

INTEGRAÇÃO DE NAVEGAÇÃO AUTÔNOMA NO ROBÔ WARTHOG: DESAFIOS E PROGRESSOS

Erick Suzart Souza¹; Rebeca Tourinho Lima²

¹Graduando em Eng. Computação; Iniciação Tecnológica – SENAI CIMATEC; erick.souza@fbest.org.br

²Centro Universitário SENAI CIMATEC; Salvador-BA; rebeca.lima@fieb.org.br

RESUMO

O robô terrestre Warthog¹, projetado pela empresa Clearpath Robotics, é capaz de navegar terrenos irregulares e executar operações complexas com precisão. O Warthog usa sensores IMU, odometria e GPS para determinar sua posição e orientação no espaço, sendo necessário adicionar um sistema LIDAR e uma câmera estéreo para detectar obstáculos e objetos. O ROS foi usado para integrar e controlar todos os componentes do robô, fornecendo uma plataforma eficiente e flexível para o desenvolvimento, teste e controle do Warthog. A metodologia utilizada simula e testa o robô no Gazebo antes de implementá-lo fisicamente. Além disso, discute os resultados alcançados com a adição dos sensores ao robô físico, bem como a configuração e compatibilidade do ROS 1 e ROS 2 usados para controlar os diferentes sensores. Finalmente, é apresentada a arquitetura de sistema distribuído para o Warthog, que permite o controle dos diferentes sensores por meio de diferentes contêineres Docker.

PALAVRAS-CHAVE: Robótica terrestre; navegação autônoma; ROS; integração de sensores.

1. INTRODUÇÃO

Robôs móveis autônomos são máquinas autônomas que podem navegar e interagir com seu ambiente sem intervenção humana. Esses robôs usam sensores, câmeras e algoritmos sofisticados para se movimentar e concluir tarefas como transportar mercadorias ou inspecionar um local. Com os avanços da tecnologia, os robôs móveis autônomos estão se tornando mais comuns em setores como manufatura, logística e saúde. Sua capacidade de trabalhar em ambientes não estruturados e de se adaptar a condições mutáveis os torna um ativo valioso em muitos locais de trabalho.

O Warthog possui um sistema IMU (Inertial Measurement Unit) que mede a aceleração e a rotação para determinar a posição e orientação do robô no espaço, odometria usada para estimar a velocidade do robô e a distância percorrida, e GPS utilizado para fornecer sua posição geográfica. No entanto, esses sensores não são suficientes para navegar com precisão em ambientes difíceis sendo necessário adicionar também um sistema LIDAR (Light Detection and Ranging) que permite a detecção precisa de obstáculos e a criação de mapas em três dimensões do ambiente. Complementarmente adicionou-se uma câmera estéreo usada para detecção de objetos, além de fornecer informações visuais adicionais ao sistema de navegação. Esse conjunto de sensores permite que o Warthog navegue com segurança em ambientes desafiadores, como terrenos acidentados e fechados, e execute operações complexas com precisão sem colidir com as coisas a sua volta.

O ROS⁴ foi utilizado para integrar e controlar todos os componentes do robô, fornecendo uma plataforma eficiente e flexível para o desenvolvimento, teste e controle do Warthog. Ele permite a integração dos sensores e sistemas de maneira fácil e ainda fornece uma ampla gama de ferramentas de simulação para testar e aprimorar a solução de navegação antes de implantá-la no mundo real.

2. METODOLOGIA

Iniciou-se com a simulação no Gazebo⁵ usando o robô para, posteriormente, implementar a mesma estrutura mecânica e sensores no robô físico. Dessa forma, foram realizados testes simulados e no Warthog para desenvolvimento da solução de navegação

3. RESULTADOS E DISCUSSÃO

Além dos sensores de fábrica do robô, foram adicionados uma câmera de profundidade modelo Mynt Eye⁶ e dois sensores LiDAR: o Velodyne VLP-16⁷ para monitorar o ambiente ao redor do robô e o Quanergy M8⁸ para detectar possíveis obstáculos à frente. A configuração desses LiDARs foi realizada por meio do ROS 2, porém, houve incompatibilidade para adicionar a câmera, que foi configurada no ROS 1 Melodic⁹. O uso de Docker¹⁰ foi essencial para usar as três versões do ROS em conjunto (Kinetic, Melodic e Humble) e, de forma a estabelecer comunicação entre essas versões, utilizou-se o pacote ros1_bridge¹¹.

Com a adição dos sensores ao robô físico, a descrição URDF¹² (Unified Robot Description Format) do mesmo precisou ser alterada para poder simular a configuração no Gazebo, que é o simulador padrão do ROS, evitando o uso do robô físico.

Figura 1. Disposição dos sensores no Warthog.



Fonte: Autores.

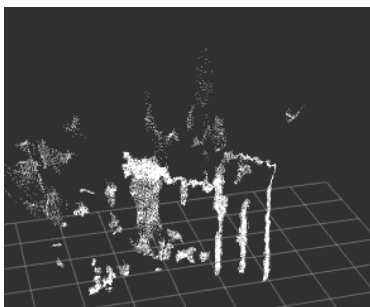
Figura 2. Modelo URDF contendo as modificações.



Fonte: Autores.

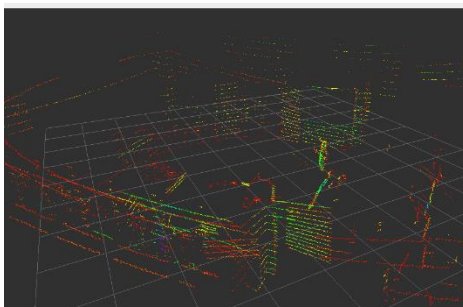
Após essa etapa de configuração, foi possível verificar que os sensores estavam devidamente funcionais, como pode ser visto nas imagens 3, 4 e 5.

Figura 3: teste da câmera estéreo MyntEye no software Rviz¹³



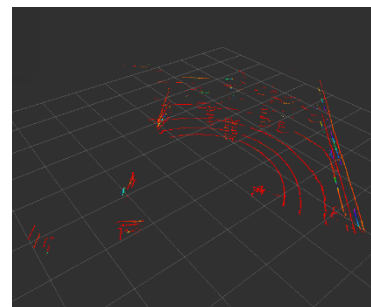
Fonte: Autores,

Figura 4: visualização dos feixes do LIDAR VLP16.



Fonte: Autores.

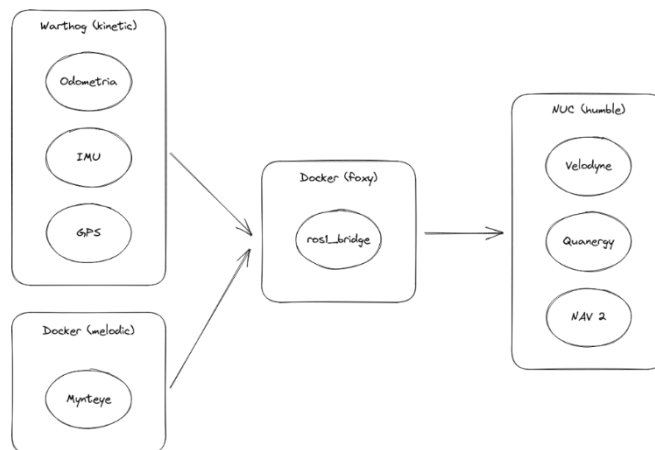
Figura 5: visualização dos feixes do LIDAR Quanergy M8.



Fonte: Autores.

Após verificação dos sensores, foi elaborada uma arquitetura de sistema distribuído, onde a odometria, a IMU e o GPS estarão em execução no computador do robô com a versão ROS 1 Kinetic. Um container Docker acolherá a câmera MyntEye na versão ROS 1 Melodic, e outro contêiner abrigará o pacote ros1_bridge na versão ROS 2 Foxy¹⁴. Por último, um outro computador executará os dois LiDARs e o pacote de navegação NAV 2¹⁵ na versão ROS 2 Humble, como é possível ver na figura 9.

Figura 6: arquitetura elaborada para executar a navegação autônoma.



Fonte: autores

4. CONSIDERAÇÕES FINAIS

A integração da navegação autônoma baseada nas novas tecnologias disponíveis no ROS 2 ao sistema antigo disponível no robô Warthog usando o ROS 1 Kinetic demonstrou ser um processo desafiador. Com a adição de novos sensores, a descrição da URDF precisou ser modificada para simular a configuração no Gazebo e uma arquitetura de sistema distribuído foi desenvolvida. Durante o processo, verificou-se que a câmera de profundidade precisava ser calibrada, embora os sensores tenham sido considerados funcionais. Apesar da navegação autônoma não ser totalmente implementada de acordo com o pacote NAV 2, este projeto proporcionou informações sobre os desafios de integrar novas tecnologias em sistemas legados e a importância de testar e avaliar a funcionalidade dos sensores antes de sua integração.

Agradecimentos

Ao SENAI CIMATEC pela oportunidade de participar de uma iniciação científica, possibilitando uma experiência enriquecedora em minha área de interesse e o desenvolvimento das minhas habilidades em pesquisa. Aos meus mentores e colegas de equipe que me guiaram e apoiaram durante o projeto. A minha orientadora, por todo apoio, incentivo e suporte ao longo do processo.

5. REFERÊNCIAS

- ¹CLEARPATH ROBOTICS. **Warthog Unmanned Ground Vehicle Robot**. Disponível em: <<https://clearpathrobotics.com/warthog-unmanned-ground-vehicle-robot/>>. Acesso em: 31 mar. 2023.
- ²OPEN ROBOTICS. **ROS 2 Documentation: Humble**. Disponível em: <<https://docs.ros.org/en/humble/index.html>>. Acesso em: 31 mar. 2023a.
- ³OPEN ROBOTICS. **ROS – Kinetic Kame**. Disponível em: <<http://wiki.ros.org/kinetic>>. Acesso em: 31 mar. 2023b.
- ⁴OPEN ROBOTICS. **ROS – Robot Operating System**. Disponível em: <<https://www.ros.org/>>. Acesso em: 31 mar. 2023c.
- ⁵OPEN ROBOTICS. **Gazebo – Open Source Robotics Simulator**. Disponível em: <<https://gazebosim.org/home>>. Acesso em: 31 mar. 2023d.
- ⁶Slightech. **MYNT@ EYE S SDK**. 5 mar. 2023. Disponível em: <<https://github.com/slightech/MYNT-EYE-S-SDK>>. Acesso em: 31 mar. 2023
- ⁷VELOCITYNE LIDAR, INC. **Puck Lidar Sensor, High-Value Surround Lidar (VLP-16)**. Disponível em: <<https://velodynelidar.com/products/puck/>>. Acesso em: 31 mar. 2023.
- ⁸QUANERGY. **M8™ LiDAR Sensor**. Disponível em: <<https://quanergy.com/products/m8/>>. Acesso em: 31 mar. 2023.
- ⁹OPEN ROBOTICS. **ROS Melodic Morenia**. Disponível em: <<http://wiki.ros.org/melodic>>. Acesso em: 31 mar. 2023e.
- ¹⁰DOCKER INC. **Docker: Accelerated, Containerized Application Development**. Disponível em: <<https://www.docker.com/>>. Acesso em: 31 mar. 2023.
- ¹¹**Bridge communication between ROS 1 and ROS 2**. Open Robotics, 30 mar. 2023. Disponível em: <https://github.com/ros2/ros1_bridge>. Acesso em: 31 mar. 2023
- ¹²OPEN ROBOTICS. **URDF – Unified Robot Description Format**. Disponível em: <<http://wiki.ros.org/urdf>>. Acesso em: 31 mar. 2023f.
- ¹³OPEN ROBOTICS. **RViz – 3D visualization tool for ROS**. Disponível em: <<http://wiki.ros.org/rviz>>. Acesso em: 31 mar. 2023g.
- ¹⁴OPEN ROBOTICS. **ROS 2 Documentation: Foxy**. Disponível em: <<https://docs.ros.org/en/foxy/index.html>>. Acesso em: 31 mar. 2023h.
- ¹⁵OPEN ROBOTICS. **NAV2: ROS2 Navigation Stack**. Disponível em: <<https://navigation.ros.org/>>. Acesso em: 31 mar. 2023i.