

## Desenvolvimento do Firmware para acionamento e aquisição de dados para veículos autônomos

Marcos V. Cruz<sup>1</sup> (IC), Giovanni Bernardes (PQ)<sup>1</sup>, Rafael Francisco dos Santos (PQ)<sup>2</sup>  
Willian G. Almeida (PQ)<sup>3</sup>

<sup>1</sup>Universidade Federal de Itajubá Campus Itabira.

**Palavras-chave:** Aquisição de dados, controle, gazebo, ros, veículo autônomo.

### Introdução

Este trabalho tem por objetivo realizar a Concepção e Desenvolvimento de um veículo terrestre autônomo e Inteligente. Dentro deste tema de veículo autônomo será implementado o modelo físico do veículo terrestre em um ambiente de simulação no qual possibilita desenvolver as estratégias de percepção e mapeamento. Além disso, é abordado o desenvolvimento do sistema de aquisição de dados e controle do modelo físico a serem utilizados nas estratégias de navegação.

A modelagem física do carrinho de golfe é realizada no ambiente de simulação GAZEBO. Tal modelo em conjunto com os sensores utilizados para a odometria do veículo, podem contribuir para a implementação de pacotes de mapeamento e navegação, tanto em ambiente de simulação quanto em ambiente real, por meio do framework ROS2.

No modelo físico foi desenvolvido o sistema de aquisição de dados e o controle do carrinho de golfe. A parte de aquisição segue com o projeto do chicote elétrico responsável por ser o canal de aquisição de dados dos sensores e controle do veículo. Todo o tratamento dos dados e controle foi realizado através do desenvolvimento do firmware na placa de aquisição de dados que contempla as leituras e escritas digitais analógicas e comunicação do tipo serial, CAN. A placa consiste de um microcontrolador da família Arm Cortex-M3 (stm32f107VC) no qual foi implementado o firmware por meio da IDE STM32CubeIDE e os pacotes FreeRTOS e uROS.

### Metodologia

A implementação física do carrinho de golf segue com o desenvolvimento da placa de aquisição e controle juntamente com o firmware que se encontram nas Figuras 1 e 2, respectivamente.

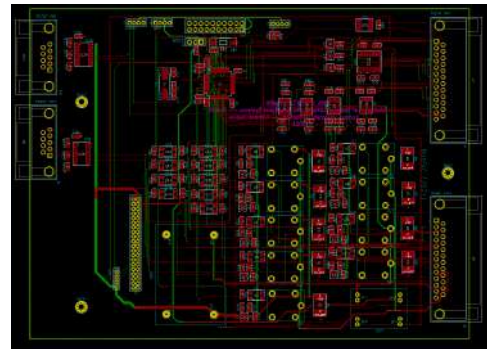


Figura 1 – Placa de aquisição de dados e controle.

Na Figura 1 contém a placa de controle e aquisição de dados do veículo. Esta placa foi desenvolvida com base no acesso aos sinais digitais e analógicos do veículo. Onde os sinais digitais consistem na ligação do veículo, ou seja, o modo de operação e os sinais analógicos do freio, acelerador e alimentação. Estes sinais são processados pelo microcontrolador stm32f107VC que contém todos os periféricos de leitura e escrita dos sinais digitais e analógicos do veículo, bem como a de sensores de posição, velocidade, posição geográfica e orientação. Tais sinais possibilitam a transferência de informações do estado do veículo via comunicação serial, CAN, o que constituem os elementos necessários para desenvolver a odometria de um veículo autônomo.

Na Figura 2 contém o diagrama que representa a implementação lógica do firmware. Este reúne as informações do estado do veículo desde os elementos básicos ( ligado, neutro e desligado) à informações de posição, velocidade e direção para tomada de decisão de alguma ação de controle ou diagnóstico. Neste diagrama contém a representação de como a leitura e escrita estão sendo realizadas no sistema embarcado, a partir dos conceitos do framework ROS2 ( nós, tópicos, serviços), FreeRTOS e informações disponibilizadas dos fabricantes dos atuadores e sensores usados na leitura de posição, velocidade linear e angular (esterçamento) do veículo.

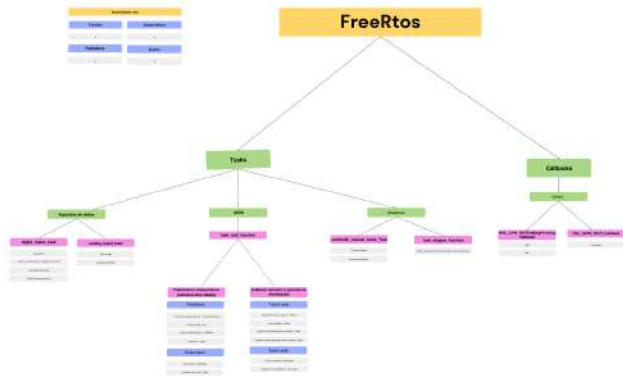


Figura 2 – Diagrama da representação da relação entre os pacotes uROS e FreeRTOS voltados a implementação do firmware do sistema embarcado.



Figura 3 – Pacote de desenvolvimento da comunicação entre o sistema embarcado e o Framework ROS com o pacote FreeRTOS.

Com o pacote do uROS implementado no sistema embarcado, é então realizado sua comunicação com o Framework ROS2 (The Robot Operating System 2) conforme ilustra a Figura 4.

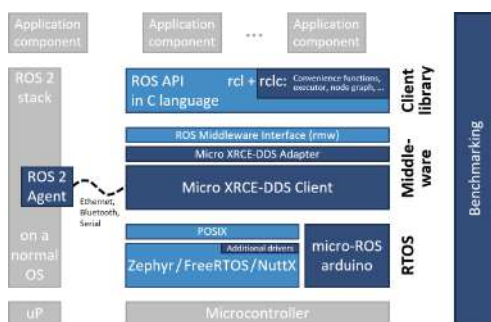


Figura 4 – Estrutura de interconexão entre o uROS e o ROS.

Além da implementação da placa de aquisição e do firmware, foi desenvolvido o chicote elétrico cuja função é conectar e comunicar com todos os componentes do veículo através da placa de aquisição e controle conforme ilustra a Figura 5.

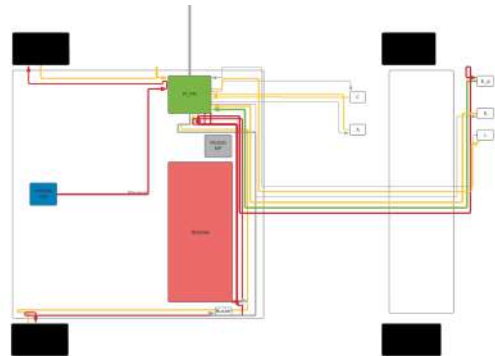


Figura 5 – Esquema do chicote elétrico do carrinho de golf.

O modelo físico do carrinho de golf foi desenvolvido no ambiente de simulação GAZEBO com a integração do Framework ROS2 e Rviz. Estas ferramentas disponibilizam a configuração do modelo físico que se adequa a topologia ackermann e disponibiliza os sensores (lasers, encoders, imu, câmera) para a aplicação de percepção, mapeamento (visualizados pelo Rviz) e navegação de algoritmos desenvolvidos pelo usuário que posteriormente podem ser validados no modelo real.

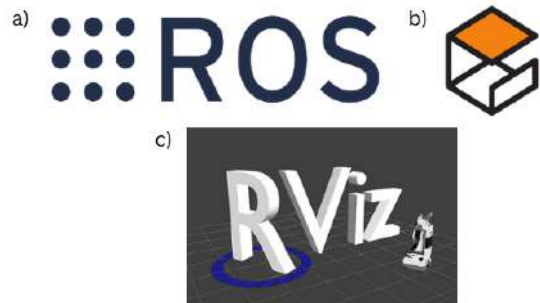


Figura 6 – Ferramentas de aplicação à robótica.

Além disso, estas ferramentas computacionais voltadas à robótica foram o ponto de partida para a compreensão de forma geral do emprego do sistema embarcado e sistemas de aquisição de dados e controle do veículo autônomo como forma de validar as aplicações simuladas em ambiente virtual.

**Resultados e discussão**

Após a implementação do sistema de baixo nível e firmware, foram realizados alguns ensaios com a finalidade de implementar algoritmos do controle de velocidade linear no sistema embarcado que segue no diagrama de bloco da Figura 7.

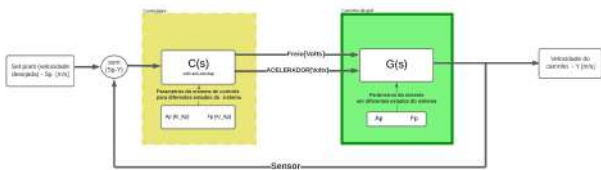


Figura 7 – Diagrama de blocos do sistema de controle de velocidade linear.

No veículo foi realizado o controle de velocidade com o controlador do tipo proporcional que relaciona velocidade de referência com a velocidade medida pelos sensores das rodas (encoders incremental) do veículo (R. Petrella; M. Tursini; L. Peretti; M. Zigliotto; 2007). Na Figura 8 está presente o resultado coletado em campo do controle de velocidade.

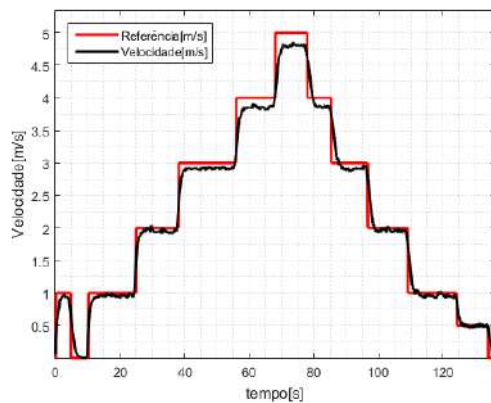


Figura 8 – Resultado da simulação do controle de velocidade linear implementado no sistema embarcado.

Além da aquisição de dados e o controle de velocidade linear no sistema embarcado, foi adicionado ao firmware o controle de direção do veículo. Este realiza a mudança de direção de forma teleoperada (Figura 9) e futuramente por comandos enviados por algum algoritmo de navegação autônoma.

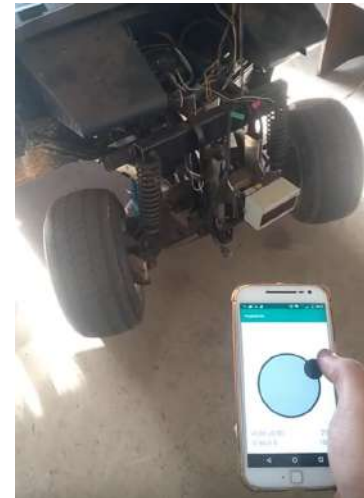


Figura 9 – Controle de direção do carrinho de golfe.

De posse das ferramentas para aplicação em robótica, obteve-se como resultado o modelo físico do carrinho de golfe (Figura 10a) implementado no ambiente gazebo (Figura 10b).



Figura 10 – Carrinho de golfe (a) implementado no ambiente Gazebo (b).

Através deste modelo foram obtidos alguns resultados simulados dos quais realizam o mapeamento de um ambiente virtual (Figura 11) onde o pacote utilizado no mapeamento pode ser implementado no ambiente físico.

Na Figura 11 é apresentada uma simulação que realiza o mapeamento do ambiente através do pacote Cartographer disponibilizado pela Google.

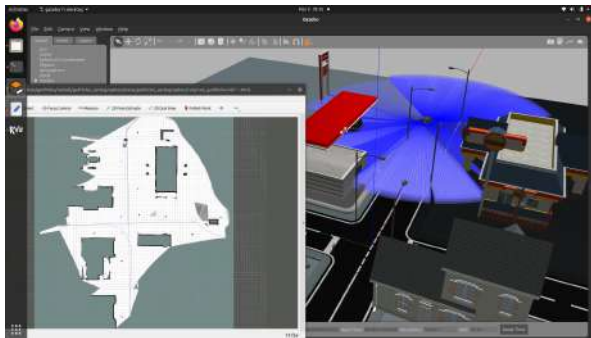


Figura 11 – Simulação do mapeamento e teleoperação do carrinho de golfe.

A ferramenta RQT disponibilizada pelo Framework ROS2 mostra a integração do algoritmo de mapeamento com o ambiente Gazebo (Figura 12) e o software de simulação Rviz utilizado para representar de forma visual as diferentes informações que estão trafegando em um sistema complexo de navegação autônoma.

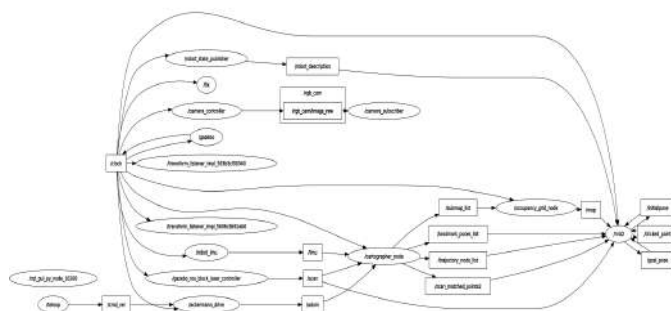


Figura 12 – Representação da interação entre nós e tópicos dos pacotes de simulação do modelo físico e mapeamento.

### Conclusões

As atividades realizadas no projeto mostram que o tema proposto para o desenvolvimento de um veículo autônomo necessita de ferramentas computacionais (ROS, Gazebo, Rviz) com a finalidade de realizar simulações de aplicações robóticas como percepção, navegação e mapeamento que depois podem ser aplicados ao modelo físico. Optar por estas ferramentas computacionais demonstram obter segurança nos ensaios de algoritmos de navegação para qualquer topologia de robô e o custo de realizar simulações com recursos do simulador físico para validação de algoritmos dos quais serão validados no modelo real. Com base nestas ferramentas foi observado que elas auxiliam na compreensão de como devem ser dispostos os sistemas de percepção do veículo para que a

realização de ensaios possam ser conduzidas da mesma forma que nas simulações. E que o desenvolvimento do sistema embarcado possibilita esta interação do veículo virtual com o veículo real.

Em trabalho futuro será implementado o controle de velocidade do tipo controle proporcional integrativo que possui uma característica de robustez e o mínimo de erro entre a referência e a velocidade do veículo ou alguma estratégia que visa obter a rastreabilidade da velocidade desejada, precisão e suavidade (J. E. A. Dias; G. A. S. Pereira; R. M. Palhares; 2015). Além disso, validar as aplicações de navegação e mapeamento no ambiente real.

### Agradecimento

Agradeço à Vale por disponibilizar os recursos para o desenvolvimento deste projeto que pode contribuir para o conhecimento teórico e prático no ambiente acadêmico.

### Referências

J. E. A. Dias, G. A. S. Pereira and R. M. Palhares, "Longitudinal Model Identification and Velocity Control of an Autonomous Car," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 776-786, April 2015, doi: 10.1109/TITS.2014.2341491.

R. Petrella, M. Tursini, L. Peretti and M. Zigliotto, "Speed measurement algorithms for low-resolution incremental encoder equipped drives: a comparative analysis," 2007 International Aegean Conference on Electrical Machines and Power Electronics, 2007, pp. 780-787, doi: 10.1109/ACEMP.2007.4510607.