

CENTRO UNIVERSITÁRIO FEI
MURILO DARCE BORGES SILVA
RODRIGO SIMÕES RUY
IDENTIFICAÇÃO DE DIFERENÇAS DE DESEMPENHO ENTRE ROBÔS REAIS,
CONTEINERIZADOS E SIMULADOS, UTILIZANDO GAZEBO SIMULATOR, ROS
2 E DOCKER.

Análise do Texto realizada por PROF. DR. PLINIO THOMAZ AQUINO JUNIOR

Sobre o TÍTULO

O termo "robôs containerizados" pode causar ambiguidade, já que quem é containerizado é o sistema/software, não o robô físico. Pode gerar confusão sem uma explicação clara no início do texto.

Sugestão de ajuste:

"Identificação de diferenças de desempenho entre sistemas robóticos reais, simulados e com software containerizado, utilizando Gazebo Simulator, ROS 2 e Docker."

Resumo (página 5)

O texto não apresenta claramente o objetivo do trabalho, nem delimita qual será a abordagem metodológica. A divisão do projeto em duas partes (simulação e execução real) aparece de forma solta, sem destacar o que será medido.

No resumo, inserir explicitamente:

"O presente trabalho tem como objetivo comparar o desempenho de sistemas robóticos utilizando ROS2 com e sem containerização, em ambientes simulados e físicos."

E explicitar que métricas serão analisadas (se forem tempo, latência, etc. no TCC2).

Capítulo 1 – Introdução (p. 5)

Problemas de redação: Frases longas e pouco objetivas prejudicam a clareza. Exemplo: "A proposta do projeto é justamente esta: integrar ROS 2 e Docker, explicando os passos utilizados e comparando o desempenho com e sem containerização, sendo dividido em 2 partes: em um simulador Gazebo, e em um robô Turtlebot real."

Sugestão de reescrita: "Este projeto propõe integrar ROS 2 e Docker, explicando os passos utilizados e comparando o desempenho com e sem containerização. A análise será dividida em duas partes: uma simulação no Gazebo e testes em um robô Turtlebot real."

Problemas de estrutura: A seção 1.1 Objetivo é muito breve. Um TCC de nível científico precisa ir além de "documentar e avaliar". É necessário explicar:

O que exatamente será avaliado?

Quais métricas serão utilizadas?

Qual a hipótese ou problema de pesquisa?

O foco da introdução está muito mais em explicar o que é containerização do que em apresentar claramente o problema científico e o gap de pesquisa. A pergunta orientadora do projeto não aparece. Não há menção às hipóteses (ex.: espera-se perda de desempenho? vantagens em modularidade?).

Capítulo 2

Na prática, o capítulo mistura conceitos teóricos (como containerização e meta-SO) com descrição de ferramentas (Docker, ROS2, Gazebo, TurtleBot3 Burger). Isso compromete a clareza da função do capítulo. Recomendo separar claramente os conceitos teóricos dos softwares e ferramentas utilizadas, ou criar um novo capítulo exclusivo para ferramentas.

Alguns trechos são informais ou pouco aprofundados. Exemplos: "mas ainda ter sua consistência e isolamento do Sistema Operacional principal". Isso é tecnicamente vago. A definição de containerização poderia ser mais precisa e respaldada em autores com maior rigor técnico.

"Meta-sistema operacional... Este conceito é aplicado para o ROS." A definição está confusa e sem referência direta. O termo "meta-sistema operacional" pode ser melhor explicado com base em literatura de sistemas embarcados ou ROS.

"um robô que possui o intuito de ser barato... não possui uma grande funcionalidade ou qualidade". Linguagem inadequada para contexto científico. "Baixa funcionalidade ou qualidade" carece de base e objetividade. Deve-se evitar juízo de valor impreciso.

As definições carecem de embasamento acadêmico sólido. Muitas referências são páginas de produto ou manuais online. Por exemplo: O trecho sobre containerização cita (WEN et al., 2023), mas não detalha a definição técnica conforme apresentada no artigo. A descrição de ROS2 é apoiada em um manual, e não em literatura científica sobre sistemas robóticos distribuídos.

Problemas de coesão e clareza. Exemplo de frase problemática: "Para o desenvolvimento do projeto, serão utilizados softwares para que os testes possam ser realizados e analisados. Estes testes serão: Docker...". Aqui, há confusão: os testes não são o Docker, ROS2 ou Gazebo. Estes são ferramentas para realizar os testes. Isso precisa ser reescrito para evitar erro conceitual.

Problemas:

O middleware utilizado (FastDDS, CycloneDDS) não é explicado em nenhuma parte conceitual, mesmo sendo central para a execução dos testes e motivo de um problema técnico importante.

O uso de GitHub Actions e automatização de builds também não é introduzido em "conceitos", embora apareça no capítulo de resultados (p. 11), o que quebra a consistência.

Capítulo 3 – Trabalhos Relacionados (página 9 do TCC).

Pouca profundidade na análise dos artigos

O capítulo cita três trabalhos:

1. Wen et al. (2023) – desempenho de virtualização em sistemas embarcados.
2. Sobieraj & Kotyński (2024) – testes de desempenho do Docker em diferentes SOs.
3. Gao et al. (2023) – desempenho físico de um robô multi-junta.

Embora os artigos sejam relevantes em termos de contexto técnico, a análise de cada um é muito superficial. Há apenas descrições do conteúdo, mas sem comparações críticas com a proposta do TCC. Exemplo de citação superficial: “Este artigo auxilia no entendimento com relação aos tipos de testes que podem ser realizados para analisar o desempenho...”

Faltou:

Quais métricas os autores utilizaram?

O que os autores encontraram?

O que o presente trabalho faz de forma semelhante ou diferente?

Como os resultados anteriores orientaram o desenvolvimento do projeto?

Ausência de justificativa clara para a escolha dos trabalhos. O texto menciona que os artigos “auxiliam no entendimento”, mas não deixa claro por que esses e não outros foram escolhidos. A seleção dos trabalhos deveria ser guiada por: Proximidade metodológica; Semelhança com o tipo de experimento; Aplicabilidade no contexto ROS2/Docker/Gazebo.

Não há citação de trabalhos diretamente relacionados ao uso combinado de ROS2 + Docker + Gazebo: Este seria um ponto central do projeto. Era esperado encontrar: Estudos que tratem da integração de ROS2 e Docker. Trabalhos com benchmarks de performance em simulação robótica containerizada. Pesquisas que testam latência, throughput ou consumo de recursos no uso de ROS2 em containers.

1. Adicionar explicações conceituais sobre:

FastDDS e CycloneDDS

GitHub Workflows e CI/CD no contexto do Docker

2. Expandir os trabalhos relacionados com:

Artigos que tratem de desempenho de ROS2 em containers

Estudos com TurtleBot3 em ambientes simulados ou reais

Avaliações de middlewares do ROS2

Capítulo 4 – Metodologia, localizado na página 10.

Descrição genérica e sem critérios experimentais definidos. O texto apresenta o plano geral de testes, mas não define critérios objetivos para avaliar desempenho.

Faltam respostas para:

O que será medido? (tempo de resposta? latência? uso de CPU? tempo de execução de tarefa?)

Como será medido? (com quais ferramentas? sob quais condições?)

Quantas execuções por cenário? (quantidade de amostras?)

Como os dados serão analisados? (média? desvio padrão? testes estatísticos?)

Exemplo: “Após a análise do desempenho nos testes, os dados serão armazenados, analisados e finalmente comparados.” Isso é vago para um trabalho de natureza científica. Essa etapa precisa ser descrita com muito mais rigor técnico e metodológico.

Falta de estruturação em tópicos/metodologia científica. Uma apresentação mais sólida da metodologia exigiria subdivisões como:

Ambiente de Simulação

Ambiente Físico

Variáveis e Métricas

Procedimentos Experimentais

Ferramentas de Análise

Na frase “A arena possui um caminho, mas o mesmo pode ser alterado com algumas placas que funcionam como paredes, essas paredes permitem a criação de um caminho diferente...”

Sugestão: “A arena é modular, com placas móveis que permitem alterar o caminho do robô, simulando diferentes configurações como labirintos.”

Capítulo 5 – Experimentos e Resultados, que ocupa as páginas 11 a 16.

Falta de análise formal dos dados coletados. Apesar de o título do capítulo ser “Experimentos e Resultados”, nenhum dado quantitativo é apresentado. Não há:

Tabelas com valores de tempo de execução;

Métricas de desempenho (tempo, uso de CPU/memória, latência, estabilidade);

Nenhuma representação gráfica comparativa;

Nenhuma descrição formal de um experimento controlado.

Isso descaracteriza o capítulo como um de “resultados” - ele é, de fato, um relato de progresso da implementação, sem análise científica ainda.

Verifica-se um relato descritivo, não técnico-científico. Exemplo: “Houve certas dificuldades para integrar o ROS2 dentro do contêiner Docker... não mostrava nenhuma....”

Embora seja legítimo relatar problemas técnicos, o texto carece de estrutura científica.

Falta: Descrição objetiva do cenário de teste; Detalhamento de comandos, logs, parâmetros usados; Explicação técnica mais precisa sobre por que o middleware falhava; Citação de documentação ou artigos sobre problemas semelhantes.

O capítulo poderia ser dividido, por exemplo, em:

5.1 Preparação do ambiente

5.2 Experimentos preliminares

5.3 Problemas identificados

5.4 Soluções testadas

5.5 Resultados qualitativos obtidos

Cronograma e próximos passos

Problema grave: ausência de cronograma. O TCC1 exige obrigatoriamente um cronograma de atividades e a sinalização clara do que será realizado no TCC2. No entanto:

Não há seção com esse título;

Não há quadro, tabela ou mesmo texto corrido sinalizando as etapas futuras;

Nenhuma menção direta a entregas previstas, como testes físicos, coleta de dados ou análise estatística no TCC2.

Inserir um quadro ou tabela com as atividades previstas, como:

Integração do ROS2 no robô físico;

Execução dos testes comparativos;

Coleta de métricas de desempenho;

Análise estatística dos resultados;

A consistência temática entre a metodologia proposta e os conteúdos abordados nos capítulos de “Conceitos Fundamentais” e “Trabalhos Relacionados” apresenta falhas importantes no texto analisado. A metodologia, descrita no Capítulo 4, delineia claramente um experimento dividido em duas frentes: (1) testes com robôs simulados no Gazebo e (2) testes físicos com o TurtleBot3, ambos comparando desempenho com e sem containerização via Docker, utilizando o ROS2.

A seção de “Conceitos Fundamentais” aborda, de forma geral, a containerização, o ROS2, o Gazebo Simulator e o TurtleBot. No entanto, essa apresentação sofre de algumas fragilidades. Primeiro, há excesso de superficialidade: os conceitos são apresentados com definições vagas e linguagem pouco técnica, e em alguns casos com construções frasais repetitivas ou informais. Isso prejudica a função central do capítulo, que é servir como base teórica sólida para a compreensão da proposta.

Há dois elementos fundamentais da metodologia que estão ausentes dos “conceitos” e que, portanto, geram uma inconsistência direta: o primeiro deles é o middleware do ROS2, especificamente a distinção entre FastDDS e CycloneDDS, que têm papel importante nos resultados obtidos. Os testes falharam quando o sistema operava com FastDDS e só funcionaram com CycloneDDS, o que revela uma influência direta desse componente na viabilidade da proposta. Mesmo assim, o conceito de middleware e a explicação desses dois mecanismos não aparecem na seção de conceitos, o que representa uma lacuna grave.

Outro elemento ausente nos conceitos é o uso de ferramentas de automação de build, como GitHub Actions ou workflows utilizados para compilar as imagens Docker. Essas ferramentas são mencionadas no capítulo de experimentos como fator essencial para viabilizar os testes, mas não são introduzidas ou conceitualizadas anteriormente. A

ausência desse tópico descontextualiza o leitor e quebra a lógica esperada em um trabalho com sequência científica estruturada.

Quanto ao capítulo “Trabalhos Relacionados”, também há deficiências importantes. Embora sejam apresentados três artigos que tangenciam os temas da containerização e da avaliação de desempenho em sistemas, nenhum deles trata diretamente da aplicação de ROS2 em containers, nem de simulações com Gazebo, nem de robôs TurtleBot. Assim, o capítulo não cobre aspectos metodológicos essenciais da proposta, como: benchmarks de desempenho com ROS2, comparações de latência em ambientes containerizados, ou estudos prévios que enfrentaram o problema de comunicação entre containers e simuladores. Isso enfraquece a justificativa técnica da escolha metodológica do projeto.

Além disso, o middleware não é citado em nenhum dos trabalhos apresentados. Essa omissão reforça a desconexão entre o que está sendo metodologicamente proposto e o que foi apresentado como estado da arte. O mesmo ocorre com o robô TurtleBot3: apesar de ser a plataforma de testes principal do projeto, não há qualquer estudo relacionado que explore seu uso em contextos semelhantes, o que seria esperado em um levantamento bibliográfico robusto.

Vocês poderiam explicar, com base no funcionamento interno do ROS2, por que o FastDDS apresenta problemas de comunicação em containers Docker e por que o CycloneDDS foi mais adequado nesse contexto?

Que métricas de desempenho vocês pretendem utilizar para realizar a comparação entre os sistemas ROS2 nativo, containerizado e simulado, e por que essas métricas são apropriadas para esse tipo de avaliação?

Como vocês pretendem garantir que os resultados obtidos na simulação Gazebo serão válidos e comparáveis aos testes com o robô real na arena física, considerando que simulações muitas vezes não representam fielmente atrasos de sensores, dinâmica real e ruídos?

Quais são as principais vantagens do ROS2 em sistemas distribuídos, e como essas vantagens podem ser exploradas ou limitadas em um cenário onde parte do sistema roda em containers e parte fora deles?

Por que vocês optaram por utilizar workflows no GitHub para automatizar o build das imagens Docker, e quais benefícios essa estratégia oferece no contexto de projetos com ROS2?