DESENVOLVIMENTO DO PROTÓTIPO DE HARDWARE E SOFTWARE PARA SISTEMA EMBARCADO DO EMULADOR DE COMPUTADOR DE BORDO DO CAMINHO FORA DE ESTRADA DO CAT793F

Oziel Ferreira da Silva[1] (IC), Giovani Bernardes Vitor (PQ)[1]
[1]*Universidade Federal de Itajubá – Campus Itabira*

Palavras-chave: Interface, Simulação, Usabilidade.

Introdução

Nos últimos anos, com os avanços tecnológicos, tornou-se possível realizar simulações altamente realistas em ambientes virtuais. Isso é particularmente relevante no contexto dos veículos pesados, onde a necessidade de simular o comportamento de caminhões reais em ambientes simulados tornou-se evidente em razão do treinamento efetivo de operadores. Durante essas simulações, o caminhão é capaz de percorrer trajetos que podem replicar situações do mundo real, incluindo cenários que são de difícil reprodução em ambientes reais. Neste processo de capacitação, se faz necessário que o operador esteja hábil a manusear o computador de bordo do caminho fora de estrada CAT793F.

A segurança desempenha um papel fundamental em todas as simulações, especialmente quando se trata de situações que podem ocorrer no mundo real. O aprimoramento contínuo do computador de bordo do caminhão, que atua como o centro de informações para o operador, desempenha um papel crucial na manutenção da vigilância e na preparação dos operadores para situações desafiadoras que podem surgir durante a simulação. O desenvolvimento do computador de bordo interno é essencial para assegurar que o operador esteja completamente consciente das situações que podem ser de difícil identificação.

Neste contexto, o presente trabalho visa apresentar o desenvolvimento do emulador do computador de bordo do caminho fora de estrada CAT793F, envolvendo duas principais etapas. A primeira consistiu em realizar uma revisão sistemática da literatura teórica e conceitual. Durante essa fase, foram analisados diversos artigos, manuais e trabalhos científicos, com foco na pesquisa sobre os *frameworks* ROS2 (do inglês, *Robot Operation System 2*) e QML (do inglês, *Qt Modeling Language*). Esses *frameworks* incorporam uma variedade de ferramentas e pacotes utilizados na comunicação, processamento e visualização de dados em sistemas autônomos e interfaces gráficas interativas, respectivamente. A revisão sistemática da literatura teórica permitiu estabelecer uma base sólida de conhecimento sobre essas estruturas tecnológicas e suas aplicações.

A segunda etapa da pesquisa concentrou-se na implementação prática e na integração desses *frameworks* em um ambiente de desenvolvimento. Isso envolveu a configuração e personalização das ferramentas ROS2 e QML para atender às necessidades específicas do sistema em questão, ou seja, o computador de bordo do CAT793F. Durante essa fase, foi estabelecida uma ponte entre ROS2, Python e QML para a execução do emulador por meio do dispositivo ODROID. Foram realizados testes cuidadosos e ajustes precisos para garantir que a integração e a interação entre esses componentes funcionassem de maneira eficaz.

Por meio desta pesquisa, almeja-se o aprimoramento constante da tecnologia do emulador do caminhão, tornando-o uma ferramenta ainda mais vital na simulação de situações do mundo real e na preparação dos operadores para desafios complexos. Este trabalho contribui significativamente para o avanço do conhecimento e da prática no campo de veículos pesados e simulações, ao mesmo tempo em que fortalece a segurança e a eficiência em situações que podem ocorrer no mundo real.

Metodologia

Revisão Sistemática da Literatura

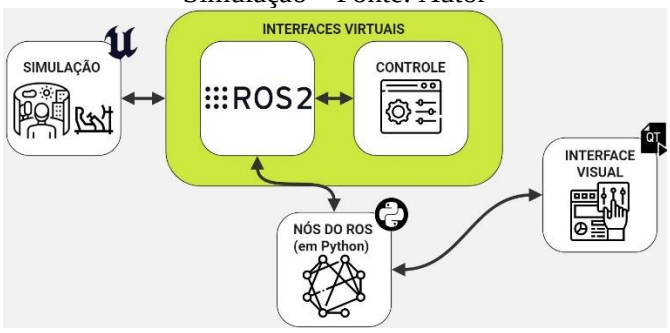
Na fase inicial da pesquisa, foi realizada uma extensa revisão bibliográfica abrangendo os frameworks ROS2 e QML. Essa etapa envolveu a coleta de diversos recursos, incluindo artigos, manuais e trabalhos científicos, provenientes de fontes acadêmicas e bibliotecas digitais. O objetivo era compreender profundamente essas estruturas tecnológicas, tanto em seus fundamentos teóricos quanto em suas aplicações práticas.

"*Exploring the performance of ROS2*" (Maruyama, Kato, Azumi; 2016), ofereceu uma visão abrangente do ROS2, cobrindo seus conceitos fundamentais e fornecendo um guia prático detalhado para sua implementação. Esse recurso foi essencial para compreender a arquitetura do ROS2, comunicação entre nós e desenvolvimento de pacotes. "*Open-Source Tools for Efficient ROS and ROS2-based 2D Human-Robot Interface Development* " (Fabian, Stryk; 2020) desempenhou um papel crucial na criação de interfaces de usuário baseadas em QML para

o controle de maquinaria pesada. Os princípios discutidos nesse artigo foram aplicados na personalização das ferramentas QML no contexto do CAT793F. E, por fim, "*Mobile Robot Simulation and Navigation in ROS and Gazebo*" (Chikurtev; 2020), que aborda sobre simulação de robôs móveis usando o ROS2 e o Gazebo. Embora o foco fosse em robôs móveis, os princípios de integração e teste de sistemas baseados em ROS2 foram aplicados com sucesso na fase de testes e ajustes do emulador do CAT793F. Esses são alguns dos artigos que foram conferidos durante a execução desse projeto.
Esses recursos desempenharam um papel fundamental na pesquisa, fornecendo informações valiosas que contribuíram para o desenvolvimento e implementação bem-sucedida dos frameworks ROS2 e QML no contexto do projeto CAT793F.

Desenvolvimento do módulo de obtenção de dados
A aquisição de dados em ambientes virtuais envolve a coleta constante de informações de sensores integrados, como velocidade, pressão e temperatura, em veículos virtuais, contribuindo para a validação e otimização desses sistemas. Os dados coletados passam por um processo de pré-processamento para eliminar ruídos. Em seguida, são transmitidos para um sistema central, como um computador de bordo, para análise, tomada de decisões em tempo real e exibição por meio de interfaces gráficas. Esse processo é essencial para melhorar a precisão e a eficácia das simulações virtuais. A Figura 1, apresentada a seguir, mostra um diagrama de fluxo de como os dados são passados até a interface do computador de bordo.

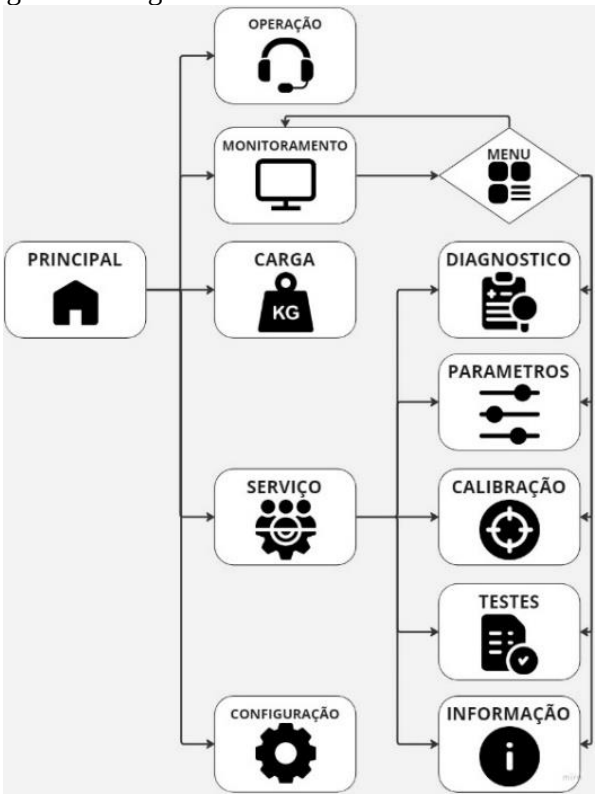
Figura 1 - Funcionamento da Coleta de Dados de Simulação – Fonte: Autor



Dessa maneira, o *software* voltado para a integração entre o ROS2 e Python se faz essencial para a transmissão de dados dentro do sistema de funcionamento da interface visual. As utilidades de bibliotecas geradas a partir do ROS2 para o controle, sistema de erros, sistema de gerenciamento de computador de bordo e outros foram essenciais para o desenvolvimento de um sistema que representasse fielmente as informações que são transmitidas durante a simulação.

Desenvolvimento do módulo de exibição de dados
O módulo de exibição de dados em tempo real desempenha um papel vital ao apresentar informações essenciais para o monitoramento e controle de sistemas simulados. Ele oferece uma interface visual desenvolvida com as bibliotecas Python e QML, permitindo interações intuitivas do usuário por meio de botões e suportando simulações em computador. Além dos recursos predefinidos, há um menu personalizável que permite ao operador escolher quais variáveis monitorar em tempo real, promovendo uma experiência adaptável e focada nas informações mais relevantes para segurança e preferências pessoais.
O fluxo de telas para essa aplicação é representado em um diagrama, conforme apresentado na Figura 2. As telas são projetadas de forma a serem interrompidas em caso de erros de operação, sinalizando a necessidade de atenção imediata por parte do operador. A tela de serviços desempenha um papel central ao coletar e distribuir todas as informações por meio de menus interativos. Esses menus estabelecem conexões entre os componentes ROS2, Python e QML, permitindo a coleta contínua de dados em tempo real e a apresentação desses dados de forma eficaz ao operador.

Figura 2 - Diagrama de Fluxo de Telas – Fonte: Autor



Resultados e discussão

Funcionamento e validação de botões no emulador

Os botões no emulador operam detectando mudanças de estado quando são pressionados, o que é realizado através da leitura dos pinos digitais. Esses botões desempenham um papel importante no controle do sistema simulado, permitindo que o operador envie comandos e interaja com o ambiente virtual.

Figura 3 – Emulador com botões – Fonte: Autor

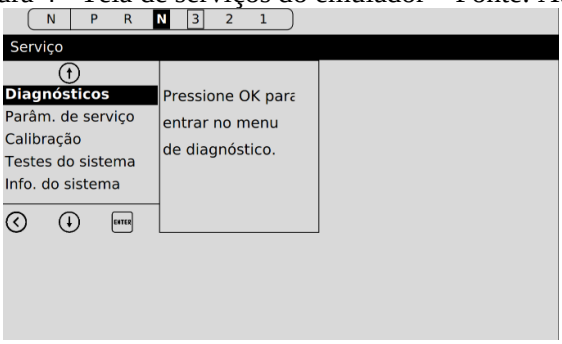
Para garantir a confiabilidade das entradas dos botões, é essencial que sejam debatidos para evitar falsos acionamentos devido a flutuações elétricas nos contatos. Para isso, algumas variáveis são usadas para rastrear o estado anterior dos botões e garantir que apenas mudanças de estado relevantes sejam registradas, contribuindo para a validação e eficácia das entradas dos botões no emulador.

Funcionamento do módulo de exibição de dados

O módulo QML da tela do *Advisor* fornece interface interativa para acessar informações em tempo real. Através de menus, botões e gráficos, o operador pode monitorar variáveis essenciais, como temperatura do líquido de arrefecimento, pressão do óleo, marcha atual selecionada e carga útil, proporcionando uma visão abrangente do desempenho do veículo. Os parâmetros são organizados no computador de bordo com clareza, incluindo caixas de seleção para submenus à direita e informações sobre a marcha atual, horário, dia atual e o estado do parâmetro ARC na parte superior.

Dentro do contexto do sistema de interface, é possível acessar os parâmetros pertinentes ao operador de duas maneiras principais: por meio dos submenus de serviços ou do submenu de seleção de parâmetros destinados à exibição. Esses parâmetros consistem em informações em tempo real provenientes da interface ROS2, sendo adquiridos por meio de conexões que foram estabelecidas de maneira funcional e eficaz.

Figura 4 - Tela de serviços do emulador – Fonte: Autor



No que tange à exibição de informações críticas, é de suma importância destacar que quaisquer erros que venham a surgir em tempo real durante a execução da simulação são tratados de forma abrangente. Esses eventos adversos são registrados de maneira precisa e imediata e, posteriormente, são prontamente apresentados ao operador. Esse processo de exibição de erros ocorre através de uma interface de tela especialmente projetada com a finalidade de proporcionar uma comunicação clara e eficaz. Por meio desta interface dedicada, notificações e alertas são exibidos de modo a permitir que o operador esteja plenamente ciente de qualquer anomalia ou problema que possa afetar a integridade da simulação em andamento.

Assim, a combinação entre a disponibilidade dos parâmetros em tempo real e a pronta identificação e apresentação de erros contribui de forma substancial para a operação suave e controlada do sistema, assegurando um ambiente de simulação altamente responsivo e orientado para a excelência. Essa abordagem garante um monitoramento eficaz e uma resposta ágil a situações adversas, mantendo a integridade do sistema de interface e a confiança do operador no ambiente de simulação virtual.

A detecção e comunicação de erros não se limitam à interface de usuário, sendo processadas em um nível mais profundo, envolvendo o sistema operacional responsável pelo controle dos sistemas autônomos na simulação. Isso mantém a integridade e confiabilidade da simulação, permitindo respostas imediatas a eventos adversos. Essa abordagem garante não apenas a detecção de erros, mas também a capacidade de implementar correções em

tempo real, criando um ambiente altamente controlado e responsivo na simulação.

Figura 5- Tela de Erro - Fonte: Autor

Validação do módulo de exibição de dados

Para assegurar a máxima eficácia e precisão do sistema de interface, uma série abrangente de testes meticulosamente planejados e executados foi conduzida. O processo de validação teve início com a realização de testes no *frontend*, que se concentram na avaliação da interface visual do sistema. Nestes testes, foram efetuados ajustes em tempo real nos parâmetros do veículo dentro do ambiente de simulação. O objetivo primordial era verificar a capacidade da interface gráfica de responder de forma dinâmica e imediata às alterações nos dados inseridos.

Durante esse processo de testes, observou-se com grande satisfação que a interface se comportava de acordo com as expectativas estabelecidas. Cada vez que os parâmetros do veículo eram modificados, a interface reagia prontamente, refletindo instantaneamente as atualizações nos parâmetros do veículo. Esse nível de responsividade demonstrou a robustez e a eficiência do *frontend* do sistema de interface, garantindo aos operadores um ambiente de trabalho ágil e altamente interativo para o monitoramento e controle das simulações em andamento. Esse sucesso inicial nos testes de *frontend* proporcionou uma base sólida para as etapas subsequentes de validação do sistema como um todo.

Conclusões

Ao longo desse trabalho, foram enfrentados uma série de desafios multidisciplinares, abrangendo engenharia, robótica e simulações virtuais. Esses desafios proporcionam experiência e capacitação que são valiosas no campo da ciência, permitindo direcionar com sucesso a pesquisa para o desenvolvimento prático do módulo *Advisor*.

Com trabalho em diferentes áreas englobando *hardware* e *software* foi possível estabelecer uma comunicação de alto nível com o veículo, resultando na criação de ferramentas novas para o sistema. A interface incorpora os parâmetros mais relevantes do veículo, além de permitir o monitoramento contínuo de *status* durante toda a simulação. A abordagem usada proporcionou a capacidade de identificar falhas antes que causassem adversidades significativas, possibilitando uma forma operacional de tomada de decisão e resolução de problemas.

Agradecimentos

Agradeço à FAPEMIG, Vale e UNIFEI, além de também todos os integrantes que contribuíram para a realização do projeto.

Referências

Python Software Foundation. **Python 3.11.5 Documentation**. Disponível em: https://docs.python.org/3/. Acesso em: 04 de setembro de 2023.

Qt. **QML Documentation**. Disponível em: https://doc.qt.io/qt-6/qmlapplications.html. Acesso em: 04 de setembro de 2023.

ROS 2 Documentation. Disponível em: https://docs.ros.org/. Acesso em: 04 de setembro de 2023.

Maruyama, Y., Kato, S., Azumi, T.: **Exploring the performance of ROS2**. Em: 2016 International Conference on Embedded Software (EMSOFT), pp. 1–10 (2016)

S. Fabian and O. v. Stryk, "**Open-Source Tools for Efficient ROS and ROS2-based 2D Human-Robot Interface Development**," 2021 European Conference on Mobile Robots (ECMR), Bonn, Germany, 2021, pp. 1-6, doi: 10.1109/ECMR50962.2021.9568801.

D. Chikurtev, "**Mobile Robot Simulation and Navigation in ROS and Gazebo**," 2020 International Conference Automatics and Informatics (ICAI), Varna, Bulgaria, 2020, pp. 1-6, doi: 10.1109/ICAI50593.2020.9311330.