

UGV SACI: INTEGRADO COM DETECÇÃO VISUAL E MANIPULADOR

Relatório do Projeto UGV

Autores:

Anderson Queiroz do Vale
Jéssica Lima Motta
Mateus Santos Cerqueira

Facilitadores:

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

**Salvador
Bahia, Brasil**

Novembro de 2020

| | |
|---|--------|
| Título: UGV Saci: integrado com detecção visual e manipulador | |
| PROD. TEC. CCRoSA - 001 / 2020 | Versão |
| Classificação: () 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)

| |
|--|
| Anderson Queiroz do Vale Jéssica Lima Motta Mateus Santos Cerqueira |
| 000 Lucas Cruz da Silva Rebeca Tourinho Lima Tiago Pereira de Souza Marco Antonio dos Reis |
| UGV Saci: integrado com detecção visual e manipulador Salvador Bahia, Brasil Novembro de 2020 |
| Keywords: 1. Mobile robots . 2. Manipulator. 3. Computer vision. 000 |

SUMÁRIO EXECUTIVO

O projeto do UGV Saci- Desafio 3.0, também conhecido como **Saci** 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 04 de Novembro de 2020.

O prazo de execução planejado foi de 30 dias.

RESUMO

O Saci integra o veículo autônomo da *Clearpath Robotics Warthog* equipado com sensores (câmeras, LiDAR e GPS) e o manipulador robótico JeRoTIMON, com o propósito de transformá-lo em um robô autônomo. Este robô foi construído com o intuito de que o mesmo tivesse navegação autônoma para realizar investigação em ambiente externo e construir um mapa deste, detectasse a "bomba" escondida nesse ambiente, e realizasse o desarme da bomba através do manipulador. Para a simulação foram utilizados o software *Gazebo* e a ferramenta de visualização *Rviz*, e para o manipulador foi utilizado *MoveIt*. Este robô também possui sua versão real onde foi possível realizar testes e verificar seu desempenho em campo.

ABSTRACT

Saci is an *UGV*, called *Warthog* made by *Clearpath Robotics* equipped with sensors (cameras, LiDAR and GPS) and with a robotic manipulator JeRoTIMON. This robot was built with the intention that it had autonomous navigation to carry out research in an external environment and to build a map of it, to detect the "bomb"hidden in that environment, and to disarm this "bomb"through the manipulator. For the simulation, the software *Gazebo* and the visualization tool *Rviz* were used, and for the manipulator *MoveIt* was used. This robot also has its real version where it was possible to carry out tests and verify its performance in the field.

LISTA DE FIGURAS

| | | |
|------------|---|----|
| Figura 1: | Frames para o manipulador Elbow. | 16 |
| Figura 2: | Simulação do desafio no <i>Gazebo</i> | 17 |
| Figura 3: | Arquitetura geral do sistema. | 20 |
| Figura 4: | Modelo esquemático. | 21 |
| Figura 5: | Plataforma veicular Warthog, Clearpath Robotics. | 22 |
| Figura 6: | NUC Xi7 (frontal), Intel. | 23 |
| Figura 7: | NUC Xi7 (traseira), Intel. | 23 |
| Figura 8: | Lidar VLP-16, Velodyne. | 24 |
| Figura 9: | GPS SMART6-L, NovAtel. | 25 |
| Figura 10: | Câmera Basler. | 26 |
| Figura 11: | IMU SEN-14001, Sparkfun. | 26 |
| Figura 12: | <i>Switch</i> TL-SG105, TP-Link. | 27 |
| Figura 13: | Estrutura analítica do protótipo. | 27 |
| Figura 14: | Diagrama da localização. | 28 |
| Figura 15: | Diagrama representando o mapeamento. | 29 |
| Figura 16: | Diagrama representando o planejamento. | 31 |
| Figura 17: | Diagrama das interações das funcionalidades com a Navegação. | 32 |
| Figura 18: | Representação da bomba. | 33 |
| Figura 19: | Identificação com filtro de cor. | 33 |
| Figura 20: | Identificação da bomba. | 34 |
| Figura 21: | Fluxograma representando a detecção. | 35 |
| Figura 22: | Representação da identificação da bomba. | 36 |
| Figura 23: | Diagrama da manipulação. | 37 |
| Figura 24: | Integração simplificado das funcionalidades. | 37 |
| Figura 25: | Vista frontal e lateral da plataforma móvel e seus componentes. | 39 |
| Figura 26: | Basler ace acA4600. | 40 |
| Figura 27: | Lente Kowa LM8HC. | 40 |
| Figura 28: | Manipulador. | 41 |
| Figura 29: | Análise a partir do Diagrama de Ishikawa. | 46 |
| Figura 30: | Especificações da câmera Basler ace. | 57 |

LISTA DE TABELAS

| | | |
|-----------|--|----|
| Tabela 1: | Parâmetros dos links para o manipulador Elbow. | 16 |
| Tabela 2: | Especificações da câmera Basler ace. | 25 |
| Tabela 3: | Lições aprendidas | 45 |

LISTA DE SÍMBOLOS E ABREVIATURAS

CCRoSA Centro de Competência de Robótica e Sistemas Autônomos

OMPL Open Motion Planning Library

OpenCV Open Source Computer Vision

LiDAR Light Detection And Ranging

URDF Unified Robot Description Format

NUC Next Unit of Computing

ROS *Robot Operating System*

GPS *Global Positioning System*

IMU *Inertial Measurement Unit*

UGV *Unmanned Ground Vehicle*

DoF *Degrees of Freedom*

SLAM *Simultaneous Localization And Mapping*

LIDAR *Light Detection And Ranging*

UART *Universal Asynchronous Receiver/Transmitter*

CPU *Central Process Unit*

SOTA *State Of The Art*

EKF Extended Kalman Filter

RRT Rapidly-exploring Random Tree

SUMÁRIO

| | |
|--|-----------|
| 1 INTRODUÇÃO | 11 |
| 1.1 Objetivos | 11 |
| 1.2 Justificativa | 11 |
| 1.3 Organização do relatório | 12 |
| 2 CONCEITO DO SISTEMA | 13 |
| 2.1 Parâmetros básicos | 13 |
| 2.1.1 Requisitos do cliente | 13 |
| 2.1.2 Requisitos técnicos | 14 |
| 2.1.3 Estudo do estado da arte | 14 |
| 2.1.4 Ambiente de operação | 17 |
| 3 DESENVOLVIMENTO DO SISTEMA | 19 |
| 3.1 Descrição do sistema | 19 |
| 3.1.1 Arquitetura geral | 19 |
| 3.2 Modelo Esquemático | 20 |
| 3.3 Especificação de componentes | 21 |
| 3.3.1 UGV - Warthog | 21 |
| 3.3.2 Computador central (NUC) | 22 |
| 3.3.3 <i>LiDAR</i> | 24 |
| 3.3.4 Sistema de georreferenciamento | 24 |
| 3.3.5 Câmera Basler ace acA4600 | 24 |
| 3.3.6 IMU | 25 |
| 3.3.7 <i>Switch</i> | 26 |
| 3.3.8 Estrutura analítica do protótipo | 27 |
| 3.4 Funcionalidades | 27 |
| 3.4.1 Localização | 28 |
| 3.4.1.1 Dependências | 28 |
| 3.4.1.2 Saídas | 29 |
| 3.4.2 Mapeamento | 29 |

| | |
|---|-----------|
| 3.4.2.1 Dependências | 30 |
| 3.4.2.2 Saídas | 30 |
| 3.4.3 Planejamento | 30 |
| 3.4.3.1 Dependências | 30 |
| 3.4.3.2 Saídas | 31 |
| 3.4.4 Navegação | 31 |
| 3.4.4.1 Dependências | 32 |
| 3.4.4.2 Saídas | 32 |
| 3.4.5 Detecção | 32 |
| 3.4.5.1 Dependências | 33 |
| 3.4.5.2 Saídas | 34 |
| 3.4.5.3 Dependências | 34 |
| 3.4.5.4 Saídas | 34 |
| 3.4.6 Manipulação | 35 |
| 3.4.6.1 Dependências | 35 |
| 3.4.6.2 Saídas | 36 |
| 3.5 Arquitetura de software | 36 |
| 4 IMPLEMENTAÇÃO | 39 |
| 4.1 Estrutura física | 39 |
| 4.1.1 Câmera | 39 |
| 4.1.2 LiDAR | 40 |
| 4.1.3 Warthog | 40 |
| 4.1.4 Manipulador | 41 |
| 5 RESULTADOS E DISCUSSÕES | 43 |
| 6 GESTÃO DO CONHECIMENTO | 45 |
| 6.1 Lições aprendidas | 45 |
| 6.2 Análise através do Diagrama de Ishikawa | 45 |
| 6.3 Guia de uso | 46 |
| 7 CONCLUSÃO | 49 |

| | |
|--|-----------|
| REFERÊNCIAS | 51 |
| ANEXO A Warthog | 53 |
| ANEXO B Câmera Basler ace acA4600 | 57 |
| ANEXO C Lente Kowa LM8HC | 67 |
| ANEXO D Velodyne LiDAR Puck LITE | 69 |

1 INTRODUÇÃO

A robótica é um ramo da tecnologia que engloba área de eletrônica, mecânica e computação, com graus de teoria de controle, microeletrônica, inteligência artificial, fatores humanos e de produção (PIMENTA, 2009). A busca por robôs autônomos equipados com sensores e atuadores, como por exemplo manipuladores robóticos, para realizar atividades que colocam em risco a vida de seres humanos, seja pelo desempenho de uma atividade perigosa ou por o local ser de difícil acesso, vem se tornando o foco de pesquisas na área de robótica.

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

1.1 Objetivos

O objetivo do projeto é desenvolver um *UGV* também conhecido como veículo terrestre não tripulado. E este deve possuir navegação autônoma, capaz de explorar um ambiente externo onde este possui marcadores visuais para limitar a exploração do robô, identificar uma bomba por meio de uma câmera RGB e proceder com a função de desarmá-la a partir de um braço robótico. Para isso, os objetivos específicos são:

- Realizar estudo do Estado da Arte(*SOTA*) sobre *UGVs*, detecção visual de objetos e integração com manipuladores.
- Modelagem para a simulação.
- Parametrizar o pacote de reconhecimento de marcadores visuais.
- Desenvolver um pacote de configuração do manipulador no *MoveIt*.
- Desenvolver o código para realizar a missão.
- Realizar a simulação em software.
- Implementar a versão física do protótipo.
- Realização de ajuste de parâmetros e testes.

1.2 Justificativa

A busca por meios de realizar atividades em locais que a presença do ser humano torna-se difícil, arriscado e até mesmo impossível, com precisão e eficiência, têm se tornado

prioridade. Além da capacidade do robô de interagir com o ambiente a partir da captura e análise de estímulos visuais ([LEITE, 2005](#)).

Há uma demanda crescente, porém falta 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.

Este projeto traz a utilização de veículos autônomos que sejam capazes de identificar marcadores visuais para saber o seu limite de exploração e realizar a tarefa de desarmar uma bomba. Espera-se que este projeto seja continuado e que seus resultados sejam compartilhados na comunidade científica, contribuindo para a construção de outros *UGVs* com características e/ou objetivos semelhantes.

1.3 Organização do relatório

Este documento visa agrupar todos os conteúdos utilizados para o desenvolvimento desse projeto desde a fase conceitual até a implementação e testes. O 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 *UGV*. São expostos seus parâmetros de configuração e sua estrutura.

O capítulo [5](#), Resultados, são apresentados os resultados alcançados nesse projeto.

No capítulo [6](#), Gestão do Conhecimento, é feito um estudo sobre as lições aprendidas, foi realizada uma análise usando o Diagrama de Ishikawa, além de apresentar o guia de uso.

Por fim, o capítulo [7](#) 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 robô autônomo como uma máquina na qual o mecanismo programável, geralmente, em um ou mais eixos com graus de liberdade, hábil para execução de tarefas usando estado físico e/ou sensores sem intervenção humana. Em outras palavras, robô autônomo é um equipamento programável baseado em atuadores e sensores, projetado para realizar uma variedade de atividades, assim como realização de diversos processos industriais (ISO-8373, 2012).

O desenvolvimento deste projeto está situado na concepção de plataformas móveis que podem ser utilizadas na navegação em ambientes abertos, detecção e desarmamento de um tipo específico de bomba. Esta plataforma terá modularidade para diversas outras aplicações, tornando possível adaptá-la para ambientes externos e internos devido a diversidade de sensores inseridos no seu corpo. Além dos sensores, o veículo possui um manipulador robótico para realizar as tarefas de corte do fio ou pegar um objeto.

Neste capítulo serão tratados os requisitos solicitados pelo cliente, os requisitos técnicos do projeto e o estudo do estado da arte sobre [UGV](#).

2.1 Parâmetros básicos

Nesta seção serão detalhados os requisitos solicitados pelo cliente, referentes ao projeto, no qual a tarefa precisa ser realizada em um ambiente externo. Além disso, são exibidos os requisitos técnicos que tratam das especificações do sistema e uma breve revisão teórica de conceitos relacionados ao [UGV](#).

2.1.1 Requisitos do cliente

Requisitos pré-determinados 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. A plataforma móvel usada será o *Warthog*;
2. O *Warthog* sempre será o principal meio para conclusão da solução;
3. O desenvolvimento da solução deve ser implementado no computador externo à plataforma;
4. A equipe usará a câmera RGB (Basler), [LiDAR](#) (Velodyne) e [SLAM](#) (EKF SLAM);

5. O desenvolvimento do **SLAM** deve ter cinco características principais: mapeamento, localização, planejamento de trajetória, destino alvo e desvio de obstáculos;
6. A busca pela bomba e desativação da plataforma móvel devem ser realizados de forma autônoma.

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 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 utilizado para a missão será o do projeto JeRoTimon;
2. O *Warthog* deve se localizar no mundo;
3. A plataforma móvel deve identificar uma região que não é de interesse a partir da detecção de um marco fiducial;
4. Deve ser feito o uso do *framework ROS*;
5. Integrar um minicomputador (NUC) ao sistema para realizar o processamento;
6. O manipulador de desativação da bomba será posicionado em frente ao robô móvel;
7. A plataforma deve identificar a “bomba” a partir de uma câmera RGB e desabilitá-la, aproximando o *end effector* do fio com a coloração específica;

2.1.3 Estudo do estado da arte

Com a necessidade da automatização de tarefas repetitivas e atividades de alto risco, a implementação de robôs autônomos tem crescido exponencialmente. (**ROBOS-AUTONOMOS, s.d.**).

A partir da definição dada por (**MATARIĆ et al., 2007**) sobre um robô móvel autônomo, é possível ter uma noção dos componentes essenciais que o compõe. Os sensores são fundamentais para que o robô seja capaz de perceber o ambiente em que ele se encontra. Da mesma forma, atuadores são fundamentais para que o robô consiga atuar sobre este ambiente. Por fim, para que o robô seja autônomo, é necessário um sistema de controle não supervisionado.

De acordo com (**TILBURY; ULSOY, 2010**), os *UGVs* possuem muitas potenciais aplicações, desde uso em ações militares como por exemplo descarte de material bélico

explosivo, vigilância e reconhecimento em combate, até seu uso industrial e doméstico, como a limpeza de pisos, ou tarefas especiais como operações de resgate.

Para realizar as operações de forma autônoma o *UGV* precisa de métodos de planejamento de caminho e estes possuem duas classificações: método de planejamento global e método de planejamento local (SEDIGHI et al., 2004). Para o método de planejamento global, o *UGV* requer conhecimento sobre o ambiente e assume que o terreno é estático. Em contrapartida, para o planejamento do caminho local, o ambiente é parcialmente conhecido ou até mesmo completamente desconhecido. E para este método o *UGV* utiliza a informação proveniente de sensores que enviam esses sinais em tempo real durante sua navegação local (WANG et al., 2005). Os sensores também auxiliam ao veículo autônomo a se movimentar no ambiente sem colidir com os obstáculos encontrados.

Para que esses robôs possuem navegação autônoma uma das técnicas utilizadas é o uso da visão computacional, onde esta pode ser definida como um conjunto de técnicas e teorias com o objetivo de obtenção de informação por meio de imagens, sendo de modo geral, o estudo da capacidade das máquinas em visualizar o mundo real (tridimensional), por meio de sensores, câmeras e outros dispositivos que extraem informação dos ambientes (ALVES, 2005). A identificação de imagem monocromática pode ser feita a partir de uma função bidimensional, $f(x,y)$, onde x e y são coordenadas espaciais no plano e o número de pixels e sua intensidade em tons de cinza variam de acordo com a coloração imagem formada no plano. (GONZALEZ; WOODS, 2000). Um objeto pode ser encontrado de diversas maneiras, a partir da identificação de características específicas como: cor, geometria, tamanho e composição do material. Para realizar o controle da tarefa de identificação, a "bomba" é identificada a partir de um algoritmo escrito em *Python* e que utiliza bibliotecas do *OpenCV*.

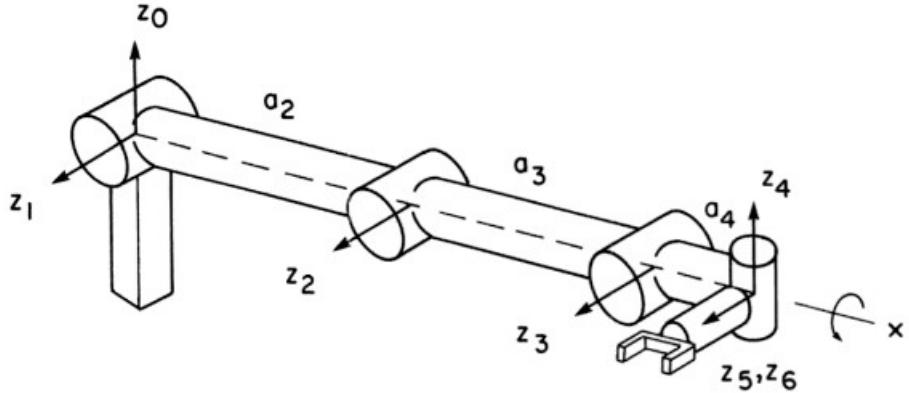
Na parte da atuação para o desarme da "bomba" foi utilizado um manipulador robótico e muitas pesquisas têm sido realizadas nesta área. Em (HERNANDEZ-MENDEZ et al., 2017) é descrito o desenvolvimento de um manipulador robótico com 3 *DoF* e dois dedos independentes. Este robô foi desenvolvido com o propósito 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 *end effector*¹ durante o processo de *pick and place*². 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 estudo da dinâmica e da cinemática apresenta a síntese do projeto de um dispositivo,

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

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

Figura 1: Frames para o manipulador Elbow.



Fonte: ([PAUL, 1981](#)).

expressando matematicamente as relações de movimento de um mecanismo na execução de determinada tarefa.

Para a análise cinemática, utiliza-se os parâmetros de Denavit-Hartenberg (D-H). Este realiza uma cadeia cinemática espacial através da fixação de sistemas de referência aos elos ([PAUL, 1981](#)). Esta notação de Denavit-Hartenberg é uma ferramenta comumente utilizada para sistematizar a descrição cinemática de sistemas mecânicos articulados com n graus de liberdade ([JR; DENAVIT; HARTENBERG, 1964](#)). Na Tabela 1 tem-se a cadeia cinemática através dos parâmetros D-H para o manipulador da Figura 1, nesta tabela tem-se o elo (*link*), θ refere-se ao ângulo de rotação no eixo z e α rotação no eixo x , a refere-se a translação ao longo do eixo de rotação x e d translação ao longo do eixo z . Na cinemática direta é possível observar que a partir dos ângulos das juntas do manipulador encontra-se a posição e orientação da ferramenta com relação a estação de trabalho. Contudo, quando projeta-se um manipulador, é fundamental realizar os cálculos da cinemática inversa. Nesta, por sua vez, são obtidos dados de posição e orientação da ferramenta com relação a estação de trabalho e calcula-se os ângulos das juntas ([CRAIG, 2012](#)).

Tabela 1: Parâmetros dos links para o manipulador Elbow.

| Link | Variável | α | a | d | $\cos \alpha$ | $\sin \alpha$ |
|------|------------|----------|-------|-----|---------------|---------------|
| 1 | θ_1 | 90° | 0 | 0 | 0 | 1 |
| 2 | θ_2 | 0° | a_2 | 0 | 1 | 0 |
| 3 | θ_3 | 0° | a_3 | 0 | 1 | 0 |
| 4 | θ_4 | -90° | a_4 | 0 | 0 | -1 |
| 5 | θ_5 | 90° | 0 | 0 | 0 | 1 |
| 6 | θ_6 | 0° | 0 | 0 | 1 | 0 |

Fonte: ([PAUL, 1981](#)).

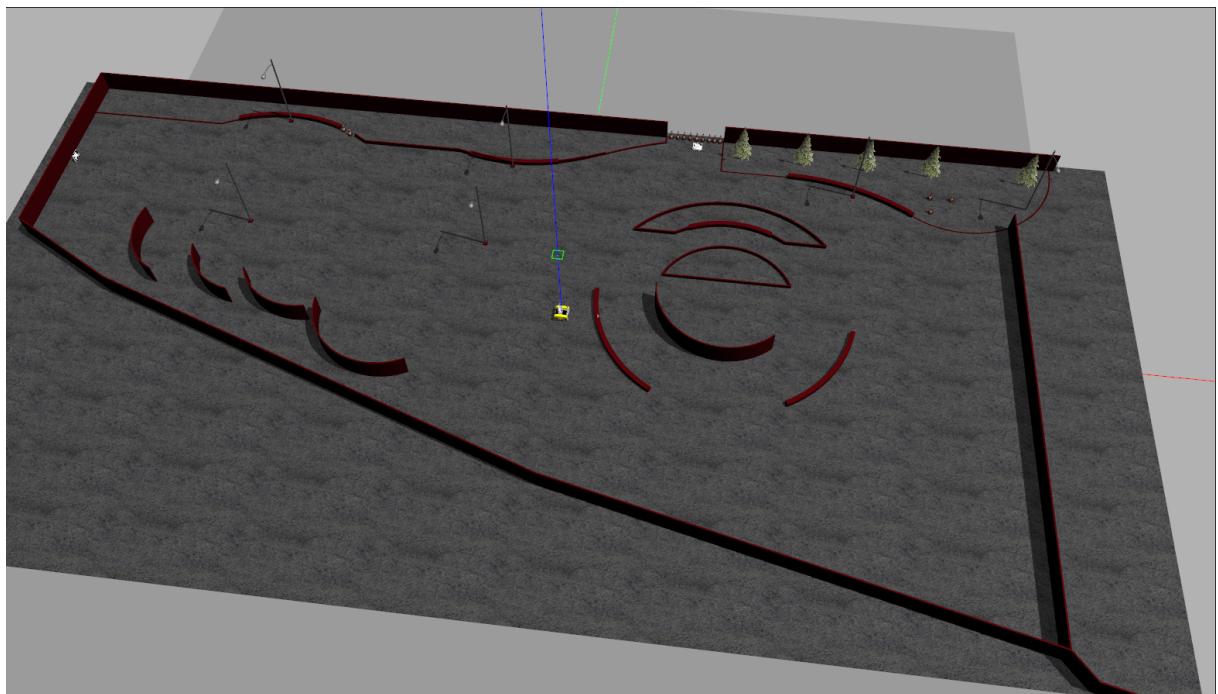
O controle da cinemática e o planejamento da movimentação evitam que o manipulador colida com outros elementos do seu espaço de trabalho. Para tal é necessário aplicar técnicas de controle e utilização de sensores, estes últimos fazem com que o robô tenha capacidade de percepção coletando dados tanto internos, sobre seu sistema mecânico, quanto externos, no meio ambiente em qual está inserido (SCIAVICCO; SICILIANO, 2012).

A dinâmica do manipulador dependerá do seu design mecânico, logo esta definição irá variar de um manipulador para outro, e depende também de já haver definido as formulações cinemáticas (NORTON, 2001). Estas por sua vez, fornecem os elementos geométricos de um sistema que são a base para se obter as energias que compõem as equações dinâmicas.

2.1.4 Ambiente de operação

O ambiente para a realização do desafio, onde serão incluídos o *Warthog*, equipado com os sensores e o manipulador, os *ArUcos* e a "bomba", é no ambiente externo entre os prédios 3 e 4 do SENAI CIMATEC. Na Figura 2 tem-se o ambiente de operação simulado no Gazebo para que possam ser realizados testes de navegação, exploração, detecção visual, reconhecimento de marcadores fiduciais e atuação do manipulador, e realização de ajustes e validação dos parâmetros aplicados.

Figura 2: Simulação do desafio no *Gazebo*



Fonte: Autoria própria

3 DESENVOLVIMENTO DO SISTEMA

Nesta seção serão detalhados o que foi utilizado no desenvolvimento do **UGV**, abordando os sistemas que o compõem em *software* e em *hardware*. Será mostrado a arquitetura geral do sistema, as especificações dos componentes e as principais funcionalidades.

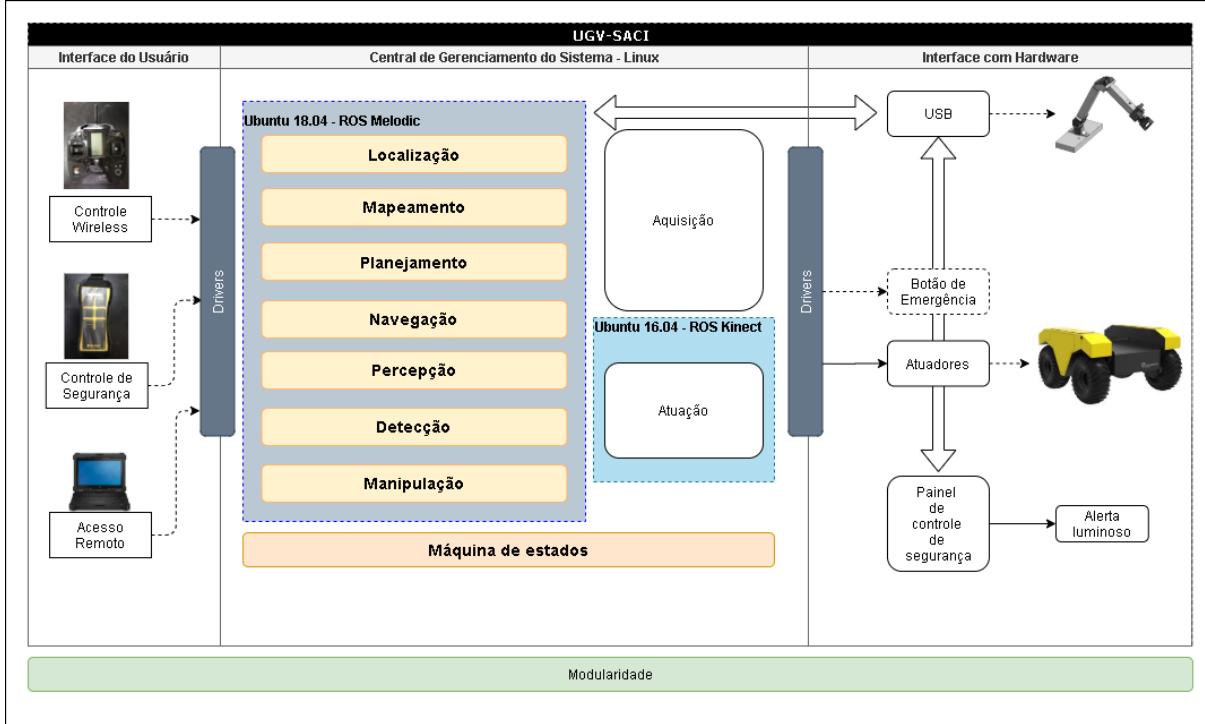
3.1 Descrição do sistema

O **UGV** é um veículo desenvolvido para atender aos desafios relacionados ao desarmamento de bombas. Esses desafios incluem navegação autônoma *outdoor*, detecção e estimativa da posição do objeto e manipular o objeto através de um manipulador robótico. Os pacotes que constituem este robô foram concebidos através do *software* de simulação *Gazebo*, da ferramenta de visualização *Rviz* e do *framework* **ROS**. O uso dessas ferramentas possibilitam que uma grande variedade de atividades, que venham a ser realizadas no mundo real, tenham sido previamente testadas em um ambiente simulado.

3.1.1 Arquitetura geral

A Figura 3 mostra a arquitetura geral do sistema que faz um panorama geral da relação entre interface de usuário, a central de gerenciamento do sistema e o *hardware*. A interface de usuário é responsável pelo contato direto com o usuário, sendo possível intervir na movimentação através do *joystick*, em que você pode movimentar o **UGV**, ou através do controle de segurança, no qual é possível impor um estado de parada de movimento do **UGV**. Por um *notebook* é possível realizar o acesso remoto ao sistema, podendo assim monitorar as tarefas realizadas de forma autônoma pelo **UGV**.

Figura 3: Arquitetura geral do sistema.



Fonte: Autoria própria.

A central de gerenciamento do sistema possui duas variações da distribuição *Linux - Ubuntu*. Para o *Ubuntu 18.04* foi utilizado uma [Next Unit of Computing \(NUC\)](#) com o [ROS Melodic](#). Essa [NUC](#) é responsável pela aquisição de alguns sensores e pelo processamento das funcionalidades presentes na arquitetura (Figura 3.) Para o *Ubuntu 16.04* foi utilizado o próprio sistema de processamento do [UGV](#). Ele é encarregado na atuação dos motores e na aquisição dos sensores do próprio fabricante ([Global Positioning System \(GPS\)](#), Encoder e Resolver). Essa decisão de adicionar uma [NUC](#) ao [UGV](#) foi para reduzir o processamento do próprio sistema e a possibilidade de adicionar funcionalidades que precise de um poder de processamento maior. A máquina de estado é responsável em gerenciar as funcionalidades, passando para atuação do manipulador ou do [UGV](#).

3.2 Modelo Esquemático

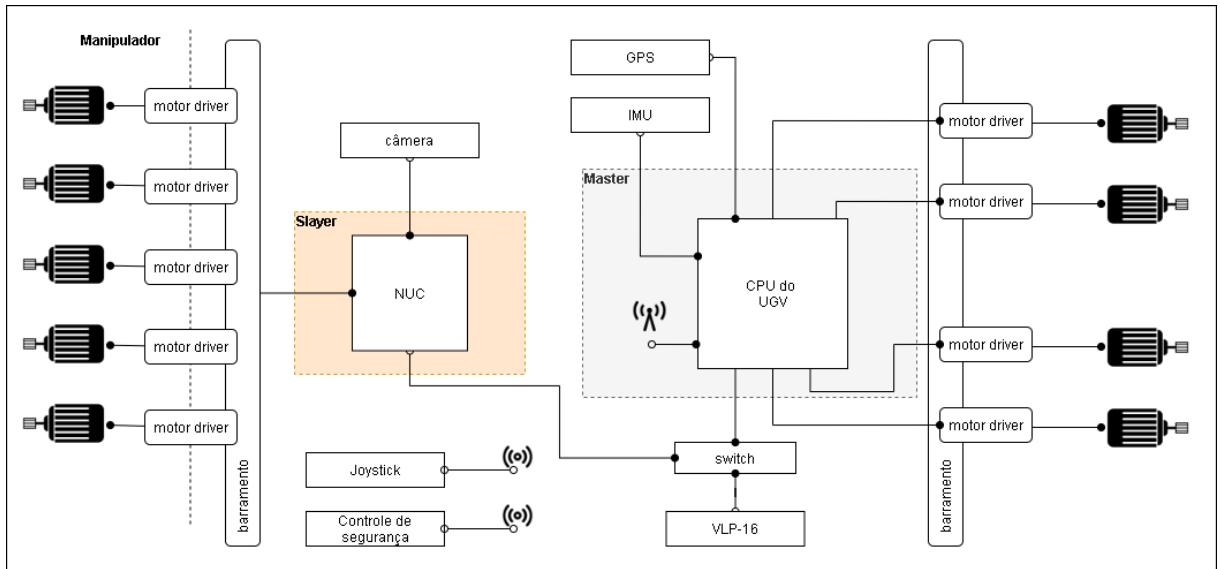
O modelo esquemático, representado na Figura 4, mostra os componentes utilizados no desenvolvimento do projeto, com intuito de representar a forma como estão conectados.

A ligação entre a *NUC* e a *Central Process Unit (CPU)* do [UGV](#) é conectada através do *switch*. A câmera e o manipulador robótico é ligado diretamente na *NUC*. O [Light Detection And Ranging \(LIDAR\)](#), que é representado pelo VLP-16 na Figura 4, é conectado no *switch*, mas seu processamento é feito exclusivamente na *NUC*. O mesmo ocorre para o

GPS e a *Inertial Measurement Unit* (**IMU**), a CPU do **UGV** coleta esses dados e lança-os nos seus respectivos tópicos para serem utilizados posteriormente pela **NUC**.

A unidade de processamento do **UGV** fica aguardando os comandos de movimentação que serão enviados pela **NUC** ou pelo *joystick*. O *joystick* envia informações para um receptor de radiofrequência conectado no computador central. Se o *joystick* estiver ligado e a **NUC** enviar comandos de movimento do **UGV**, irá ocorrer conflito com do *joystick* com o *slayer* e o **UGV** não vai conseguir se movimentar.

Figura 4: Modelo esquemático.



Fonte: Autoria própria.

3.3 Especificação de componentes

A especificação de componentes refere-se a uma relação e descrição dos componentes de hardware e software que constituem o sistema. Os componentes selecionados para compor o sistema são descritos a seguir.

3.3.1 UGV - Warthog

Para que os sensores necessários ao mapeamento sejam integrados e tenham mobilidade sobre uma plataforma robusta, compacta e resistente a condições adversas como impacto, poeira e umidade, foi selecionado um veículo modelo *Warthog* (Figura 5) do fornecedor *Clearpath Robotics*. Esse veículo apresenta interface para comunicação através de uma API para integração ao framework ROS, oferece recursos para controle que possibilitam a teleoperação da plataforma, possui capacidade de carga de 272kg e dimensões adequadas (1.52 m x 1.38 m x 0.83 m) para suportar a instrumentação que equipará o protótipo. Este

modelo possui autonomia elétrica de 2.5 horas, atendendo à necessidade de 2 horas de operação, ângulo de escalada de até 45°, o que possibilita ultrapassar pequenos obstáculos, capacidade de carga de até 272 Kg, além de grau de proteção contra poeira e água IP67 compatíveis com o projeto.

A plataforma inclui freios, suspensão articulada passiva geométrica, saídas de tensão regulada para 5/12/24 V, saída não-regulada 48V, computador embarcado Mini-ITX i5 com 8GB de RAM e 120 SSD, Ubuntu 16.04 e ROS Kinect instalados e configurados para operar o veículo, modelo de simulação 3D para o Gazebo, além de um GPS e uma unidade inercial.

Figura 5: Plataforma veicular Warthog, Clearpath Robotics.



Fonte: Clearpath Robotics.

3.3.2 Computador central (NUC)

Dado o volume de informações necessárias para as atividades de navegação e controle dos diversos equipamentos ligados ao UGV, foi definido um computador central portátil para esse gerenciamento.

Esse computador é o modelo Xi7 da Intel (Figuras 6 e 7). 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 M de memória cache;
- 16 GB de memória RAM DDR3L com frequência de 1333 MHz;
- Placa de vídeo integrada HD Intel 4200 com resolução máxima de 2560x1600@60Hz;
- 4 portas USB (2 portas 2.0 e 2 portas 3.0);
- Saída de vídeo HDMI;
- Saída de áudio e entrada de microfone 3.5 mm;
- Placa de rede que suporta cabeamento RJ45 e Wi-Fi com velocidade de 10/100/1000 Mbps;
- Armazenamento interno via SSD de 256 GB.

Figura 6: NUC Xi7 (frontal), Intel.



Fonte: Intel.

Figura 7: NUC Xi7 (traseira), Intel.



Fonte: Intel.

3.3.3 *LiDAR*

Dispositivo óptico responsável pela detecção de obstáculos no ambiente. O modelo selecionado para esta aplicação foi o VLP-16, da fabricante *Velodyne*, conforme Figura 8. Possui um alcance de até 100 m podendo realizar um *scan* de 360°. Possui resolução angular de 2°, com um campo de visão de ± 15°, 16 canais, tem massa equivalente a 830 g e dimensões de 71 mm de altura e 88,9 mm de diâmetro. Ele trabalha com uma frequência de escaneamento de 5 - 20 Hz e possui classe de laser 1, ou seja, inofensivo aos olhos humanos.

Figura 8: Lidar VLP-16, Velodyne.



Fonte: Velodyne.

3.3.4 Sistema de georreferenciamento

No processo de mapeamento da rota de trabalho é necessário o registro da geolocalização do UGV para que ele possa confirmar a posição correta da atividade de exploração e estimar melhor sua localização. Dito isto, o **GPS** é responsável pelo fornecimento de informações sobre o posicionamento do robô.

O dispositivo utilizado foi o modelo *SMART6-L* da fabricante *NovAtel*, como mostra a Figura 9. É um módulo **GPS** com dimensões de 81 mm de altura e 155 mm de diâmetro. Possui uma frequência de leitura de 20 Hz, precisão de 15 cm e tem um peso de 470 g. Sua conectividade é *Universal Asynchronous Receiver/Transmitter* (**UART**), podendo ser alimentado por uma tensão regulada de 12 V.

3.3.5 Câmera Basler ace acA4600

A ficha resumo contendo as principais especificações da câmera *Basler ace acA4600* está disposta na Tabela 2. A Figura 10 mostra a câmera *Basler*. Essa câmera permite customizar sua imagem a partir de um *software* desenvolvido pelo próprio fabricante, chamado *Pylon*. A partir desse *software* é possível reduzir o tamanho da imagem, definir a



Figura 9: GPS SMART6-L, NovAtel.

escala de cor utilizada e atribuir algumas outras funcionalidades à câmera. Dessa forma, a câmera *Basler* pode ser utilizado em diversas aplicações, devido a sua customização para se adaptar ao tipo de processamento utilizado.

Tabela 2: Especificações da câmera Basler ace.

| 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 μ 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 [$^{\circ}$ C] |
| Massa | < 80 [g] |
| Dimensões (L x A x P) | 29,3 x 29 x 29 [mm ³] |
| Potência necessária | ~ 2,5 W a 5 V _{DC} , máximo de 2,8 W |

Fonte: Basler.

3.3.6 IMU

A **IMU** (Figura 11) é um dispositivo eletrônico capaz de medir e informar valores de aceleração linear, velocidade angular e rotação do robô. O modelo escolhido foi o SEN-14001, da fabricante *Sparkfun*.

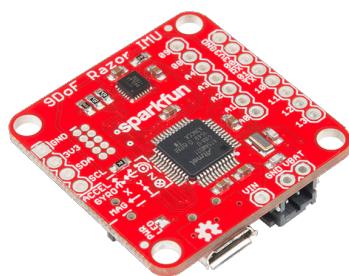
O módulo SEN-14001 combina um microprocessador SAMD21 com um sensor MPU-9250, de 9 graus de liberdade. O MPU-9250 possui três sensores de 3 eixos, um acelerômetro,

Figura 10: Câmera Basler.



Fonte: Basler.

Figura 11: IMU SEN-14001, Sparkfun.



Fonte: Sparkfun.

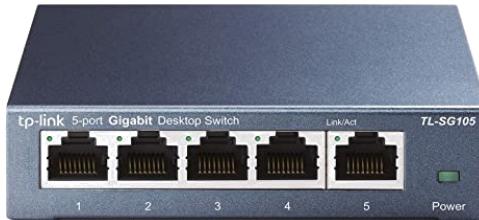
giroscópio e magnetômetro, que permitem detectar aceleração linear, velocidade de rotação angular e vetores de campo magnético.

O microprocessador integrado, SAMD21 da Atmel, é um microcontrolador ARM Cortex-M0+ de 32 bits. O modelo também inclui um soquete para cartão micro SD e chave de controle de energia.

3.3.7 *Switch*

É um dispositivo capaz de distribuir comunicação *ethernet* com os diversos componentes do sistema. O modelo TL-SG105 da TP-Link (Figura 12) foi escolhido por apresentar um número de portas suficiente e velocidade de comunicação necessária para a aplicação. Ele possui 5 portas de conexão com velocidades de 10 Mbps, 100 Mbps e 1000 Mbps. Seus protocolos e padrões operacionais são: IEEE 802.3 / 802.3u / 802.3ab / 802.3x CSMA / CD. Tem 400 gramas e possui dimensões de 99.8 mm x 98 mm x 25 mm, podendo operar em temperaturas de 0 °C até 40 °C e umidade entre 10% a 90% (sem condensação).

Figura 12: *Switch* TL-SG105, TP-Link.

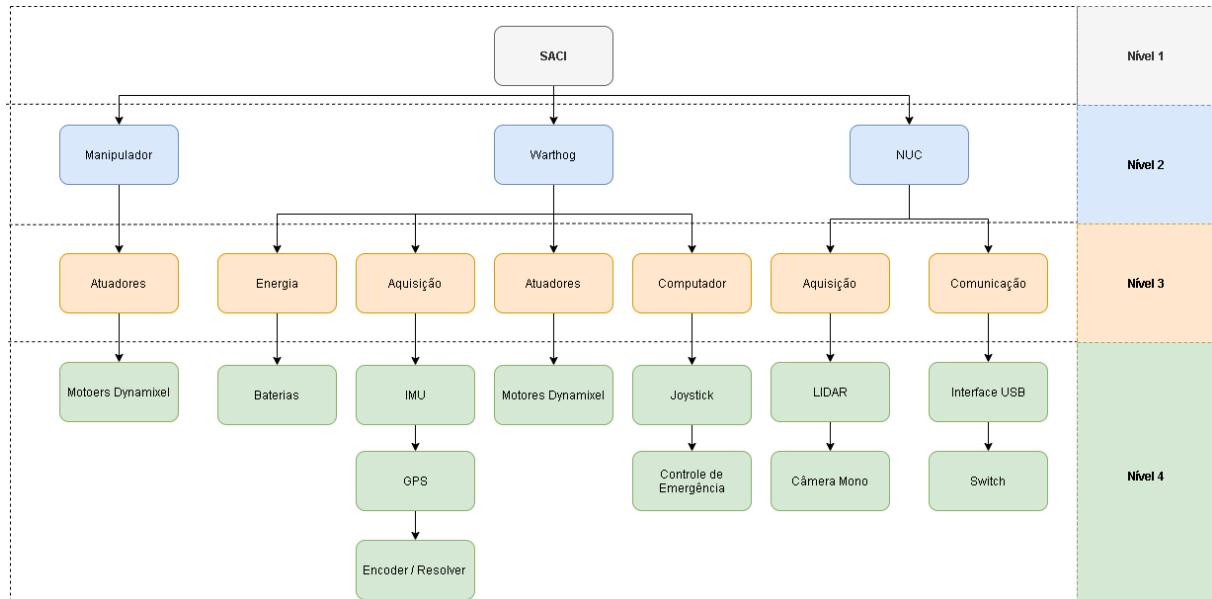


Fonte: TP-Link.

3.3.8 Estrutura analítica do protótipo

A estrutura analítica do protótipo mostrada na Figura 13 exibe as relações sistemáticas entre as partes que compõem o manipulador, o *warthog* e a NUC. A estrutura hierárquica possui quatro níveis: o primeiro, referente a aplicação principal SACI; o segundo nível mostra os principais sistemas da aplicação; o terceiro, mostra os principais subsistemas de cada módulo; o quarto nível mostra os componentes que cada sistema utiliza.

Figura 13: Estrutura analítica do protótipo.



Fonte: Autoria própria.

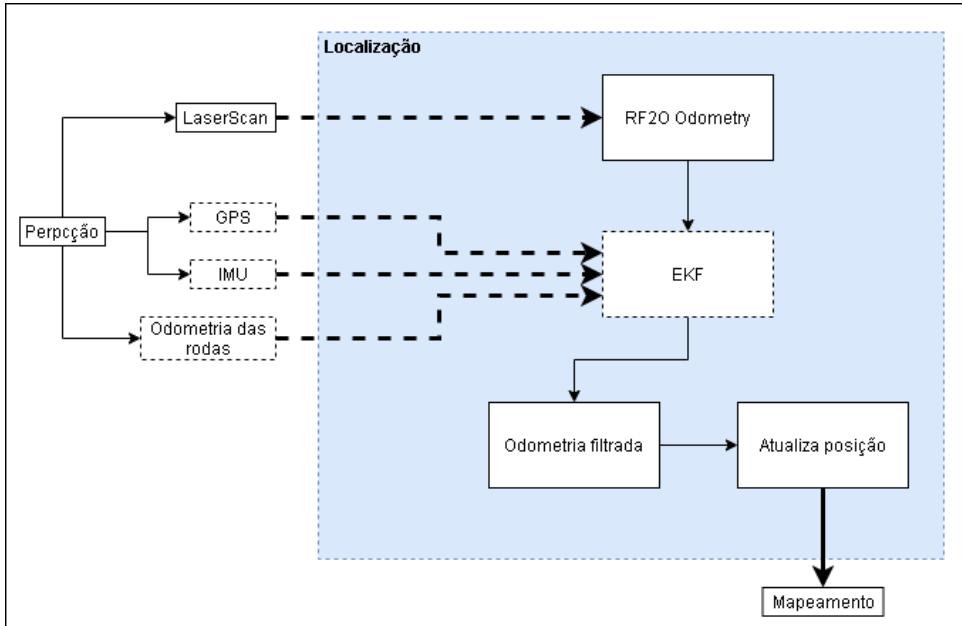
3.4 Funcionalidades

Nesta sessão serão detalhados as funcionalidades desenvolvidas no projeto. As funcionalidades presentes no projeto são: Localização, Mapeamento, Planejamento, Navegação, Percepção, Detecção, Manipulação e Atuação.

3.4.1 Localização

A localização engloba o constante monitoramento da posição e orientação do UGV dentro do ambiente no qual está contido. A localização serve de base para outras funcionalidades que permitem prover autonomia ao robô. A funcionalidade de localização está representada na Figura 14.

Figura 14: Diagrama da localização.



Fonte: Autoria própria.

Para estimar a odometria a partir dos dados da *Point Cloud* do *LIDAR* foi utilizado o pacote *rf2o laser odometry*. Esse pacote estima a odometria 2D com base em varreduras planares a laser. Para isso, foi preciso converter as informações da *Point cloud*, que estão em 3D para 2D, como mostra no diagrama da Figura 14, no qual essa conversão é feita pelo pacote *Point Cloud to Laser scan*.

Com os dados do *LIDAR* devidamente ajustado, o pacote *rf2o laser odometry* estima a odometria e envia-o para o *Extended Kalman Filter (EKF)* fazer a fusão com os sensores da *IMU*, *GPS* e a odometria da roda do UGV, tornando a odometria ainda mais precisa. Nessa parte está com as linhas pontilhadas no diagrama devido a esse método funcionar apenas na simulação, no real foi descartado o EKF por causa do conflito gerado com os pacotes internos do UGV. Assim, os sensores que seriam utilizados como entrada do EKF também não foram utilizados.

3.4.1.1 Dependências

A localização tem como dependência a percepção. A funcionalidade no real depende apenas dos dados enviados pelo LIDAR, já na simulação são utilizados a IMU e a odometria

da roda do UGV. O GPS não é utilizado na simulação devido a não ser possível reproduzir essas informações no ambiente simulado.

3.4.1.2 Saídas

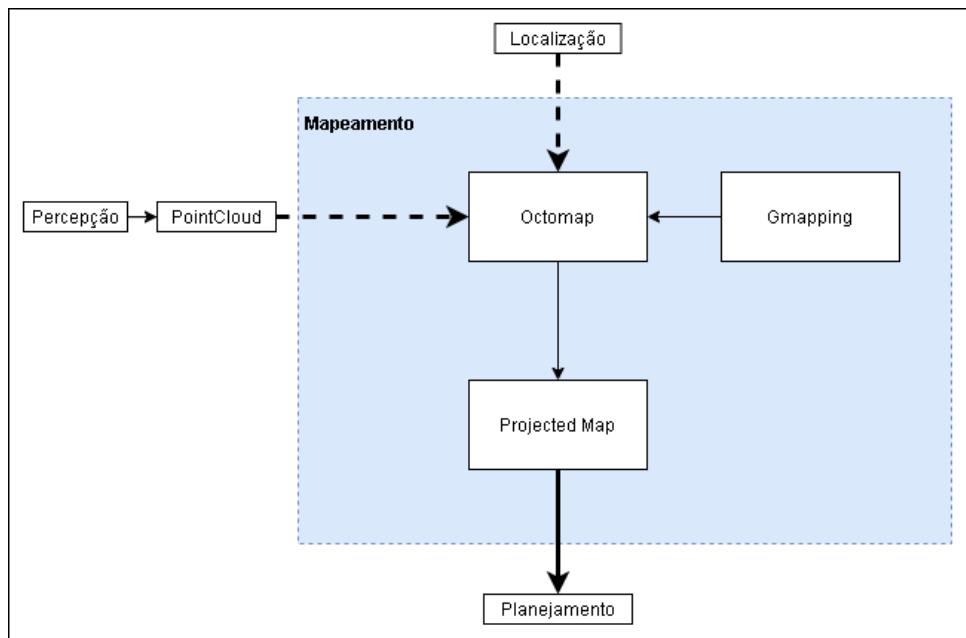
A funcionalidade irá atualizar a posição do UGV, reduzindo o erro gerado pela derivada da função que calcula a odometria. Dessa forma o UGV pode navegar com segurança, pois ele sabe sua real localização e distância entre os obstáculos.

3.4.2 Mapeamento

Dentre as funcionalidades mais utilizadas na robótica móvel, encontra-se a funcionalidade de mapeamento de ambientes. A capacidade de um robô de mapear o ambiente ao seu redor é essencial para que o mesmo consiga realizar a tarefa de movimentação de forma autônoma em ambientes internos ou externos. Para que tal tarefa torne-se possível, é imprescindível que o robô seja equipado com sensores capazes de fornecer boas informações espaciais do ambiente, assim como dotado de um algoritmo capaz de realizar a integração desses dados sensoriais de forma precisa e em todo instante.

A seguir é possível visualizar um diagrama, na Figura 15, representando o processo de mapeamento de um ambiente por parte do robô. A partir de sensores devidamente escolhidos e que sejam compatíveis com o algoritmo de mapeamento utilizado, é possível realizar, em todo instante, o mapeamento da área percorrida pelo robô.

Figura 15: Diagrama representando o mapeamento.



Fonte: Autoria própria.

Como mostra na Figura 15, o pacote responsável em criar o mapa é o *OctoMap*. Ele é

um método que, a partir da sua localização, mapeia uma área utilizando um LIDAR 3D, mas ele converte essa informação de 3D para 2D, construindo uma representação gráfica do mapa em 2D. Assim, com um mapa em mãos, o UGV pode realizar sua navegação de forma segura, realizando todas as trajetórias feitas pelo planejamento.

3.4.2.1 Dependências

O método *OctoMap* tem como dependências a localização, um *frame* de referência do mapa, que é criado pelo *Gmapping*, e os dados do LIDAR em forma de *Point Cloud*. O *Gmapping*, apesar de ser um *SLAM* que fornece localização e mapeamento diretamente, para esta aplicação está servindo apenas para criar o *frame* do mapa para referência do robô. Esse método é essencial porque não é possível criar essa referência diretamente.

3.4.2.2 Saídas

A funcionalidade tem como saída um mapa global representativo do ambiente percorrido pelo robô.

3.4.3 Planejamento

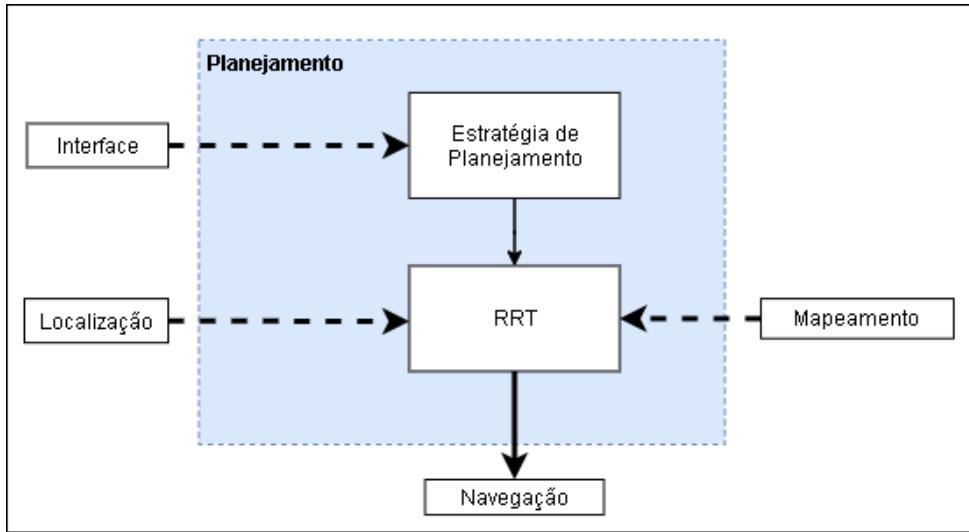
O planejamento de trajetória do UGV deve assegurar que, dadas posições inicial e final ou uma determinada área, seja traçada uma rota de forma que a estratégia de planejamento seja cumprida. Além disto, como o UGV estará realizando suas tarefas em ambientes não necessariamente estáticos, o planejamento deve também levar em consideração dados referentes à detecção de obstáculos caso haja a necessidade de replanejar a rota inicial com o intuito de evitar possíveis colisões.

A Figura 16 exibe o diagrama de funcionamento da funcionalidade. Através da interface, é solicitado ao robô que planeje uma trajetória para cumprir a sua missão, delimitando uma região de exploração. Com base no mapa global, será gerada uma trajetória inicial pelo planejador. Como mostra no diagrama, o *Rapidly-exploring Random Tree (RRT)* é um planejador de trajetória para explorar um ambiente. Ele busca pelos caminhos que ainda não foram explorados dentro de uma região definida pelo usuário.

3.4.3.1 Dependências

O planejamento depende da informação sobre qual será a missão, proveniente da Interface; odometria do robô, proveniente da Localização e o mapa global, provenientes do Mapeamento.

Figura 16: Diagrama representando o planejamento.



Fonte: Autoria própria.

3.4.3.2 Saídas

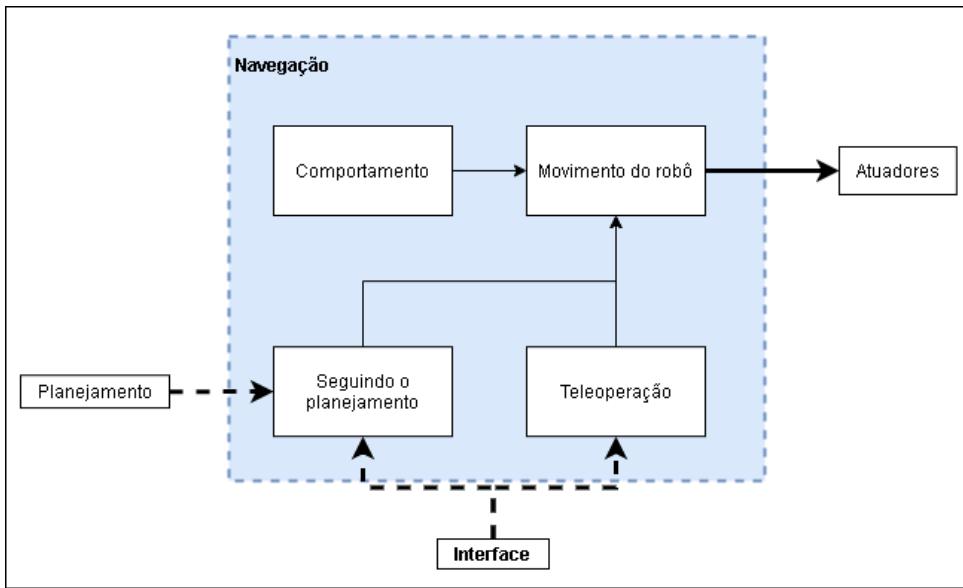
A saída desta funcionalidade será composta por trajetórias que irão para a Navegação do robô.

3.4.4 Navegação

Na navegação de um robô móvel autônomo é utilizado a combinação de três competências fundamentais para transitar de forma segura no ambiente: A localização, o planejamento e a percepção. A capacidade de localização do robô estabelece sua própria posição e orientação dentro do quadro de referência. O planejamento da rota utiliza da localização, extraíndo a posição atual do robô, e a da posição do objetivo, traçando uma rota que deverá ser perseguida pelo robô. A percepção será a capacidade do robô em detectar objetos, um bloqueio ou uma situação de perigo, evitando assim os danos de colisão e realizar uma navegação segura. Portanto, a navegação de robôs móveis autônomos precisa dessas funcionalidades para oferecer os meios no qual o robô autônomo se move de forma segura de um local para outro em qualquer ambiente.

A funcionalidade navegação está representada pelo diagrama de interação entre as funcionalidades da Figura 17. O modo de navegação do robô será escolhido pela interface que poderá ser autônomo (Seguindo o plano) ou manual (Teleoperação). O movimento do robô está destacado como sendo o principal método da navegação. Nele está contido a ligação com os atuadores e a saída da funcionalidade.

Figura 17: Diagrama das interações das funcionalidades com a Navegação.



Fonte: Autoria própria.

3.4.4.1 Dependências

O sistema da navegação possui as seguintes funcionalidades como dependências: Planejamento e Interface. O planejamento deve fornecer os dados que serão a trajetória até o objetivo do UGV. O comportamento será quem irá decidir a forma da navegação, controlando assim a força e a velocidade dos atuadores. A interface irá monitorar a navegação e decidir se será feito de forma autônoma ou teleoperado. O State-machine realizar o controle de todas as funcionalidades.

3.4.4.2 Saídas

Como saída da funcionalidade de navegação será enviado os sinais de movimento dos motores para os atuadores.

3.4.5 Detecção

A detecção de um objeto específico, "bomba", demonstrada na figura 18 foi feito usando a biblioteca [OpenCV](#), uma multiplataforma desenvolvida pela Intel, livre ao uso acadêmico e comercial. A detecção inicial foi feita a partir de um filtro de coloração com o intuito de identificar uma caixa de cor única e específica demonstrada na figura 19. A aplicação foi falha quando a coloração do ambiente externo se igualava com a do objeto.

A aplicação da identificação do objeto com características específicas demonstrada na figura 18 foi feita a partir do uso de detecção de objetos usando classificadores em cascata baseados em recursos Haar, nessa abordagem é realizado um treinamento de função

Figura 18: Representação da bomba.

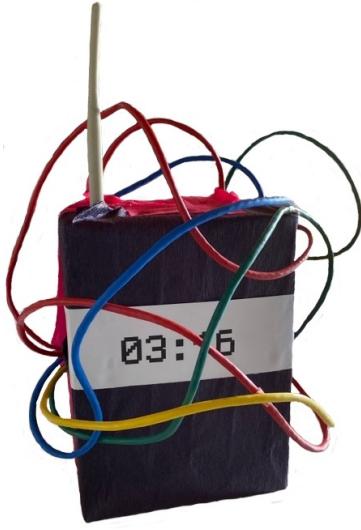
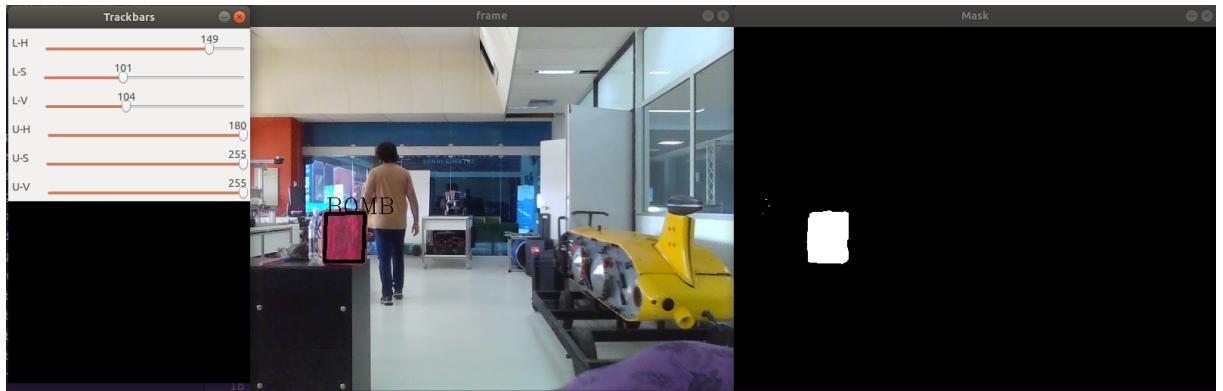


Figura 19: Identificação com filtro de cor.

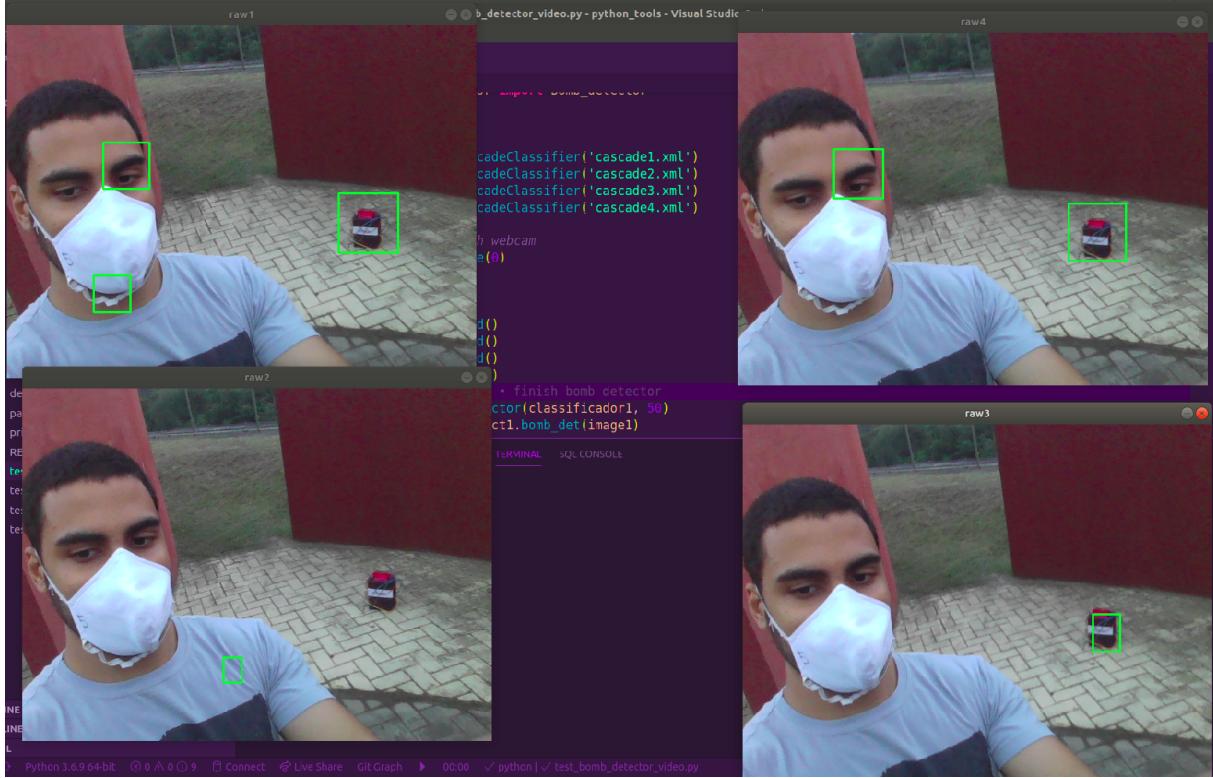


cascata, usando muitas imagens negativas, cujo objeto de interesse não esteja nela, e as imagens positivas, as quais cujo o objeto esteja inserido no ambiente. O treinamento e os métodos para identificação foram feitos usando a biblioteca [OpenCV](#), a identificação da imagens positivas e negativas faz com que algoritmo de treinamento procure características semelhantes na identificação do objeto.

3.4.5.1 Dependências

Esse tipo de identificação tem como dependência um arquivo em formato XML que indica a característica do treinamento feito com as imagens, o uso desta aplicação com funções da biblioteca [OpenCV](#) é utilizado para a identificação de características específicas da imagem podendo ajustar a detecção do objeto com uso de filtros da biblioteca, como demonstrada na figura 21

Figura 20: Identificação da bomba.



3.4.5.2 Saídas

Como saída da detecção o código feito em python te retorna a posição x e y da imagem em pixel e a largura e a altura do *bounding box*, quadrado que identifica o objeto, demonstrado de maneira representativa na figura 22

3.4.5.3 Dependências

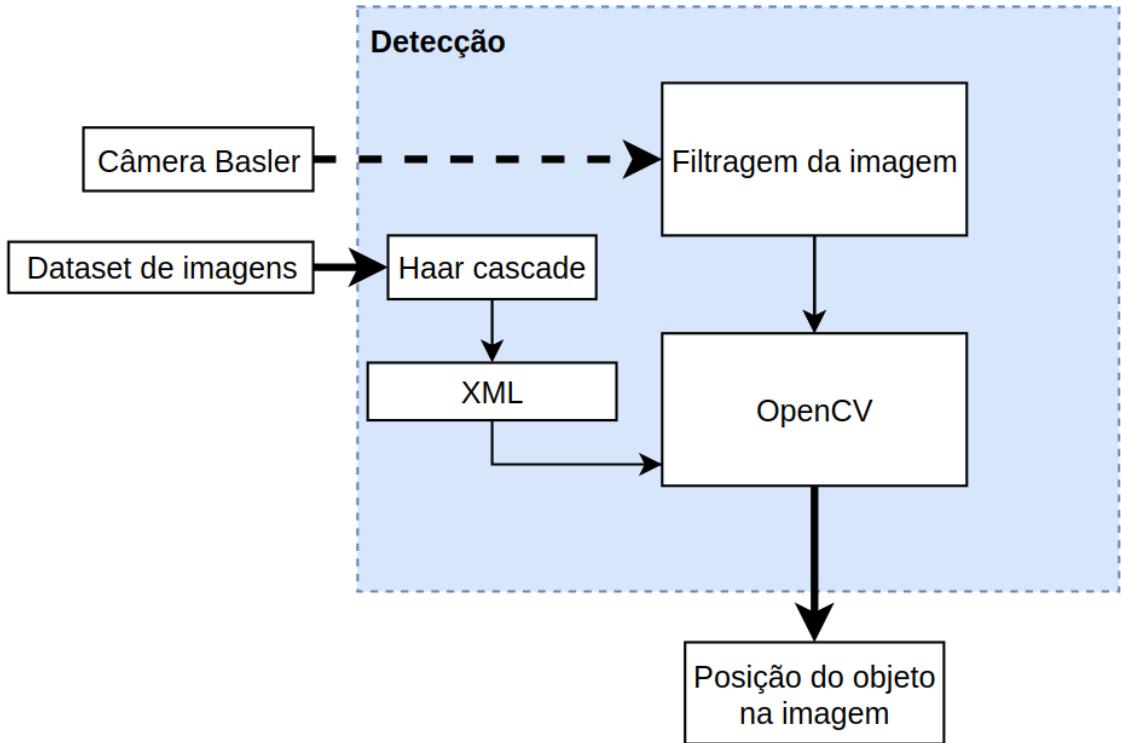
A funcionalidade de percepção depende dos dados oriundos dos sensores que estão incorporados na estrutura do *UGV*.

3.4.5.4 Saídas

As principais saídas desta funcionalidade são:

- *Image msg*: Mensagem que contém as informações sobre as propriedades da imagem que será enviada para localização para que possa ser realizada a detecção visual de objetos.
- *LaserScan msg*: Mensagem que será enviada para a localização no qual é utilizado para odometria.
- *PointCloud msg*: Mensagem que será enviada para o mapeamento pois contem

Figura 21: Fluxograma representando a detecção.



informações do ambiente.

- *Imu msg*: Mensagem contendo informações da orientação, aceleração, velocidade, posição do UGV que será enviada para o sistema de localização.
- *NavSatFix msg*: Mensagem contendo as informações da posição global e relativa do robô e deverá ser enviada para o sistema de localização.

3.4.6 Manipulação

A funcionalidade de manipulação está ligada ao manipulador robótico, em que será responsável pela movimentação do braço. Nela é utilizado o pacote do *Moveit* para controle e planejador de trajetória do manipulador. O diagrama da Figura 23 mostra a aplicação desta funcionalidade.

3.4.6.1 Dependências

A funcionalidade de manipulação fica aguardando uma posição e orientação para realizar seu objetivo.

Figura 22: Representação da identificação da bomba.



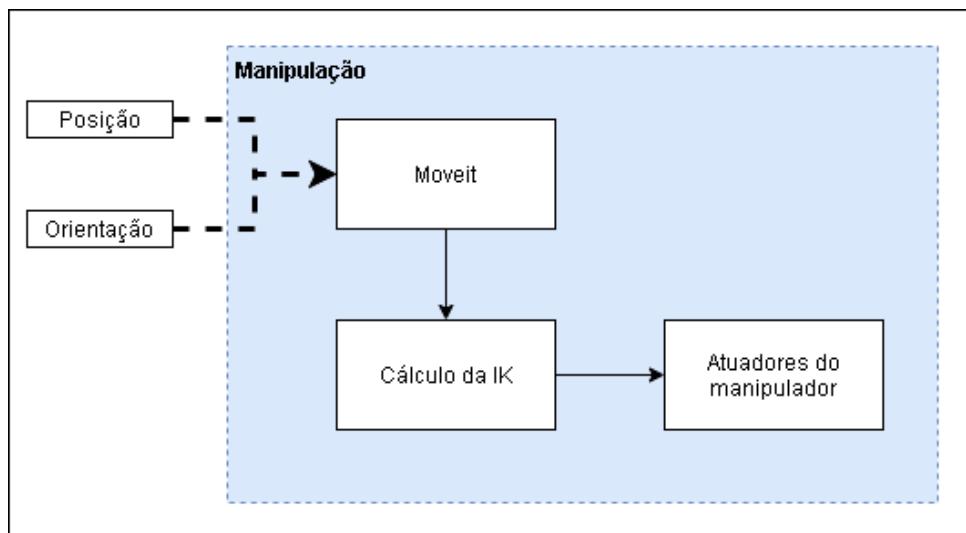
3.4.6.2 Saídas

Como saída, será enviado as posições ângulares de cada junta, estabelecendo assim uma posição e orientação final do *end effector*.

3.5 Arquitetura de software

O UGV 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 realizar conexão das funcionalidades com os modelos externos (*hardware*) de modo que estabeleça uma navegação estável e uma precisão na estimativa da posição do objeto detectado. A Figura 24 mostra a arquitetura simplificada das funcionalidades. Nele é possível ver a interação de uma funcionalidade com a outra. A máquina de estado, representado na Figura 24, está conectado na navegação, na detecção, no manipulador e nos atuadores. Essa configuração é devido a máquina ter controle apenas nessas funcionalidades

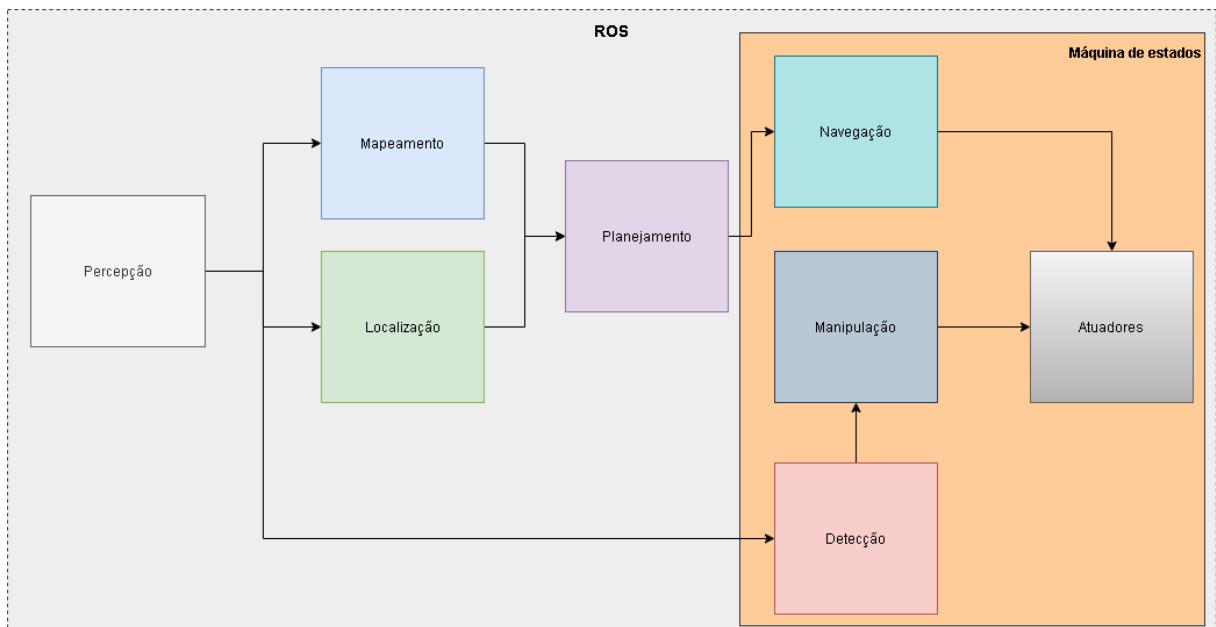
Figura 23: Diagrama da manipulação.



Fonte: Autoria própria.

listadas, e as demais são gerenciadas pelo próprio *framework* do *ROS*.

Figura 24: Integração simplificado das funcionalidades.



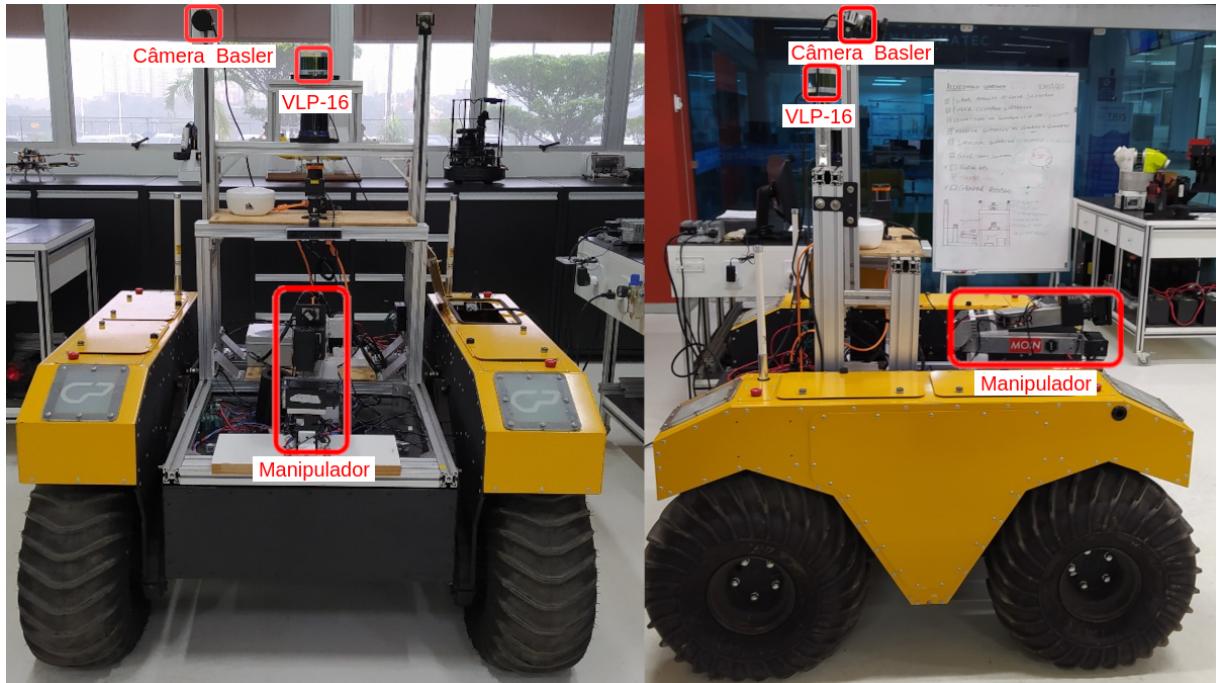
Fonte: Autoria própria.

4 IMPLEMENTAÇÃO

4.1 Estrutura física

A estrutura física do projeto Saci foi baseada no uso do [UGV](#), *Warthog*, demonstrado na figura 25 montado com algumas câmeras e sensores e neste projeto fará uso da câmera Basler anexo B [LiDAR](#) velodyne vlp-16 anexo D.

Figura 25: Vista frontal e lateral da plataforma móvel e seus componentes.



Fonte: Autores.

4.1.1 Câmera

Para prover o sistema com capacidade de detecção de objeto e *tag* para desta forma, obter dados necessários para aquisição de pose e identificação de objetos, utilizou-se uma câmera Basler ace acA4600 (Figura 26) acoplada a lente Kowa LM8HC(Figura 27).

Figura 26: Basler ace acA4600.



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

Figura 27: Lente Kowa LM8HC.



Fonte: Kowa American Corp

4.1.2 LiDAR

O sensor [LiDAR](#) Velodyne VLP-16 usado para detecção de obstáculos e mapeamento. O sensor gera uma nuvem de pontos que caracteriza e identifica o ambiente ao seu redor gerando assim, um mapa com posicionamento de obstáculos em sua trajetória. As demais informações encontram-se no anexo [D](#).

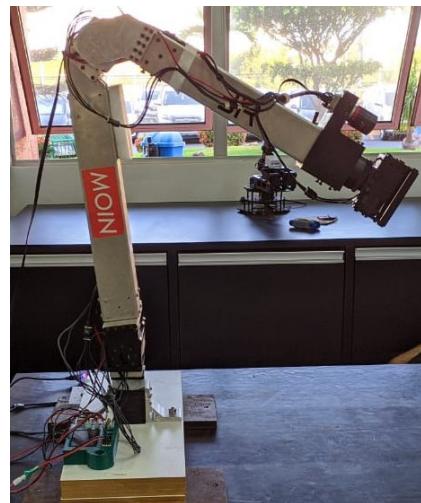
4.1.3 Warthog

Um [UGV](#) de grande porte foi utilizado para aplicação do desafio e comunicação com o uso da *framework* [ROS](#) para transporte de equipamentos e execução do desafio proposto. As características e demais informações encontram-se no *datasheet* anexo [A](#)

4.1.4 Manipulador

O manipulador desenvolvido no projeto Jerotimon demonstrado na figura 28 foi implementado na parte frontal do UGV com a função de desarmamento da "bomba", a partir da aproximação do *end effector* de um fio com coloração específica.

Figura 28: Manipulador.



Fonte: Autores

5 RESULTADOS E DISCUSSÕES

Os resultados obtidos no desenvolvimento do projeto de detecção e desarmamento da bomba consistiram em conhecimento, devido ao protótipo não ter se comportado de forma estável no ambiente real. Não houve coleta de dados a partir dos testes realizados, porém, notou-se que o comportamento instável do UGV estava ligado a estrutura e a falta de controle no movimento. Essa instabilidade no movimento gerava um movimento de balanço e o LIDAR passava a detectar o chão como obstáculo, impedindo que o UGV navegassem por aquela trajetória previamente estabelecida.

A detecção foi feita em 3 formas de treinamento. Na primeira, 3 mil imagens aleatórias, de dimensão 100x100 pixels as quais geravam falso positivos e a filtragem não aprimorava a detecção da "bomba", por esse motivo não havia uma detecção precisa da "bomba". Na segunda, 20 mil imagens do ambiente, medindo 195x108 pixels onde o UGV trabalharia gerou outros falso positivos na imagem, porém isso se resolvendo com filtragem devido a rigorosidade da semelhança do objeto, melhorando a sua detecção. Na terceira, 5 mil imagens variadas do ambiente eram formadas onde o UGV trabalharia de tamanho 500x500 pixels onde teria uma baixa filtragem, haveria uma detecção mais fácil e menos errônea, principalmente no ambiente de trabalho do UGV onde esses testes foram realizados em poucas imagens, e a filmagem com câmera de detecção no ambiente para aprimorar a precisão da identificação da "bomba".

Também, não foi possível utilizar o pacote do EKF para fazer o SLAM devido ao conflito com o próprio EKF do Warthog. Esse método EKF apresentava problemas na transmissão da mensagem e alguns sensores não eram conectados na árvore de TF. Essa conexão é essencial para a troca de mensagens no padrão *ROS*. Sem essas mensagens, os sensores enviam as informações, porém, as funcionalidades não conseguem coletar esses dados. Por motivos desconhecidos, existe um problema estabelecido na criação da árvore de TF, o que impede nas trocas de mensagens. Dessa forma, para que o mapeamento, a localização e as demais funcionalidades consigam utilizar os dados provenientes dos sensores, foi preciso remover o nó do EKF do warthog e conectar novamente a árvore de TF.

6 GESTÃO DO CONHECIMENTO

Neste capítulo estão descritas as lições aprendidas que foram adquiridas durante o processo de desenvolvimento do protótipo desde a etapa conceitual até a fase de testes com o modelo real. Na seção 6.2 está o Diagrama de Ishikawa, onde foi realizada uma análise para descobrir quais foram as causas que levaram a entrega do projeto incompleto.

Na seção 6.3 traz as informações essenciais para realizar a simulação do *UGV* e a utilização modelo real. Este guia tem o propósito de auxiliar o usuário na replicação dos experimentos realizados neste relatório.

6.1 Lições aprendidas

A Tabela 3 mostra as lições aprendidas durante todo o projeto, identificando para cada lição um: Tema, Fase, Impacto, O que ocorreu?, Como foi resolvido?, Resultados e Recomendações para os próximos projetos. Essa análise tem como foco o objetivo deste estudo e a correção dos impactos negativos para os projetos subsequentes.

Tabela 3: Lições aprendidas

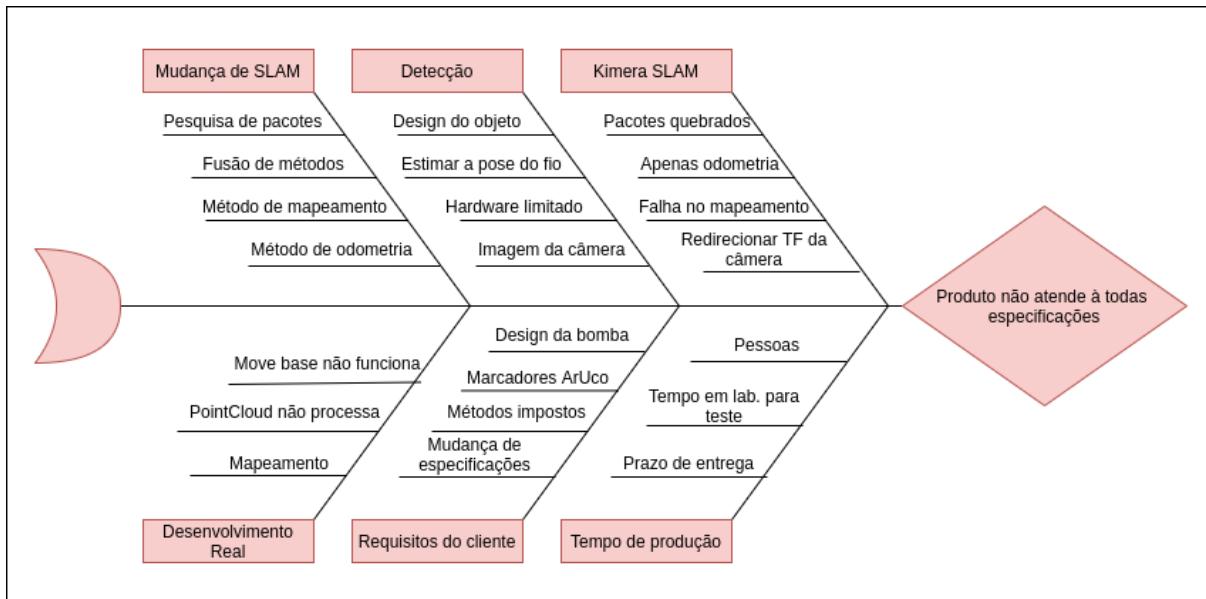
| LIÇÕES APRENDIDAS | | | | | | |
|-------------------|--------------|----------|---|---|--|---|
| Tema | Fase | Impacto | O que ocorreu? | Como foi resolvido? | Resultados | Recomendações para os próximos projetos |
| Gestão | Planejamento | Negativo | Planejamento do tempo de projeto não foi baseado nas ferramentas utilizadas | Foi testado outro método | SLAM não funciona | Conhecer as ferramentas antes de estipular o prazo para o projeto |
| Gestão | Execução | Negativo | Tempo insuficiente para a realização de testes | Tentou-se realizar testes sempre que havia disponibilidade do UGV | Não atende a todas especificações | Estipular prazo maior para a fase de testes |
| Gestão | Execução | Negativo | Alterações nos requisitos do cliente | Testou-se outros métodos de detecção | Não foi possível pegar a pose da bomba | Quando verificar que as modificações solicitadas vão causar atraso no projeto, solicitar nova data de entrega |
| Tecnológico | Execução | Negativo | O frame Map não envia a pose para o move base do robô | Outros métodos foram testados e foi pedido ajuda | Utilizou-se o gmapping para fazer a conexão entre os frames map e odom | Usar um método de SLAM que realize essa conexão |

Fonte: Autoria própria.

6.2 Análise através do Diagrama de Ishikawa

Próximo ao prazo final do projeto foi observado que o mesmo seria entregue porém sem atender a todas funcionalidades do mesmo. Logo, foi realizada uma análise do que causou isso. Para tal foi utilizado o Diagrama de Ishikawa mostrado na Figura 29, onde foram levantados os problemas decorridos da Mudança de SLAM, da detecção, do Kimera SLAM, do Tempo de Produção, dos Requisitos do Cliente e do Desenvolvimento Real, que por sua vez resultaram em um produto que não atende à todas as especificações do projeto.

Figura 29: Análise a partir do Diagrama de Ishikawa.



Fonte: Autoria própria.

6.3 Guia de uso

Para replicar a simulação do *UGV*, ou a utilização do modelo real, é necessário seguir os passos descritos no link https://github.com/Brazilian-Institute-of-Robotics/wbm_ekfslam. Neste link estão todas as orientações para o uso. Recomenda-se a utilização do Ubuntu 18.04 LTS e o *ROS* Melodic Morenia, e que os pacotes mencionados sejam instalados no mesmo *workspace*.

Para criar um *workspace*:

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

```
$ mkdir nomeworkspace_ws
```

- Entrar no *workspace*:

```
$ cd nomeworkspace_ws
```

- Criar a pasta *source*¹:

```
$ mkdir src
```

¹ Pasta onde contém os arquivos fonte.

- Entrar no *source*:

```
$ cd src
```

- Clonar os pacotes citados no link do repositório citado anteriormente

- Retornar para a raiz do *workspace*:

```
$ cd ..
```

- Compilar o *workspace*:

```
$ catkin_make
```

- Ativar o ambiente virtual do *workspace*:

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

Após realizar os procedimentos citados o *workspace* estará configurado para executar a simulação, ou o uso do modelo real. Os *launches* que deveram ser lançados e sua ordem também estão descritos no repositório desse projeto.

7 CONCLUSÃO

O presente relatório retratou desde a concepção, simulação e construção do Saci, um robô autônomo que utiliza a plataforma da *Clearpath*, o *Warthog*, integrado com sensores e um manipulador robótico com 5 *DoF*, e que utiliza o *framework ROS*, capaz de explorar e navegar no ambiente externo de forma autônoma, identificar os *ArUcos* como regiões a não serem exploradas, detectar a "bomba" e efetuar o seu desarme com o manipulador. A estrutura física do robô foi concebida de acordo com os modelos *URDF* projetados.

Como citado anteriormente, o Saci utiliza o pacote *RS2O_Laser_Odometry* para estabelecer a odometria através do *laser* e juntamente com o pacote do *OctoMap* para realizar o mapeamento do ambiente externo. Unindo a odometria do *laser* com a *IMU* utilizando o método de EKF é estimado uma localização mais precisa. Dessa forma, com a localização e o mapa gerado foi feito o *EKF_SLAM*. Também foi utilizado o pacote *RRT* que é o planejador para onde o *UGV* deve ir.

Para possibilitar o uso da ferramenta *Moveit*, arquivos de configuração foram gerados e sua comunicação com o modelo do manipulador 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 Python um pacote capaz de comunicar o sistema de detecção da "bomba" e o sistema de execução do manipulador. A utilização da câmera Basler ace aca4600 possibilitou a identificação da "bomba" com precisão, tornando o sistema capaz de localizar a mesma e planejar uma trajetória que leve o *end effector* do manipulador até o alvo estabelecido.

Para o aprimoramento das funcionalidades do robô é necessário realizar ajustes no pacote de navegação, para que o *UGV* tenha uma estabilidade no seu movimento, melhoramento na detecção do objeto estabelecido e estimativa da posição desse objeto.

REFERÊNCIAS

- ALVES, G. T. M. *Um Estudo das Técnicas de Obtenção de Forma a partir de Estéreo e Luz Estruturada para Engenharia*. Tese (Doutorado) — PUC-Rio, 2005. Citado na página 15.
- BASLER, P. *pylon Camera Software Suite*. 2016. Disponível em: <<https://www.baslerweb.com/en/products/software/>>. Citado na página 40.
- CRAIG, J. J. Robótica. 3^a edição. *Rev. Atual*, 2012. Citado na página 16.
- GONZALEZ, R. C.; WOODS, R. E. *Processamento de imagens digitais*. [S.l.]: Editora Blucher, 2000. Citado na página 15.
- 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 15.
- ISO-8373. *Robots and robotic devices — Vocabulary*. Geneva, CH, 2012. Citado na página 13.
- JR, J. J. U.; DENAVIT, J.; HARTENBERG, R. An iterative method for the displacement analysis of spatial mechanisms. 1964. Citado na página 16.
- 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 12.
- MATARIĆ, M. J. et al. *The robotics primer*. [S.l.]: MIT press, 2007. Citado na página 14.
- NORTON, C. *Os mecanismos da escrita criativa*. [S.l.: s.n.], 2001. Citado na página 17.
- PAUL, R. P. *Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators*. [S.l.]: Richard Paul, 1981. Citado na página 16.
- PIMENTA, T. T. Controle de manipuladores robóticos. *PUC-Rio de Janeiro, Departamento de Engenharia de Controle e automação*, 12 2009. Citado na página 11.
- ROBOS-AUTONOMOS. s.d. <<https://www.voitto.com.br/blog/artigo/robos-autonomos>>. Acessado: 07-12-2020. Citado na página 14.
- SCIAVICCO, L.; SICILIANO, B. *Modelling and control of robot manipulators*. [S.l.]: Springer Science & Business Media, 2012. Citado na página 17.
- SEDIGHI, K. H. et al. Autonomous local path planning for a mobile robot using a genetic algorithm. In: IEEE. *Proceedings of the 2004 congress on evolutionary computation (IEEE Cat. No. 04TH8753)*. [S.l.], 2004. v. 2, p. 1338–1345. Citado na página 15.
- TILBURY, D.; ULSOY, A. Reliable operations of unmanned ground vehicles: Research at the ground robotics reliability center. In: *Proceedings of IARP workshop on technical challenges for dependable robots in human environments*. [S.l.: s.n.], 2010. p. 27–32. Citado na página 14.

WANG, M. et al. Fuzzy logic based robot path planning in unknown environment. In: IEEE. *2005 International Conference on Machine Learning and Cybernetics*. [S.l.], 2005. v. 2, p. 813–818. Citado na página 15.

ANEXO A

Warthog

Maiores informações sobre o *datasheet* da plataforma móvel *warthog* desenvolvida pela *Clearpath Robotics Inc* anexado a seguir.



Hardware Overview

Please see Figure 1 for a view of some of Warthog's key external-facing features.

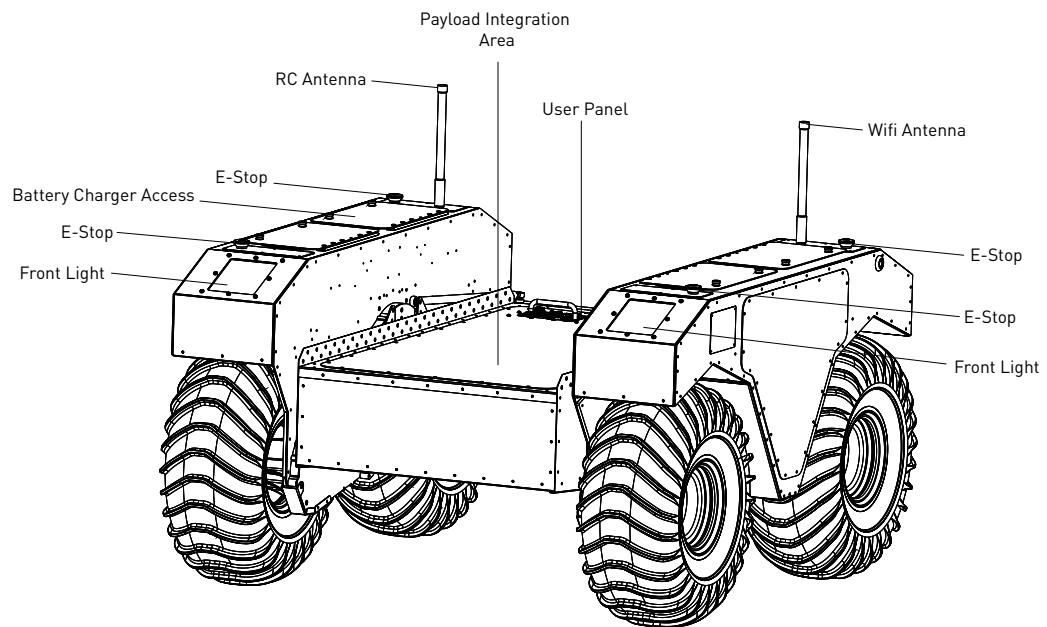


Figure 1: Warthog Hardware Overview



Battery Charger

Please see Appendix for information about the provided charger.

Bilge Pumps

The Warthog has two bilge pumps, with one situated in each of the drive units underneath the motor.

The bilge pumps are used to remove water from the main chassis and drive units during operation in water. At initial startup, an audible sound can be heard from the bilge pumps being initiated. These pumps are an automatic pumping system that check for water levels inside the drive units. The pumps automatically come on every two minutes and check for water levels. If the water level inside the drive unit exceeds a predetermined level, the pumps will remain on, otherwise they will shut off. During prolonged use in water, some water may appear in the drive unit. This is normal and acceptable.

This outlet leads from the pump to the exterior of the Warthog. It is used to remove excess water from the chassis and drive units. No obstructions should be placed in front of or around this area. Obstruction of water flow may result in damage to the internal electrical components and loss of function in the Warthog.

User Panel

The User Panel provides access to the user power panel, as well as USB, serial, and ethernet ports. The power panel can be used to power your payloads. The USB 3.0 and ethernet ports are connected directly to the onboard PC. To connect a device to the onboard network, it's suggested to give it a static IP in the 192.168.131.x subnet, avoiding IPs in use by the following pre-existing devices:

| | |
|----------------|--|
| 192.168.131.1 | Onboard PC (all ports, br0 network interface). |
| 192.168.131.2 | Ethernet-connected MCU. |
| 192.168.131.14 | Front-facing LIDAR (optional). |
| 192.168.131.13 | Rear-facing LIDAR (optional). |

Table 1: Warthog Onboard Network Devices

For more information on electrical integration, please see subsection 4.3 on page 23.

Payload Integration Area

All payloads should be mounted to the central chassis when traversing through water to prevent rolling. The primary payload of the unit should be placed on the central chassis. If necessary loading can be placed on the drive units however payloads should not exceed 50 lbs on each drive unit.

For more information and guidance on mounting payload structures on top of Warthog, please refer to subsection 4.2 on page 22.



Technical Specifications

Key specifications of Warthog are shown in Table 2.

| | |
|---------------------------------|--|
| 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 | 280 kg (551 lbs) |
| Ground Clearance | 254 mm (10 in) |
| Max Payload | 272 kg (600 lbs) |
| Max Incline | 35-45° |
| Max Speed | 18 km/h (11 mph) |
| Suspension | Geometric Passive Articulation |
| Drive Configuration | 4x4 Skid Steer |
| Operating Environment | Outdoor |
| Traction | 24" Argo tire (24" Turf tire or 12" wide Quad Track System optional) |
| Battery Chemistry | AGM sealed lead acid (Li-ion optional) |
| Capacity | 105 Ah at 48 V, expandable to 200 Ah with Li-ion option |
| Nominal Run Time | Lead acid: 2.5 hrs, Li-ion: 6 hrs |
| Charge Time | 4 Hours approx |
| User Power | 5 V, 12 V Fused (24 V, 48 V optional) |
| Control Modes | Remote control, computer controlled velocity commands, 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 Indigo (includes RViz, Gazebo support), Matlab API available |

Table 2: Warthog System Specifications

ANEXO B

Câmera Basler ace acA4600

A ficha resumo contendo as principais especificações da câmera fotográfica Basler ace acA4600 está disposta na figura 30

Maiores informações e desenho mecânico deste componente estão dispostos no *datasheet* anexado a seguir.

Figura 30: Especificações da câmera Basler ace.

| 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 μ 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 [$^{\circ}$ C] |
| Massa | < 80 [g] |
| Dimensões (L x A x P) | 29,3 x 29 x 29 [mm ³] |
| Potência necessária | ~ 2,5 W a 5 V _{DC} , máximo de 2,8 W |

Fonte: Autores.

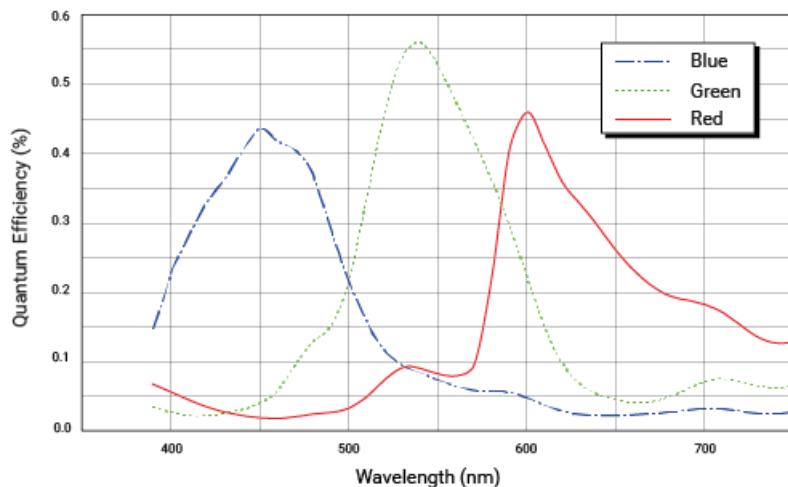
Specifications

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 Pixel Format . |
| 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 |

| | |
|------------------|--|
| | <p>1 opto-coupled output line 2 general purpose I/O (GPIO) lines</p> |
| 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 Basler website . |
| Software | Basler pylon Camera Software Suite (version 4.0 or higher) Available for Windows, Linux x86, Linux ARM, and OS X |
| Accessories | Cables for your camera model Lenses for your camera model Additional accessories for your camera model |

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.

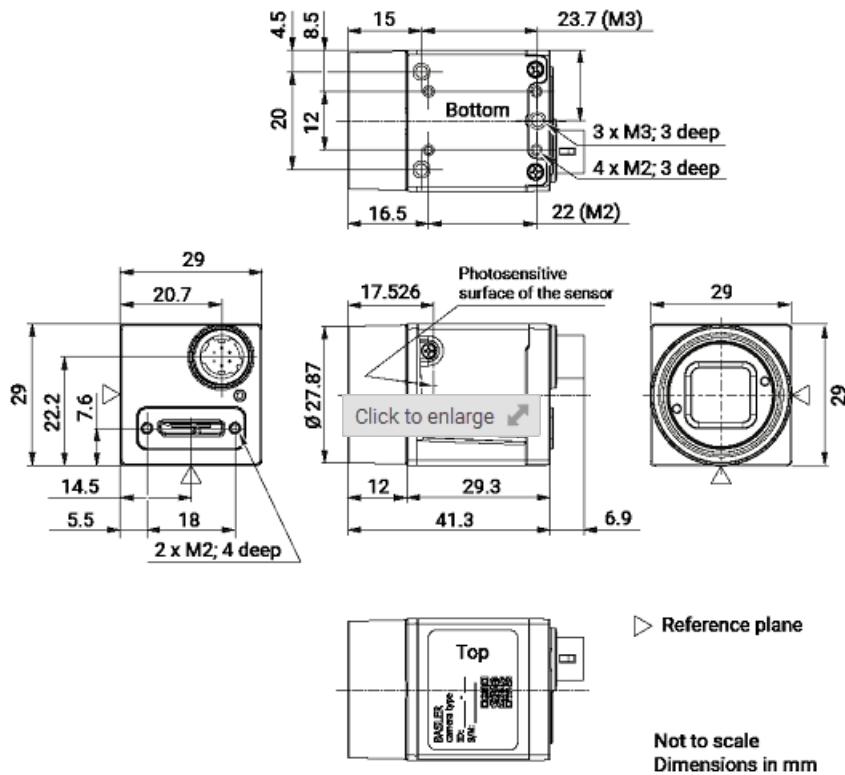
The IR cut filter has the following spectral characteristics:

| Wavelength [nm] | Transmittance |
|-----------------|--------------------------|
| 450–610 | $T_{\min} > 90 \%$ |
| 450–620 | $T_{\text{avg}} > 93 \%$ |
| 645 ± 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

| |
|--|
| DANGER |
| Electric Shock Hazard |
| WARNING |
| Fire Hazard |
| NOTICE |
| 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">• Minimum current: 5 mA• Current draw: 5–15 mA | |

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">• Leakage current: <60 μA. Actual leakage depends on operating temperature and production spread of electronic components.• Maximum load current: 50 mA• Minimum load current: Not specified. Consider the following:<ul style="list-style-type: none">• Leakage current will have a stronger effect when load currents are low.• Propagation delay of the output increases as load currents decrease.• Higher-impedance circuits tend to be more susceptible to EMI.• Higher currents cause higher voltage drops in long cables. | |

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). |

- **Current draw (high-level):** <100 µA
- **Sink current:** Your application must be able to accept 2 mA sink current from the GPIO input line without exceeding 0.8 VDC.

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. |

- **Internal pull-up resistor:** $\approx 2\text{ k}\Omega$, with open collector. Many applications will have to provide an additional pull-up resistor.
- **Residual voltage ("on" state):** $\approx 0.4\text{ V}$ at 50 mA and 25°C (77°F) housing temperature. Actual residual voltage depends on operating temperature, load current, and production spread of electronic components.
- **Leakage current:** <60 µA. Actual leakage depends on operating temperature and production spread of electronic components.
- **Maximum load current:** 50 mA
- **Minimum load current:** Not specified. However, consider the following:
 - Leakage current will have a stronger effect when load currents are low.
 - Propagation delay of the output increases as load currents decrease.
 - Higher-impedance circuits tend to be more susceptible to EMI.

- 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.

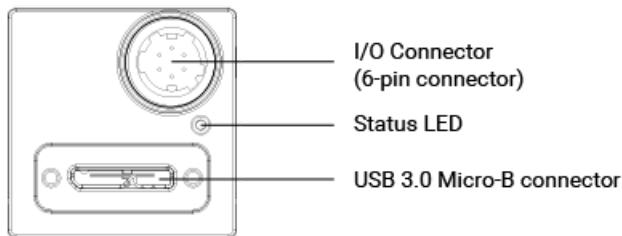
For more information about recommended USB 3.0 cables, see the [Recommended Accessories for Basler USB 3.0 Cameras](#) document.

I/O Cable

- The I/O cable must be shielded.
- The I/O cable must have a cross-section at least 0.14 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.

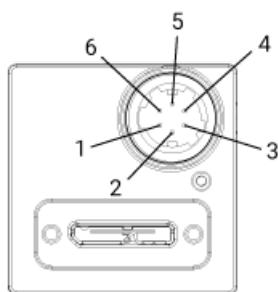
Physical Interface

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

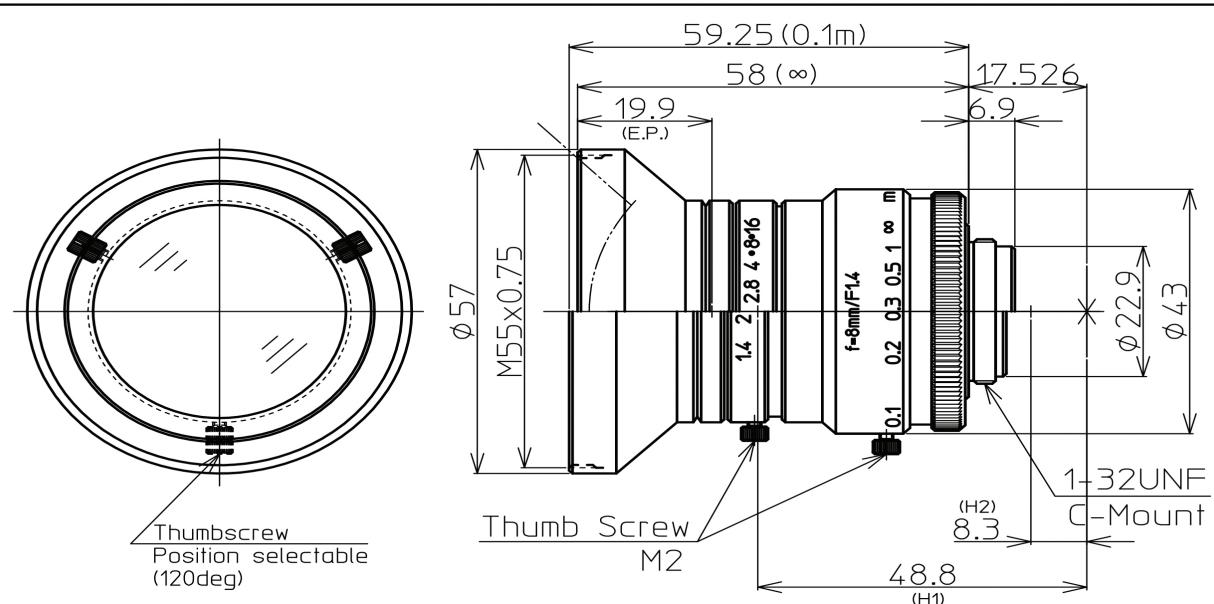
→ See [Safety Instructions for Basler ace Cameras](#).

Installation

ANEXO C

Lente Kowa LM8HC

As principais especificações da lente Kowa LM8HC utilizada em conjunto com a câmera fotográfica está disposta no *datasheet* do anexo a seguir.



| | 1" 1.4 / 8mm C | 1" | 2/3" | 1/1.8" |
|---------------------------|----------------|-----------------|----------------|---------|
| FOCAL LENGTH | f=8mm | | | |
| MAX. DIAMETER (RATIO) | F=1 : 1.4 ~ 16 | | | |
| PICTURE SIZE | 9.6×12.8mm | 6.6×8.8mm | 5.4×7.2mm | |
| ANGLE | | | | |
| VER. ANGLE | 63.0° | 44.7° | 36.9° | |
| HOR. ANGLE | 79.4° | 58.3° | 48.5° | |
| DIA. ANGLE | 92.4° | 70.7° | 59.5° | |
| DISTORTION (TV) | -1.2% | | | |
| SHOOTING RANGE AT | VER. | 143.2mm | 96.0mm | 77.9mm |
| MINIMUM FOCUS | HOR. | 196.0mm | 130.4mm | 105.2mm |
| DISTANCE | DIA. | 247.0mm | 166.2mm | 133.6mm |
| MINIMUM FOCUS DISTANCE | | 0.1m | | |
| FLANGE BACK | | 17.526mm in air | | |
| BACK FOCUS | | 11.2mm in air | | |
| FILTER SCREW SIZE | | M55x0.75 | | |
| FRONT/REAR EFFECTIVE DIA. | | FRONT Ø 37.4mm | REAR Ø 17.5mm | |
| MOUNT | | C-MOUNT | | |
| EXIT PUPIL | | -45.9mm | | |
| TEMPERATURE RANGE | | -10~+50 °C | | |
| STORAGE TEMPERATURE RANGE | | -20~+60 °C | | |
| RESOLUTION | CENTER/CORNER | CENTER 120lp/mm | CORNER 80lp/mm | |
| WEIGHT | | 205 g | | |

NOTE: Specifications and availability are subject to change without notice



| | | | |
|------------|--|-----|---------------|
| | | 1/1 | HR975NCN-3H |
| 2015.7.30 | | | 1" F1.4 f=8mm |
| 2014.12.25 | | | LM8HC |
| 2013.4.15 | | | |

ANEXO D

Velodyne LiDAR Puck LITE

Maiores informações e características físicas demostrado no *datasheet* anexado a seguir.

Velodyne LiDAR®

Puck LITE™

LIGHT WEIGHT REAL-TIME 3D LiDAR SENSOR



Puck LITE



Our Lightest Sensor Ever

Velodyne LiDAR's Puck LITE is a lighter version of the VLP-16 Puck for applications that demand a lower weight to meet their requirements. Aside from the weight, the Puck LITE has identical performance to the VLP-16. The sensor retains Velodyne's patented 360° surround view to capture real-time 3D LiDAR data that includes distance and calibrated reflectivity measurements.

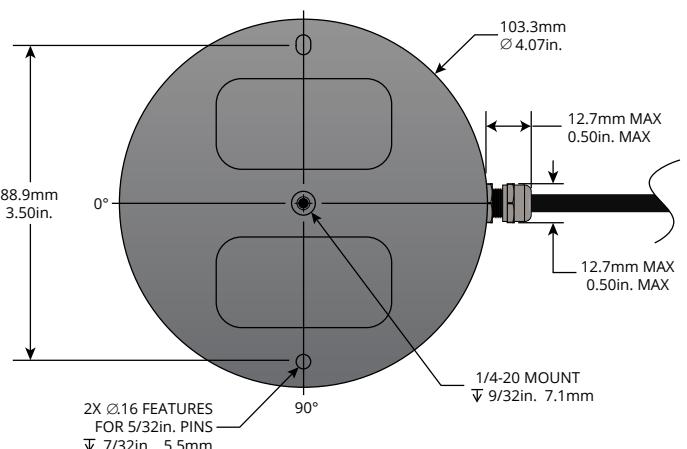
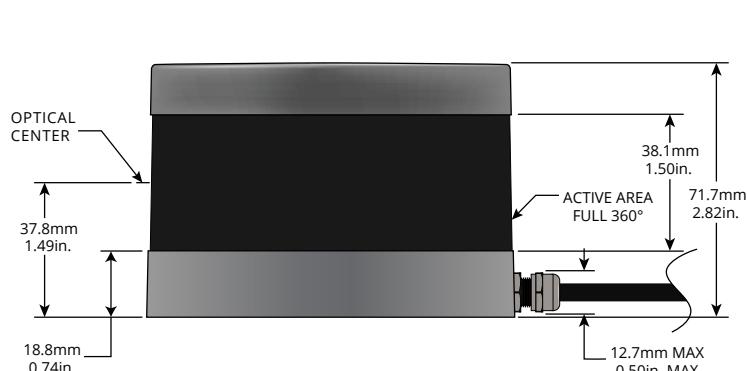
Unprecedented Field of View and Point Density

The Puck LITE has a range of 100 m with dual return mode to capture greater detail in the 3D image with a low power consumption. A compact footprint and an industry leading weight for a LiDAR sensor with high resolution makes it ideal for UAV/drone and mobile applications in the areas of 3D mapping/imaging, inspection and navigation.

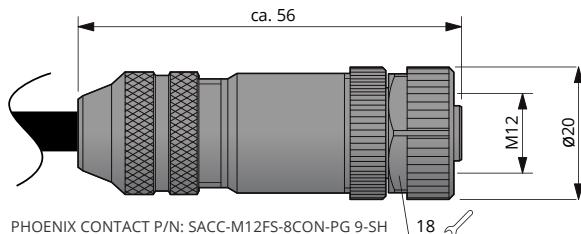
The Puck LITE supports 16 channels and generates approximately 300,000 points/second from a 360° horizontal field of view and a 30° vertical field of view ($\pm 15^\circ$ from the horizon). The Puck LITE has no visible rotating parts and is encapsulated in a package that allows it to operate over a wide temperature range and environmental conditions.



DIMENSIONS (Subject to change)



M12 CONNECTOR OPTION

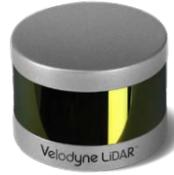


For other connector options contact
Velodyne Sales (sales@velodynelidar.com)

Puck LITE

Light Weight Real-Time 3D LiDAR Sensor

The Puck LITE provides high definition 3-dimensional information about the surrounding environment.



Specifications:

| | |
|---|--|
| Sensor: | <ul style="list-style-type: none">• 16 Channels• Measurement Range: 100 m• Range Accuracy: Up to ± 3 cm (Typical)¹• Field of View (Vertical): +15.0° to -15.0° (30°)• Angular Resolution (Vertical): 2.0°• Field of View (Horizontal): 360°• Angular Resolution (Horizontal/Azimuth): 0.1° – 0.4°• Rotation Rate: 5 Hz – 20 Hz• Integrated Web Server for Easy Monitoring and Configuration |
| Laser: | <ul style="list-style-type: none">• Laser Product Classification: Class 1 Eye-safe per IEC 60825-1:2007 & 2014• Wavelength: 903 nm |
| Mechanical/ Electrical/ Operational | <ul style="list-style-type: none">• Power Consumption: 8 W (Typical)²• Operating Voltage: 9 V – 18 V (with Interface Box and Regulated Power Supply)• Weight: ~590 g (without Cabling and Interface Box)• Dimensions: See diagram on previous page• Environmental Protection: IP67• Operating Temperature: -10°C to +60°C³• Storage Temperature: -40°C to +105°C |
| Output: | <ul style="list-style-type: none">• 3D LiDAR Data Points Generated:<ul style="list-style-type: none">- Single Return Mode: ~300,000 points per second- Dual Return Mode: ~600,000 points per second• 100 Mbps Ethernet Connection• UDP Packets Contain:<ul style="list-style-type: none">- Time of Flight Distance Measurement- Calibrated Reflectivity Measurement- Rotation Angles- Synchronized Time Stamps (μs resolution)• GPS: \$GPRMC and \$GPGGA NMEA Sentences from GPS Receiver (GPS not included) |

63-9286 Rev-H

For more details and ordering information, contact Velodyne Sales (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 ©2018 Velodyne LiDAR, Inc. Specifications are subject to change. Other trademarks or registered trademarks are property of their respective owners.