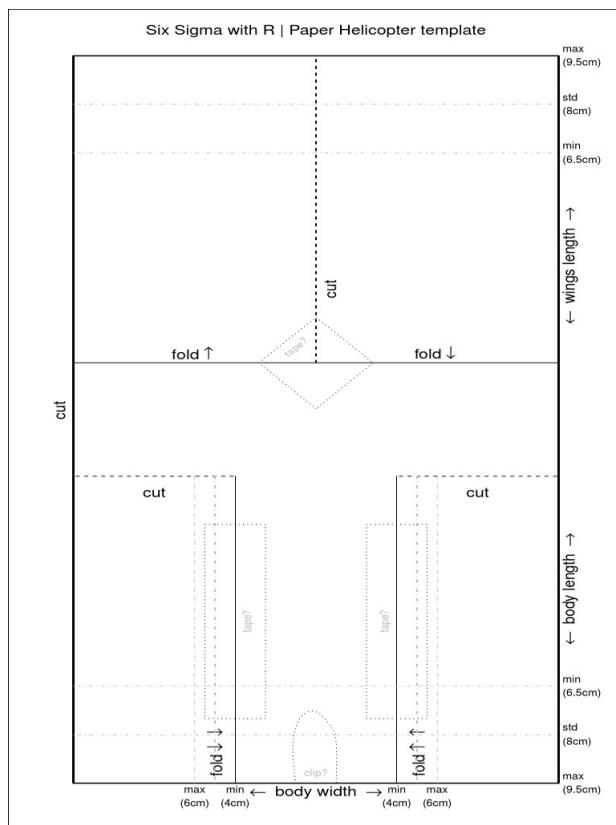
### 3 Metodologia

A metodologia será explanada de acordo com as etapas definidas na subseção 2.1.2.

#### Caracterização do problema e determinação do modelo

Deseja-se avaliar a interação entre algumas variáveis de interesse e o tempo de queda de um “helicóptero” de papel. Para isso foi utilizado o modelo apresentado a na Figura 3, onde se pode ver os cortes e dobraduras a serem realizados. Pode-se notar que há diferentes dimensões que podem ser utilizadas no recorte, bem como a posição onde se pode colocar fita adesiva e um clipe de papel. As dimensões adotadas nesse DoE foram: 9,5 cm para comprimento do corpo (*body length*) e comprimento das asas (*wings length*) e 4 cm para a largura do corpo (*body width*).

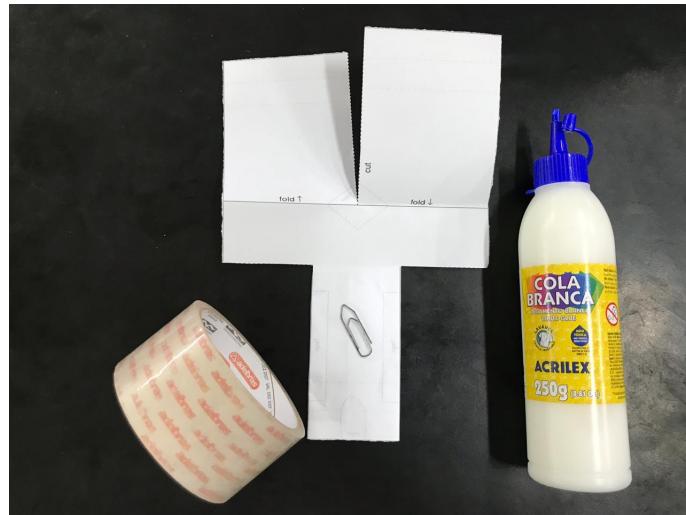
Figura 3: Referência para criação dos modelos.



Foram utilizadas folhas de papel A4 de gramatura  $75 \text{ g/m}^2$  e clipe de papel nº 2/0, possuindo aproximadamente 1g. No modelo há três regiões destinadas a receber fita adesiva e para tanto foi utilizada fita adesiva transparente conforme a Figura 4. Além

disso, a dobradura do corpo foi colada com uma camada fina de cola branca, de modo que a dobradura não abrisse durante o experimento.

Figura 4: Referência para criação dos modelos.



### Fatores de influência e variáveis de resposta

Foram confeccionados 4 modelos básicos de “helicópteros”, que sofreram alterações na massa por meio da adição de um clipe de papel na parte inferior, resultando em 8 modelos. Essas características podem ser resumidamente vistas na Tabela 1, onde foram selecionadas 4 variáveis independentes para o processo, cada uma dividida em dois níveis, e como variável dependente foi selecionado o tempo como parâmetro quantificador dos experimentos.

Tabela 1: Características dos modelos.

Variáveis independentes	Níveis		Variável dependente
Clipe	Com	Sem	
Adesivo lateral	Esquerda	Direita	Tempo de queda
Adesivo topo	Com	Sem	
Altura	1,30 m	2,10 m	

### Determinação de planejamento do experimento

A sequência de planejamento foi gerada aleatoriamente utilizando-se *R*. Isso gerou um total de 16 experimentos, cada um deles realizado duas vezes, ou seja, duas medidas de

tempo foram tomadas por duas pessoas diferentes utilizando cronômetros distintos. A partir disso, um tempo médio de queda foi estabelecido. A Tabela 2 contempla o planejamento gerado em R. O valores “S” e “C” significam “sem” e “com”, respectivamente; similarmente, “E” e “D” significam “esquerda” e “direita”.

Tabela 2: Ordenação do planejamento

Teste n°	Clipe	Adesivo topo	Adesivo lateral	Altura (m)
1	S	S	E	1.30
2	C	S	E	1.30
3	S	C	E	1.30
4	C	C	E	1.30
5	S	S	D	1.30
6	C	S	D	1.30
7	S	C	D	1.30
8	C	C	D	1.30
9	S	S	E	2.10
10	C	S	E	2.10
11	S	C	E	2.10
12	C	C	E	2.10
13	S	S	D	2.10
14	C	S	D	2.10
15	S	C	D	2.10
16	C	C	D	2.10

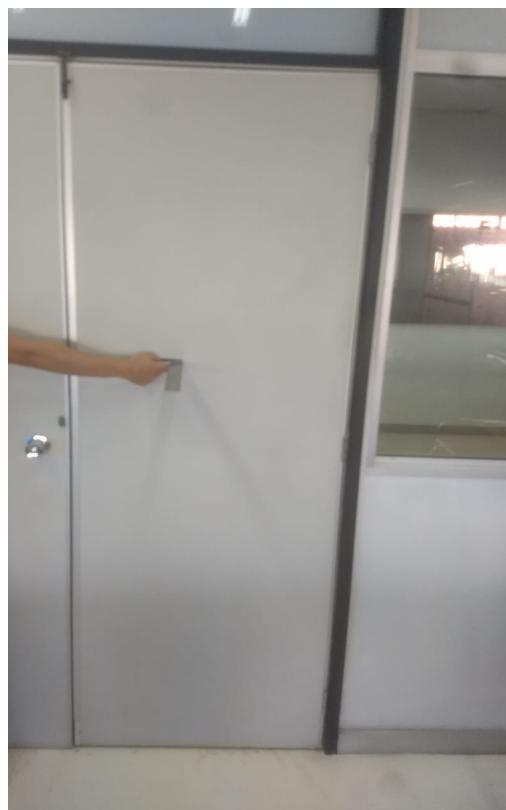
### Condução do experimento

O experimento foi realizado pelo mesma pessoa, no mesmo ambiente, sob mesmas condições climáticas, onde o helicóptero foi abandonado das alturas especificadas tendo sua base colocada nessas alturas, o que pode ser visto na Figura 5.

### Análise dos dados e conclusões

Uma análise de variância realizada utilizando *R* gerou os dados que serão analisados na seção a seguir. Ela irá revelar o peso de cada variável dependente no tempo de queda, mostrando se é possível adotar um bom modelo linearizado para o fenômeno.

Figura 5: Ambiente de realização do experimento.



(a) Altura de 1,30 m.



(b) Altura de 2,10 m.

## 4 Resultados

A partir dos testes realizados sob orientação do planejamento, anteriormente mencionado na seção X, foram coletados os resultados do tempo de vôo do helicóptero. As amostras dos testes coletados, como pode ser visto na tabela 3, possui os seguintes parâmetros: **clip** que pode ser com ou sem; **AT** que representa o adesivo no topo do helicóptero, sendo com ou sem; **ADLat** é o adesivo lateral, podendo ser colocado o adesivo do lado direito ou do lado esquerdo; **AL** é a altura de partida do helicóptero, em que foi definido como 1,30 m e 2,10 m e o tempo de queda em segundos. Dessa forma, com base nestes valores de tempo, a partir das disposições dos demais parâmetros, foi feito o teste de regressão para analisar a influência da relação entre os parâmetros com o tempo.

Tabela 3: Resultados das amostras coletadas.

	clip	AT	ADLat	AL	tempo
1	S	S	E	1.30	0.92
2	C	S	E	1.30	0.88
3	S	C	E	1.30	1.04
4	C	C	E	1.30	1.10
5	S	S	D	1.30	1.10
6	C	S	D	1.30	0.99
7	S	C	D	1.30	1.07
8	C	C	D	1.30	0.92
9	S	S	E	2.10	1.76
10	C	S	E	2.10	1.23
11	S	C	E	2.10	1.88
12	C	C	E	2.10	1.52
13	S	S	D	2.10	1.70
14	C	S	D	2.10	1.72
15	S	C	D	2.10	1.46
16	C	C	D	2.10	1.42

A análise de regressão deste sistema foi dividida em 2 modelos: primeira e segunda ordem. Como o sistema possui 4 variáveis independentes, o modelo máximo atingido pode ser representado pela relação entre os 4 parâmetros. Porém, a partir do terceiro modelo não há mais significância entre os resultados obtidos, que será mostrado em diante, descartando assim a análise com um número maior de relações.

## 4.1 Análise de Primeira Ordem - Modelo 1

A partir da análise linear aplicada aos resultados das amostras, aplicando-se a primeira ordem, o resultado é mostrado na tabela 4. O primeiro modelo dessa análise pode ser visualizado na Equação 2. Esta equação mostra um modelo que consegue explicar o valor do tempo a partir da relação entre os parâmetros. Os valores dentro dos parâmetros são as variáveis que mais apresentaram valores significantes ao sistema, dentre as opções estabelecidas.

Tabela 4: Resultado da análise linear de primeira ordem do modelo 1.

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.0622	0.0910	11.67	1.54e-07
clipC	-0.1419	0.0814	-1.74	0.1092
ATC	0.0119	0.0814	0.15	0.8866
ADLatD	0.0081	0.0814	0.10	0.9223
AL2.10	0.5844	0.0814	7.18	1.80e-05

$$\begin{aligned} tempo = & 1.062187 - 0.141875 \cdot clip(C) + 0.011875 \cdot AT(C) \\ & + 0.008125 \cdot ADLat(D) + 0.584375 \cdot AL2,10 \end{aligned} \quad (2)$$

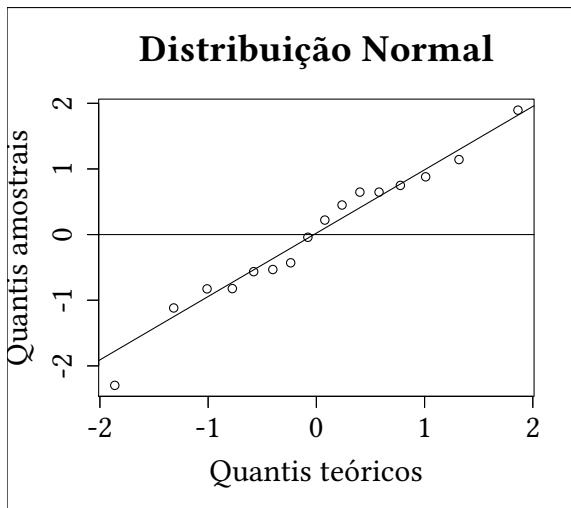
Com base na Tabela 4, pode ser visto que estatisticamente os estimadores dos parâmetros clipe, AT e ADLat são iguais a zero, pois, ao se assumir o nível de significância a 5% de probabilidade, o  $\rho$  valor desses coeficientes é superior a 0,05, aceitando a hipótese nula, ou seja, as estimativas desses parâmetros são números muitos próximos de zero no qual pelo teste t afirma-se que esses valores aceitam a hipótese  $H_0$ . Portanto, se os parâmetros clipe, AT e ADLat são estatisticamente iguais a zero, eles não precisam estar no modelo pois não vai ter nenhuma contribuição significante no valor de tempo de queda do helicóptero. Então, o novo modelo pode ser representado na equação 3, refazendo-se a análise linear do tempo de queda com relação apenas ao parâmetro AL. O resultado dos coeficientes pode ser visto na Tabela 4.

Tabela 5: Análise linear da relação do tempo com AL.

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.0012	0.0577	17.35	7.29e-11
AL2.10	0.5844	0.0816	7.16	4.84e-06

$$tempo = 1.00125 + 0.58438 \cdot AL2,10 \quad (3)$$

Figura 6: Distribuição normal dos resíduos.



O modelo apresentado na equação 2, que representa uma regressão múltipla, foi reduzido para uma regressão simples, representado pela equação 3. Desse modo, para explicar a equação reduzida pode-se dizer que para cada acréscimo a partir da altura de 2,10 m, o tempo de queda aumenta em 0,58 s. Com base no valor do  $R^2$ , que foi de 78,56%, pode se afirmar a porcentagem do dados que são explicados pelo modelo da equação 3. O valor do  $R^2$  ajustado será desconsiderado pois esse último modelo foi o único em que todos os estimadores dos parâmetros foram significativos, no caso dessa análise linear do modelo 1.

Analisando os resíduos do modelo 1 simplificado, foi aplicado o teste de normalidade de *Shapiro-Wilk*. Nesse teste pode-se afirmar que, a partir da verificação dos resíduos desse modelo mostrou-se que no teste da normalidade apresentou um valor de 0,97528 com o  $\rho$  valor respectivo de 0,9152, e adotando o nível de significância de 5%, não houve violação da normalidade dos resíduos. Então, pode-se concluir que esses erros (resíduos) tem uma distribuição normal, como pode ser visto nos gráficos das Figuras 6 e 7. Dessa forma, assume-se a independência dos resíduos pelo fato que a estrutura de coleta declara essa independência. Portanto, não é preciso atribuir testes de independência.

Com base nisso, foi feito o modelo do tempo mínimo e máximo a partir dos coeficientes dos resíduos, como mostrado na figura 8. O gráfico mostra que o valor mínimo do modelo atingido dessa relação é de 1,00125 s e o valor máximo atingido é de 1,5856 s. Portanto pode-se confirmar a relação positiva da altura com o aumento do tempo de queda do helicóptero.

Figura 7: Histograma dos resíduos.

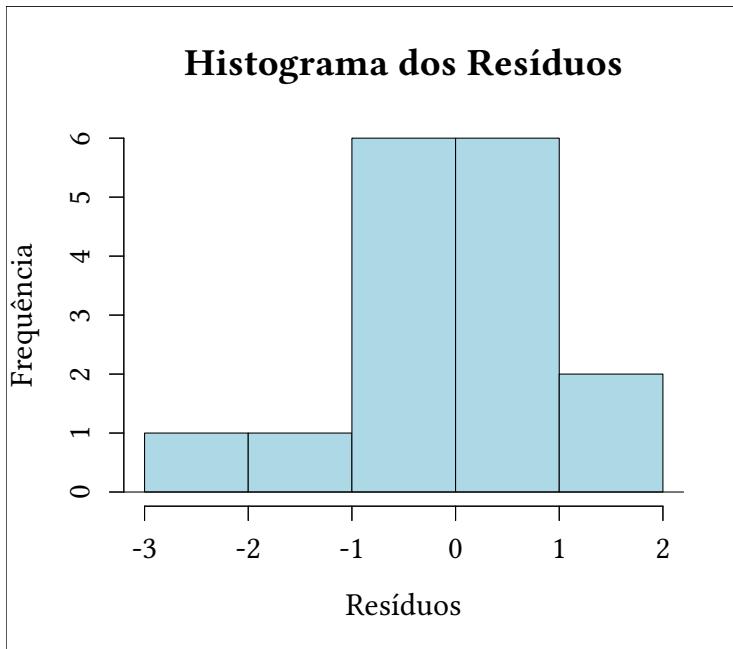
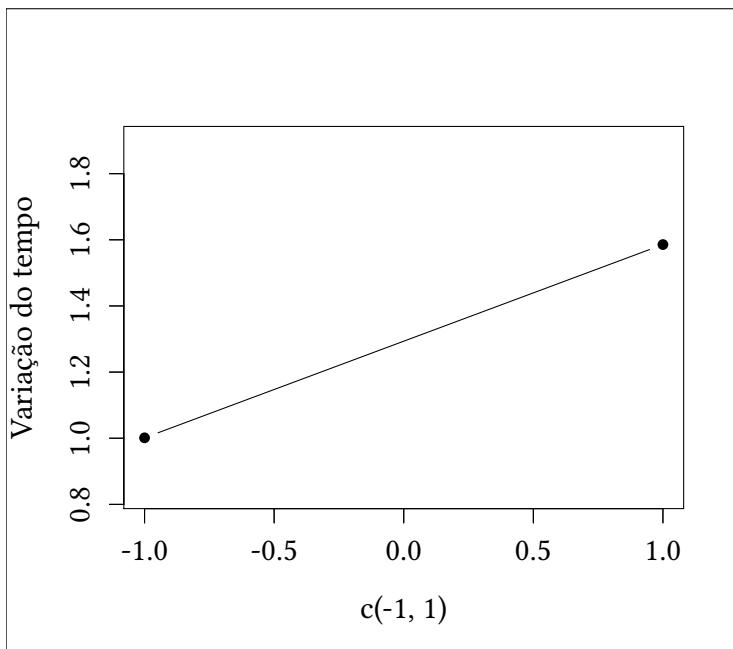


Figura 8: Relação da interação dos resíduos com o tempo.



## 4.2 Análise de Segunda Ordem - Modelo 2

No modelo 2 foi feito a análise linear de segunda ordem. O resultado dessa análise pode ser visto na Tabela 6. A partir dos valores da Tabela 6 foi calculado o modelo 2, representado pela Equação 4.

Tabela 6: Resultado da análise linear de segunda ordem.

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.9203	0.0206	44.73	0.0142
clipC	-0.0506	0.0281	-1.80	0.3227
ATC	0.1094	0.0281	3.89	0.1602
ADLatD	0.1744	0.0281	6.20	0.1018
AL2,10	0.8344	0.0281	29.68	0.0214
clipC:ATC	0.1262	0.0368	3.43	0.1806
clipC:ADLatD	-0.0438	0.0368	-1.19	0.4453
clipC:AL2,10	-0.4638	0.0368	-12.60	0.0504
ATC:ADLatD	-0.1288	0.0368	-3.50	0.1773
ATC:AL2,10	0.0162	0.0368	0.44	0.7353
ADLatD:AL2,10	-0.2238	0.0368	-6.08	0.1038
clipC:ATC:ADLatD	-0.1925	0.0425	-4.53	0.1383
clipC:ATC:AL2,10	0.0225	0.0425	0.53	0.6900
clipC:ADLatD:AL2,10	0.5675	0.0425	13.35	0.0476
ATC:ADLatD:AL2,10	-0.2475	0.0425	-5.82	0.1083

$$\begin{aligned}
 tempo = & 0.93906 - 0.15 \cdot clipC + 0.21375 \cdot ATC + 0.1425 \cdot ADLatD \\
 & + 0.74875 \cdot AL2,10 + 0.04125 \cdot clipC \cdot ATC + 0.14375 \cdot clipC \cdot ADLatD \\
 & - 0.16875 \cdot clipC \cdot AL2,10 - 0.34875 \cdot ATC \cdot ADLatD - 0.09625 \cdot ATC \cdot AL2,10 \\
 & - 0.06375 \cdot ADLatD \cdot AL2,10
 \end{aligned} \tag{4}$$

Essa equação do modelo 2 consegue explicar o valor do tempo a partir da relação de segunda ordem dos parâmetros. Como pode ser visto na Tabela 6, o parâmetro que possui os estimadores mais significantes, se assumir o nível de significância de 5%, é novamente o AL, ou seja, estatisticamente os valores do  $\rho$  valor são maiores que 0.05 em todos exceto o AL2,10. Como isso considera a hipótese nula em que as estimativas desses coeficientes possuem valores muito próximo de zero, podendo também ser retirado da equação da relação de segunda ordem, pois, estatisticamente esses parâmetros não apresentam contribuição significativa no valor do tempo de queda do helicóptero. Desta maneira, simplificando a Equação 4 do modelo 2 irá ficar da mesma forma que a Equação 3, propiciando o encerramento da análise de regressão para o modelo 2.

## 5 Conclusão

O planejamento foi realizado respeitando os princípios de replicação e aletoriedade, bem como as etapas de execução de um estudo estatístico baseado em **DoE**.

O estudo estatístico realizado na Seção 4 revela a altura como sendo a variável mais relevante na saída do sistema, e o clipe constitui a variável mais expressiva na alteração do momento de inércia do “helicóptero”. Desse modo, a relação entre os outros parâmetros apresentados não houve nenhuma significância com a coleta das amostras dos testes realizados e o modelo fica adequadamente representado pela Equação 3.

O valor de 78,56% obtido no parâmetro  $R^2$  indica que o modelo consegue explicar por meio de uma relação linear a relação entre as variáveis dependentes e a variável independente de modo satisfatório.

## Referências

- [Action, sd] Action, P. (s.d.). Análise de regressão.  
<http://www.portalaction.com.br/analise-de-regressao>.
- [COLEMAN, 1993] COLEMAN, D. E.; MONTGOMERY, D. C. (1993). *A systematic approach to planning for a designed industrial experiment*. Technometrics.
- [CORRÊA, 2007] CORRÊA, H. L. (2007). *ADMINISTRAÇÃO DE PRODUÇÃO E OPERAÇÕES*. Editora Atlas.
- [COUTINHO, 2019] COUTINHO, T. (2019). Conheça a metodologia seis sigma e como ela otimiza processos produtivos.  
<https://www.voitto.com.br/blog/artigo/metodologia-seis-sigma>.
- [MARIA, 2018] MARIA, J. (2018). 15 tipos de regressão mais frequentes.  
<https://rpubs.com/JulhinhaM/395633>.
- [MONTGOMERY, 2012] MONTGOMERY, Donald C.; RUNGER, G. (2012). *Estatística Aplicada e Probabilidade Para Engenheiros*. LTC; 5<sup>a</sup> ed.
- [TAHARA, 2008] TAHARA, S. (2008). Melhores práticas - planejamento de experimentos (doe).  
<http://www.portaldeconhecimentos.org.br/index.php/por/Conteudo/Planejamento-de-Experimentos-DOE>.
- [TOLEDO, 2003] TOLEDO, T. P. A. (2003). Tuiuiu papercóptero: Um exemplo didático para a metodologia seis sigma na otimização de duas respostas simultâneas utilizando planejamento de experimentos.  
[http://www.abepro.org.br/biblioteca/ENEGET2003\\_TR1102\\_1545.pdf](http://www.abepro.org.br/biblioteca/ENEGET2003_TR1102_1545.pdf).

---

**RAJA: MANIPULADOR ROBÓTICO DE 5 DOF  
COM DETEÇÃO VISUAL INTEGRADA**

---

## RAJA: MANIPULADOR ROBÓTICO DE 5 DOF COM DETECÇÃO VISUAL INTEGRADA

**Anderson Queiroz do Vale<sup>1</sup>**; Rodrigo Formiga Farias<sup>2</sup>; Aziel Martins de Freitas Júnior<sup>3</sup>;

Jean Paulo Silva<sup>4</sup>; Rebeca Tourinho Lima<sup>5</sup>

<sup>1</sup> Bolsista; Programa Novos Talentos; anderson.vale@fbter.org.br

<sup>2</sup> Bolsista; Programa Novos Talentos; rodrigo.farias@fbter.org.br

<sup>3</sup> Bolsista; Programa Novos Talentos; aziel.freitas@fbter.org.br

<sup>4</sup> Bolsista; Programa Novos Talentos; jean.silva@fbter.org.br

<sup>5</sup> Mestre; Centro Universitário SENAI CIMATEC; Salvador-BA; rebeca.lima@fieb.org.br

### RESUMO

O presente trabalho consiste no desenvolvimento de um manipulador antropomórfico com cinco graus de liberdade com capacidade de detecção visual. O desenvolvimento ocorreu em meio simulado mas fisicamente concluído no Laboratório Robótica e Sistemas Autônomos do SENAI CIMATEC. Foi realizada modelagem estrutural e matemática e simulação. Foram utilizados motores de precisão, perfis de alumínio e uma câmera monocular de alta definição. O uso do framework ROS em sua distribuição Melodic deu suporte para gerenciamento dos algoritmos envolvidos juntamente com suas ferramentas: RViz para visualização gráfica; Gazebo para simulação física e MoveIt, uma API (Application Programming Interface) para planejamento e movimentação de manipuladores robóticos, cálculo de trajetórias e análise de colisões. Após a montagem e testes no protótipo, espera-se obter um protótipo de manipulador robusto, programado de maneira modularizada, validade para que se possa estender suas funcionalidades e capacidade de atuação em tarefas de forma automatizada e, posteriormente, autônoma.

**PALAVRAS-CHAVE:** Manipulador robótico; ArUco; MoveIt; Gazebo; ROS

### 1. INTRODUÇÃO

O estudo da robótica é um ramo da tecnologia que engloba área de Mecânica, Eletrônica e Computação, com graus de teoria de controle, microeletrônica, inteligência artificial, fatores humanos e de produção.<sup>1</sup> Nos últimos anos houve um forte interesse em áreas de robótica devido à necessidade de inserir uma maior padronização, aumento da produtividade e segurança no processo fabril. Na área industrial, a robótica evoluiu devido ao aumento de uso de robôs e manipuladores industriais.

O manipulador robótico consiste em um dispositivo mecânico composto de elementos rígidos (elos), que proporcionam uma sustentação e um alcance ao mesmo. Os elos são conectados entre si através de articulações (juntas), que oferecem graus de liberdade ao manipulador e controle do movimento relativo entre os elos. Essas juntas podem ser basicamente divididas em dois grupos: juntas prismáticas e juntas de rotação. Neste projeto foi utilizado as juntas de rotação.

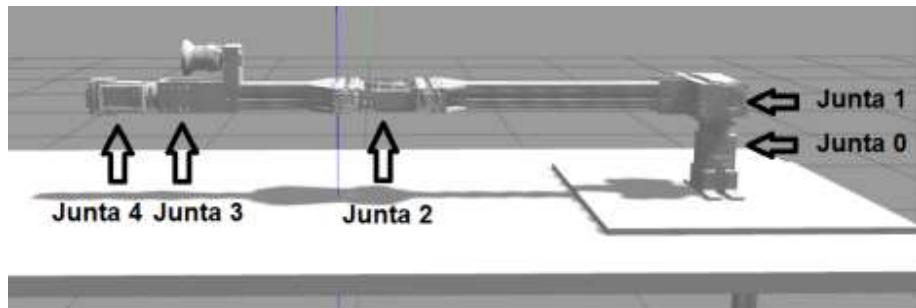
Existe atualmente muitas aplicações para os manipuladores robóticos, sendo elas de soldagem, pintura, carregamento, entre outras. Neste trabalho a aplicação do manipulador consiste em pressionar um botão a partir de uma leitura do ArUco, que é um marcador quadrado que possui um identificador, utilizando uma câmera para localizar o ArUco e poder estimar a coordenada do botão.

### 2. METODOLOGIA

O projeto desta aplicação é baseado no desenvolvimento de um manipulador robótico, utilizando o framework do ROS, e um ambiente com uma mesa, uma caixa com um botão e o ArUco. Para as articulações desse manipulador foi utilizado atuadores inteligentes Dynamixel do fabricante ROBOTIS: nas juntas 0 e 1, como mostra a figura 1, está o modelo PH54-200-S500-R; nas juntas 2 e 4 está o modelo MX-106R e na junta 3 está o modelo PH42-020-S300-R. Para fazer a conexão entre as juntas, também denominados de *Link*, foram utilizados dois perfis de alumínio estrutural com dimensões de 40x40-10 mm. O primeiro link possui um tamanho de 56 cm e o segundo 46 cm. A ferramenta de trabalho do manipulador (*end-effector*) é uma chapa de metal localizado na ponta da junta 4. Ela será responsável em realizar o pressionamento do botão. Próximo a junta 3 está localizado a câmera Teledyne Genie Nano C2590, que é responsável pela parte visual do manipulador.

No total, essa estrutura, possui um tamanho máximo de 1,16 m na vertical e um alcance máximo de 1 m na horizontal. Cada junta tem um limite de rotação de 180 °. Esses limites foi imposto para que não ocorra nenhuma quebra do equipamento ou rompimento dos cabos quando for trabalhado em um ambiente real.

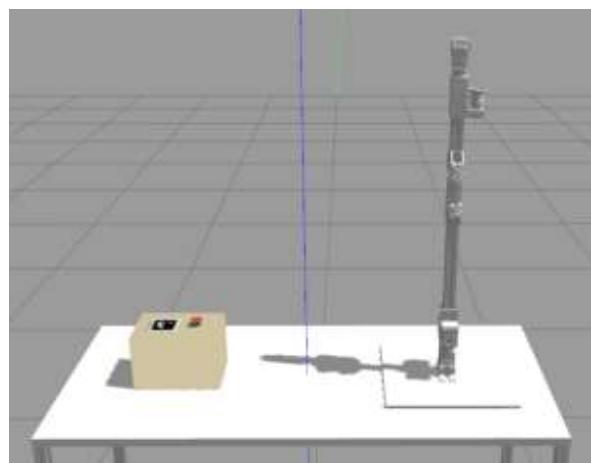
Figura 1 – Manipulador robótico.



Fonte: Os autores.

A metodologia foi dividida em duas etapas, uma etapa de simulação para analisarmos o funcionamento e o comportamento previamente, e uma etapa de construção real do manipulador. Na etapa de simulação o manipulador foi desenhado em software de CAD (*Onshape*) e toda a implementação de algoritmos de aquisição de dados e controle foi realizada via pacotes *ROS* (*Robot Operating System*). O simulador utilizado foi o *Gazebo* (versão 9) para virtualizar o manipulador em um espaço de trabalho controlado e o seu alvo, que é em uma batoeira de emergência localizado numa caixa que possui um ArUco. Esses materiais estão dispostos em cima de uma mesa, como mostra a figura 2.

Figura 2 – Ambiente de simulação.



Fonte: Os autores.

Para a segunda etapa, que é a realizar os testes em um ambiente real, os resultados obtidos em simulação auxiliarão na montagem, além da realização de testes desempenho e medições. O ambiente que será utilizado para os testes reais é localizado no laboratório de robótica e sistemas autônomos do SENAI CIMATEC.

### 3. RESULTADOS E DISCUSSÃO

A concepção do manipulador para simulação foi gerada a partir da modelagem e dimensionamento de elos, juntas e uniões. Após a modelagem estrutural e matemática do manipulador, todos os elos e juntas foram desenhados e criado um URDF (*Universal Robotic Description Format*) para iniciar a simulação no *Gazebo*. Foi realizado um levantamento da área de trabalho do manipulador tendo em vista que alguns pontos, mesmo próximos, não podem ser alcançados pelo efetuador devido as colisões da estrutura e limitações de ângulo de junta.

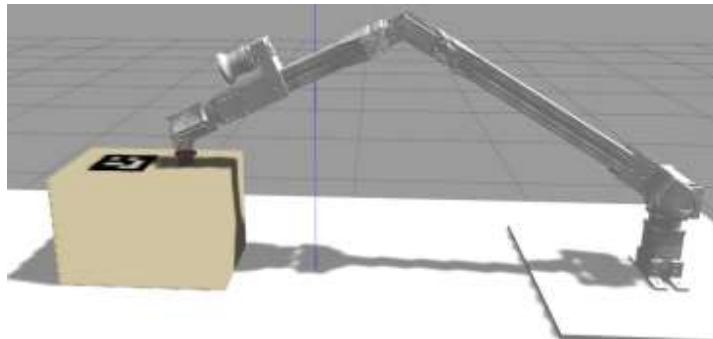
Como parte da fundamentação teórica pode-se dizer que os manipuladores são robôs programáveis que apresentam características antropomórficas, equipados com sensores e programas para possibilitar realizar o objetivo com o máximo de precisão. Possuem uma vasta aplicação na industrial, esses manipuladores se diferenciam de máquinas de automação fixa, por não serem tão limitados a determinadas atividades.<sup>3</sup> Para

realização desse projeto, devem executadas algumas etapas, entre elas, o estudo da cinemática do manipulador robótico, calibração de câmera e configuração dos motores segundo documentação do fabricante. A cinemática, consiste no estudo referente a velocidade, posição, aceleração e todas as variáveis do movimento, baseadas na geometria do manipulador. A cinemática é comumente dividida em direta e inversa. Esse estudo possibilita calcular um conjunto de soluções de possíveis ângulos de junta que poderiam ser usados para se obter determinada posição e orientação do efetuador dentro da área de trabalho do braço.<sup>4</sup>

ROS é um *framework* flexível que contém uma vasta coleção de ferramentas e bibliotecas que visam integrar e simplificar a criação de um robô.<sup>5</sup> Para utilizar muitas das funcionalidades de manipulação no ROS, utiliza-se o produto de um projeto conhecido como *MoveIt*. Já o RViz é uma ferramenta de visualização gráfica nativa do ROS e se destina a oferecer detalhamento na forma de recursos visuais para parâmetros diversos da simulação gerados em tópicos e mensagens do ROS que, tipicamente, são grandezas numéricas. O software *MoveIt* tem a finalidade de padronizar e facilitar a realização de planejamento e execução de trajetórias com manipuladores industriais. Dessa forma, a partir de um modelo baseado em elos, juntas e atuadores, é possível definir limites de operação, configurar controladores, estabelecer e detectar colisões, ter acesso a algoritmos de cinemática inversa e de movimentação, utilizar e tratar dados de percepção, tudo de maneira sistemática e integrada.<sup>6</sup>

Como resultado final para a operação do manipulador robótico, a posição em que ele conclui o sistema pode ser visto na figura 3. O *end-effector* se posiciona em cima do botão, simulando assim o pressionamento.

Figura 3 – Ambiente de simulação.



Fonte: Os autores.

#### 4. CONSIDERAÇÕES FINAIS

Os softwares utilizados, bem como os métodos de resolução da cinemática inversa deram suporte para o desenvolvimento deste projeto. O *framework* ROS deu celeridade ao processo de modelagem e desenvolvimento de algoritmos. O estudo dos esforços presentes no braço, permitiram concluir que os motores utilizados, atenderão quaisquer condições de esforços dentro da área de trabalho do manipulador.

#### 5. REFERÊNCIAS

- <sup>1</sup> PIMENTA, Thiago Tavares. **Controle de Manipuladores Robóticos**. PUC-Rio, Rio de Janeiro, 2009.
- <sup>2</sup> MATARIC, Maja J. **Robótica Industrial**. Unesp, 2014.
- <sup>3</sup> ROSÁRIO, João. **Robótica Industrial I: Modelagem, Utilização e Programação**. São Paulo: ed. Baraúna, 2010.
- <sup>4</sup>CRAIG, John; tradução Heloisa Coimbra de Souza. **Robótica**. São Paulo: Pearson Education do Brasil, 2012.
- <sup>5</sup> OPEN SOURCE ROBOTICS FOUNDATION. ROS Wiki. Acesso em 27 Mar. 2020.
- Disponível em <https://www.ros.org>
- <sup>6</sup> MOVEIT.ROS. Acesso em 28 Mar. 2020. Disponível em <https://www.moveit.ros.org>

---

**Certificado de participação no V SAPCT**

---



**INTELIGÊNCIA  
ARTIFICIAL**

## CERTIFICADO

O Centro Universitário SENAI CIMATEC confere o presente certificado a  
**Anderson Queiroz do Vale**

pelo prêmio de melhor trabalho da **Categoria de Bolsista de Projeto PD&I** do Evento V Seminário de Avaliação de Pesquisa Científica e Tecnológica de 2020, pela publicação e apresentação do trabalho: RAJA: MANIPULADOR ROBÓTICO DE 5 DOF COM DETECÇÃO VISUAL INTEGRADA dos autores Anderson Queiroz do Vale, Aziel Martins de Freitas Júnior, Jean Paulo Silva, Rodrigo Formiga Farias, Rebeca Tourino Lima.

Salvador, 05 de junho de 2020.

Jailson Bittencourt de Andrade

*Pró-reitor de pós-graduação e pesquisa do Centro Universitário SENAI CIMATEC*



PELO FUTURO DA INOVAÇÃO

---

**PERSPECTIVES ON AUTONOMOUS  
UNMANNED GROUND VEHICLES: A SURVEY**

---

## PERSPECTIVES ON AUTONOMOUS UNMANNED GROUND VEHICLES: A SURVEY

Miguel F. N. Vieira<sup>a</sup>, Leonardo M. S. Lima<sup>a</sup>, Israel C. M. Neto<sup>a</sup>, Vinicius J. G. A. Felismino<sup>a</sup>, Rodrigo F. Farias<sup>a</sup>, Jean P. Silva<sup>a</sup>, Anderson Q. Vale<sup>a</sup>, Marco A. Reis<sup>a</sup>, Rebeca T. Lima<sup>a</sup>

<sup>a</sup> Robotics and Autonomous Systems Center, SENAI CIMATEC, Brazil.

**Abstract:** Currently, mobile robots have been used in different scenarios and applications, even for household chores. Using these devices can increase safety and productivity, especially in industrial operations. This study aims to identify gap technologies through an updated selection and comparison of wheeled unmanned ground vehicles (UGVs). Based on a systematic review and classification of commercially available robots, the results indicated established technologies while opportunities for research, development and innovation on mobile robotics can be found in user interface, connectivity and intervention features.

**Keywords:** autonomous robots; mobile robotics; unmanned ground vehicles; state of the art.

## PERSPECTIVAS EM VEÍCULOS AUTÔNOMOS TERRESTRES NÃO TRIPULADOS: UM SURVEY

**Resumo:** Atualmente, os robôs móveis têm sido usados em diferentes ambientes e aplicações, até mesmo para tarefas domésticas. O uso desses dispositivos pode aumentar a segurança e a produtividade, especialmente em operações industriais. Este estudo visa a identificar gaps tecnológicos por meio de uma seleção atualizada e por uma comparação de veículos terrestres não tripulados com rodas (UGVs). Com base em uma revisão sistemática e em uma classificação de robôs comercialmente disponíveis, os resultados indicaram as tecnologias já estabelecidas, enquanto as oportunidades para pesquisa, desenvolvimento e inovação em robótica móvel podem ser encontradas em recursos de interface para usuário, conectividade e intervenção.

**Palavras-chave:** robôs autônomos; robótica móvel; veículos terrestres não tripulados, estado da arte.

## Apêndice H

### 1. INTRODUCTION

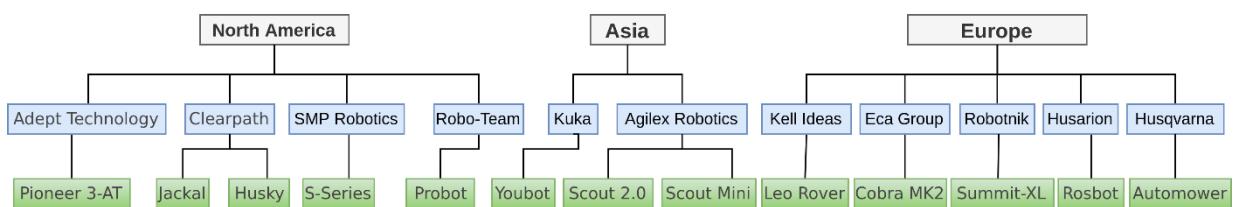
Mobile robots are electromechanical platforms equipped with a locomotion system, capable of navigating through certain environments, endowed with different levels of autonomy that may carry loads [1]. Currently, mobile robotics field is constantly evolving and there are commercially available autonomous vehicles to solve worldwide issues such as unknown locations exploration, industrial inspections or rescue missions. This paper intends to show the State of The Art (SoTA) for wheeled type Unmanned Ground Vehicles (UGVs). UGVs can be used for many applications specially the dangerous ones or considered impossible to human being to do so. These robots are useful to agriculture [2], nuclear [3] and military [4] industries. These vehicles are usually equipped with sensors and some intelligence to provide autonomous navigation or can be teleoperated by an operator in a safe distance.

This paper is organized as follows: Section 2 describes applied methodology to gather available robots' information, mostly commercial off-the-shelf and organize them according to size categories; Section 3 discusses the results on proposed criteria to classify such robots. Finally, Section 4 presents conclusion on UGVs state of the art.

### 2. METHODOLOGY

This study was led in three stages: search for up-to-date information about available wheeled UGVs on companies' websites, academic papers and technology news; gather and organize technical specification about identified robots; establish and classify the robots according to proposed criteria. Then this paper presents the state of the art on wheeled UGVs and offer directions on robotics research and development opportunities. Preliminary data processing of collected information was dealt through a mind map categorizing the robots by location and company (Figure 1).

Figure 1 – Mind map of current UGVs



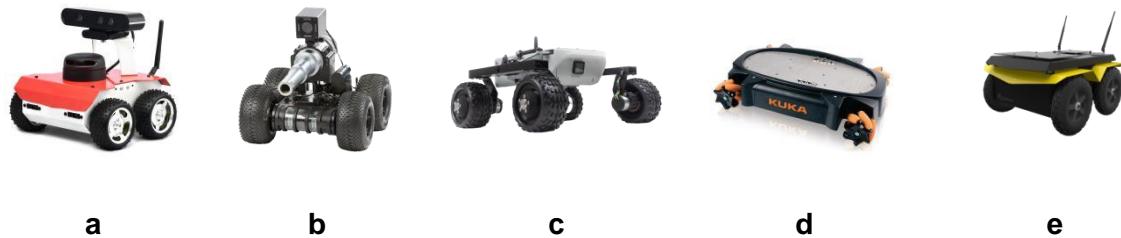
Those wheeled robots were gathered from sources around the world and it was noticed that most relevant UGVs models are available in North America, Asia and Europe. This is possibly a consequence of robotics research centers presence in these locations and great economic power of countries on these continents. These models were grouped below according to their footprint size in three categories: Small size robots, medium size robots and large size robots.

## Apêndice H

### 2.1. Small Size Robots

This category selects up to 0.25 m<sup>2</sup> footprint size robots (Figure 2). Most of these models have been developed for research and education applications, such as *ROSbot*, an open-source mobile platform developed by Husarion. This mobile platform is focused on Robotic Operating System (ROS) education and can be used as a base for a variety of robotic applications like inspection and custom services [5]. Small robots can also be used in military applications as they can reach places that are difficult for human access, for example *Cobra MK2 E*. This UGV developed for defense and security applications by the ECA Group weighs only 5 kg. It has GPS module, camera, temperature sensor, artificial intelligence and an intuitive graphical user interface [6]. *Leo Rover* is a mobile robot developed by Kell Ideas in 2019 for company and education applications. It can also be outdoors applied due to waterproof casing that offers resistance to extreme environments. By being open-source and having its architecture based on ROS, it becomes accessible for developers [7]. Although it is no longer available, many researches are still being carried out with *youBot*. This platform developed in 2010 by KUKA for research and education applications has extended movement capability due to its omnidirectional wheel system and can have one or two manipulators integrated with [8]. *Jackal* is a fast mobile robot developed by Clearpath Robotics in 2014 for field research and education applications. Due to its robust aluminum chassis and weatherproof housing, it is ideal for outdoor operation. It has localization and vision sensors fully integrated with ROS. In addition, an extensive list of advanced sensors can be added to it because of its mounting plates [9].

Figure 2. Small robots: a) ROSbot; b) Cobra MK2 E; c) Leo Rover; d) youBot; e) Jackal



### 2.2. Medium Size Robots

The robots classified in this category are from 0.25 m<sup>2</sup> to 0.60 m<sup>2</sup> footprint size. It has been noticed a greater variety of applications for these models shown in Figure 3 than in previous category. The *Pioneer 3-AT*, for example, is a research mobile robot developed by Adept Technology in 2011. Due to its reinforced tires and lacquered aluminum body, it is ideal for all-terrain exploration missions. This robot run under ROS and can be improved by Pioneer's software development kit (SDK). In addition, there are several supported and tested accessories that could be integrated with it [10]. Focused on research and education applications, *Scout Mini* is a lightweight and flexible robot developed by AgileX Robotics that can ride both indoor and outdoor environments. It can support peripherals like cameras, IMU and LiDAR [11]. Mobile robots can also help with household chores: *Automower 535 AWD* is a robotic lawn

## Apêndice H

mower developed by Husqvarna that can work outdoor, up to 30 degrees' inclinations. It has GPS, ultrasonic and height sensors [12]. *Summit-XL* is a versatile and resistant robot developed by Robotnik for surveillance and military purposes, it has high modularity and mobility and can operate indoor and outdoor due to its two kinematic configuration which incorporate rubber or mecanum wheels. This robot has ROS based architecture and it is equipped with laser scanner, GPS, and camera. It can carry heavy loads and navigate in rough environments due to its strong mechanical structure [13].

Figure 3. Medium robots: a) Pioneer 3-AT; b) Scout Mini; c) Automower 535; d) Summit-XL.



## 2.3. Large Size Robots

Robots with more than 0,6m<sup>2</sup> footprint size have been grouped in this category. As shown in Figure 4 they are generally more robust and resistant, being commonly used in security and military applications. *Scout 2.0* is a multiuse UGV developed by AgileX Robotics for research and education, indoor and outdoor security patrolling applications. It has a modular design, flexible connectivity and a powerful motor system, capable of high payload. In addition, components such as stereo camera, laser, radar, GPS, IMU and robotic manipulator can be installed on it [14]. *Husky* was the first field robotic platform to provide native support for Robotic Operating System (ROS) providing open-source code examples in ROS community. This mobile robot developed by Clearpath Robotics is suitable for robotics research in real and simulated environments. It supports additional modules such as camera, GPS, LiDAR, IMU and even manipulators [15].

Figure 4. Large robots: a) Scout 2.0; b) Husky; c) S-Series; d) Probot.



*S-Series* robots were developed by SMP robotics for outdoor security and inspection applications. Due to a rigid aluminum coating used for protection, robot can

## Apêndice H

be used in moist and rough environments. Most notable feature of these models is auto charging docking station which allows robots to charge their batteries without any human intervention [16]. Professional Robot (*Probot*) is a military robust and versatile robot developed by Roboteam for indoor and outdoor operations. It can perform semi-autonomous missions and carry heavy payloads while maintaining high speeds. It can also be teleoperated, follow the leader or use GPS waypoint navigation [17].

### 2.4. Classification Criteria

As a way to provide a comparison among presented models, 10 robotics development parameters, shorten as criteria, were defined and ranged into 3 score levels where L1 stands for 1 point, L2 stands for 3 points and L3 stands for 5 points.

- Payload Ratio (PR): Ratio of the payload of the robot to its weight. L1:  $PR < 0.5$ ; L2:  $0.5 < PR < 1.0$ ; L3:  $PR > 1.0$ .
- Speed (S): Moving speed in m/s. L1:  $S < 1.0$ ; L2:  $1.0 < S < 3.0$ ; L3:  $S > 3.0$ .
- Runtime (R): How long full charged battery lasts in hours. L1:  $R < 1.0$ ; L2:  $1.0 < R < 3.0$ ; L3:  $R > 3.0$ .
- User Interface (UI): How user may interact with the robot: L1: Command line (No GUI); L2: Installed program (GUI); L3: Web app (GUI).
- Autonomy Readiness (A): Ability of robot navigate and execute tasks without human intervention. L1: Remotely operated; L2: Autonomy-ready; L3: Full autonomous.
- Terrain Type (T): Ability of the robot to operate in different environments. L1: Flat terrain; L2: Rough terrain; L3: Soft sand.
- Perception Readiness (P): Available sensors on the robot. L1: IMU and range finder; L2: LiDAR or camera; L3: RGBD or Stereo camera.
- Connectivity (C): Robot communication standards. L1: USB and RS232; L2: Ethernet or Wi-Fi; L3: 4G/5G network.
- Ingress Protection (IP): Degree of protection against solid particles and liquid ingress. L1:  $IP < 23$ ; L2:  $23 < IP < 64$ ; L3:  $IP > 64$ .
- Terms of use (TU): Accessibility of robot technology. L1: Proprietary technology; L2: Free and Open Source Software (FOSS); L3: Free and Open Source Hardware (FOSH).

## 3. RESULTS AND DISCUSSION

The classification of selected robots based on proposed criteria is shown in the Table 1. Rate column displays an arithmetic mean as a global robot classification. Criterion score shows each parameter performance as an arithmetic mean too.

Apêndice H

Table 1. UGVs Comparison

	<b>PR</b>	<b>S</b>	<b>R</b>	<b>UI</b>	<b>A</b>	<b>T</b>	<b>P</b>	<b>C</b>	<b>IP</b>	<b>TU</b>	<b>Rate</b>
<b>ROSbot</b>	5	3	5	1	5	1	5	3	5	3	3.6
<b>Cobra MK2 E</b>	3	3	3	3	5	5	5	3	5	1	3.6
<b>Leo Rover</b>	3	1	5	5	3	3	3	3	5	5	3.6
<b>youBot</b>	3	1	3	1	3	1	3	3	1	3	2.2
<b>Jackal</b>	5	3	5	1	3	3	5	3	5	3	3.6
<b>Pioneer 3-AT</b>	3	3	5	3	3	1	5	3	1	3	3.0
<b>Scout Mini</b>	1	5	1	1	3	5	5	1	1	3	2.6
<b>Automower 535</b>	1	1	3	5	5	1	1	1	3	1	2.2
<b>Summit - XL</b>	1	5	5	1	5	5	5	3	3	3	3.6
<b>Scout 2.0</b>	3	3	3	1	3	3	5	1	1	3	2.6
<b>Husky</b>	5	3	5	1	3	3	5	1	3	3	3.2
<b>S-Series</b>	1	3	5	5	5	1	5	5	5	1	3.6
<b>Probot</b>	5	5	5	3	5	5	5	3	5	1	4.2
<b>Criterion score</b>	3,0	3,0	4,1	2,4	3,9	2,8	4,4	2,5	3,3	2,5	-

These results point out to possible development directions and indicate mobile robotics' current state of the art according to the robotics development parameters. It was not expected to have full scoring criterion, since this technology field has only just begun to be explored. However, it can be stated that a plenty of high quality sensors are already available on most robots to enhance their autonomy capabilities. While perception feature is almost consolidated, manipulator-integrated robots for mobile intervention are still uncommon and present a great opportunity for future developments. On Table 1 it can be noticed lower performance scores on user interface, connectivity, terrain type and terms of use criteria. Most of robots only offer command line access, while a few offer web based applications which indicates strong relation with poor score of connectivity criterion. Since robots become less restricted (by physical or local wireless networks) and more connected, multiplatform applications become demanding as well. Otherwise, if 4G/5G networks are not pursued, development of more complex multi-robot systems could face a barrier. The results for the terrain type criterion is expected as suppliers usually limit robot applications and this feature improvement could increase overall product costs. The result of the terms of use criterion is not restricted to the field of robotics, as recently

## Apêndice H

open software increasingly acquires market share, whereas open hardware does it at a slower pace [18].

## 4. CONCLUSION

This paper has presented some of the most relevant UGV models currently available on the market and compared according to ten classification criteria. The results indicated that perception and runtime features are established technologies whereas user interface and connectivity integration features are not so mature. Therefore, they offer great opportunities for research & development on mobile robotics. Complimentary analysis revealed that an intervention feature such as integration of robotic manipulators is a challenging one, since it is not available on most of current solutions and it is a favorable topic for future work and market investments.

## Acknowledgments

The authors would like to thank SENAI CIMATEC for supporting and funding robotics research, development and innovation.

## 5. REFERENCES

<sup>1</sup> SECCHI, H. A. **Una Introducción a los Robots Móviles**. Argentina: UNSJ, 2008.

<sup>2</sup> QUAGLIA, G. et al. **Design of an UGV powered by solar energy for precision agriculture**. Robotics, Multidisciplinary Digital Publishing Institute, v. 9, n. 1, p. 13, 2020.

<sup>3</sup> WANG, M. et al. **Autonomous robot navigation with rich information mapping in nuclear storage environments**. In: IEEE.2018 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR). [S.I.], 2018. p. 1–6.

<sup>4</sup> NARANJO, J. E. et al. **Automation kit for dual-mode military unmanned ground vehicle for surveillance missions**. IEEE Intelligent Transportation Systems Magazine, IEEE, 2018.

<sup>5</sup>HUSARION COMPANY. **ROSBOT 2.0 Manual Description**. Available at: <<https://www.husarion.com/manuals/rosbot-manual/>>. Accessed 21 July 2020.

Apêndice H

---

<sup>6</sup>ECA Group. **Cobra MK2 E Technical Specifications.** Available at: <<https://www.ecagroup.com/en/solutions/ugv>>. Accessed 21 July 2020.

<sup>7</sup> Kell Ideas Company. **LEO Rover.** Available at: <<https://www.leorover.tech/>>. Accessed 21 July 2020.

<sup>8</sup> Génération Robots. **youBot - Technical Specifications.** Available at <<https://www.generationrobots.com/img/Kuka-YouBot-Technical-Specs.pdf>>. Accessed 21 July 2020.

<sup>9</sup> Clearpath Robotics. **Jackal UGV.** Available at: <<https://clearpathrobotics.com/jackal-small-unmanned-ground-vehicle/>>. Accessed 24 July 2020.

<sup>10</sup> Génération Robots. **PIONEER 3-AT.** Available at: <<https://www.generationrobots.com/media/Pioneer3AT-P3AT-RevA-datasheet.pdf>>. Accessed 24 July 2020.

<sup>11</sup> AgileX Robotics Ltd. **SCOUT Mini.** Available at: <<http://www.agilex.ai/index/product/id/3>>. Accessed 24 July 2020.

<sup>12</sup> Systèmes-D. **Husqvarna 535 AWD.** Available at: <<https://www.systemes-d.be/manuals/Manuel535-awd.pdf>>. Accessed 24 July 2020.

<sup>13</sup> ROBOTNIK Ltd. **SUMMIT-XL.** Available at: <<https://robotnik.eu/products/mobile-robots/summit-xl-en/>>. Accessed 24 July 2020.

<sup>14</sup> AgileX Robotics Ltd. **SCOUT 2.0 User Manual.** Available at: <[https://www.generationrobots.com/media/agilex/SCOUT2.0\\_UserManual\\_v2.0\\_EN.pdf](https://www.generationrobots.com/media/agilex/SCOUT2.0_UserManual_v2.0_EN.pdf)>. Accessed 24 July 2020.

<sup>15</sup> **HUSKY.** Available at: <<https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/>>. Accessed on 25 July 2020.

<sup>16</sup> SMP Robotics. **S-Series UGVs.** Available at: <[https://smprobotics.com/products\\_autonomous\\_ugv/](https://smprobotics.com/products_autonomous_ugv/)>. Accessed 25 July 2020.

<sup>17</sup> Roboteam Corporation. **Probot - Professional Robot.** Available at: <<https://roboteam.com/products/probot/>>. Accessed 25 July 2020.

<sup>18</sup> Pearce, J. **Emerging Business Models for Open Source Hardware.** Journal of Open Hardware. 2017.

---

**Certificado de participação no VI SIINTEC**

---

Acesse <https://doity.com.br/validar-certificado> para verificar se este certificado é válido. Código de validação: 9ZU1BB-A

# VI SIINTEC

International Symposium  
on Innovation and Technology

Challenges in science, technology and innovation after COVID-19

Sistema FIEB



## CERTIFICADO

Certificamos que o trabalho PERSPECTIVES ON AUTONOMOUS UNMANNED GROUND VEHICLES: A SURVEY dos autores MIGUEL FELIPE NERY VIEIRA, LEONARDO MENDES DE SOUZA LIMA, ISRAEL CERQUEIRA MOTTA NETO, VINICIUS JOSÉ GOMES DE A. FELISMINO, RODRIGO FORMIGA FARIA, JEAN PAULO SILVA, ANDERSON QUEIROZ DO VALE, MARCO ANTONIO DOS REIS, REBECA TOURINHO LIMA foi aceito para publicação no VI International Symposium on Innovation and Technology (SIINTEC), realizado no período de 21/10/2020 a 23/10/2020.

A handwritten signature in black ink on a white horizontal line.

Profa. Dra. Lilian Lefol Nani Guarieiro  
*Coordenadora SIINTEC 2020*