

UMA REVISÃO DA TECNOLOGIA DE SIMULADORES SUBAQUÁTICA: AVANÇOS E LIMITAÇÕES

Thâmara Raíssa Monteiro Lins¹; Rodrigo Fonseca Carneiro²

¹ Graduando em Engenharia Mecânica; Iniciação científica – Robótica; thamara.lins@fbter.org.br

² Centro Universitário SENAI CIMATEC; Salvador-BA; rodrigo.carneiro@fieb.org.br

RESUMO

A simulação submarina é uma ferramenta importante para desenvolver e testar algoritmos para veículos submarinos. Nesta revisão, explora-se o estado atual dos softwares de simulação subaquática e suas aplicações. Destacando os benefícios e limitações dos simuladores subaquáticos existentes, como Gazebo, CoppeliaSim, Unreal Engine e Unity discutindo os principais desafios, como física realista e gráficos de imagem com melhores resolução. O artigo conclui com uma discussão sobre o impacto potencial da tecnologia de simuladores subaquáticos na pesquisa e validação de conceitos.

PALAVRAS-CHAVE: Simuladores Subaquáticos; Ambiente virtual; Softwares de simulação.

1. INTRODUÇÃO

Simuladores são ferramentas computacionais que permitem a simulação de situações ou sistemas em um ambiente virtual. Na robótica, os simuladores são amplamente utilizados para modelar o comportamento de robôs e testar algoritmos de controle, num ambiente dinâmico, tais como debaixo de água, aéreo, terrestre, antes mesmo dos testes práticos.¹

Os benefícios dos simuladores são diversos, incluindo a redução de custos e riscos associados a testes em robôs reais, a capacidade de testar uma grande variedade de situações em curtos períodos de tempo e a possibilidade de coletar uma grande quantidade de dados para análise.¹

Os simuladores são ferramentas que auxiliam na concepção e no desenvolvimento de veículos subaquáticos de forma eficiente e segura, permitindo a otimização do desempenho do veículo e seus sistemas. No entanto, os ambientes aquáticos apresentam desafios para testes, devido a suas limitações em relação ao escopo, nível de detalhamento e simulação das forças hidrodinâmicas aplicadas ao robô. A realização de testes e avaliações em um veículo real nem sempre é viável, devido aos riscos de danos físicos, altos custos e aos recursos limitados disponíveis.²

Os simuladores estão relacionados à sua capacidade de acelerar o processo de desenvolvimento de robôs, permitindo os testes e as validações de suas ideias de forma mais rápida e eficiente. Isso leva a uma redução no tempo e nos custos de desenvolvimento de robôs, além de aumentar a segurança e eficiência em ambientes reais.

O objetivo deste estudo é realizar uma análise das opções de software e plataformas de simulação disponíveis para robôs subaquáticos.

2. METODOLOGIA

A metodologia utilizada para desenvolver a revisão bibliográfica sobre o tema de pesquisa, se divide em três ciclos. O método utilizado para realizar a revisão bibliográfica consistiu em refinar pesquisas sobre temas específicos utilizando ferramentas como *Scopus* e *Google Acadêmico*.

Para a realização do primeiro ciclo, foi preciso entender os requisitos necessários para execução do projeto e para realizar a revisão bibliográfica. Para isso, realizou uma busca com as palavras-chave que norteiam o tema da pesquisa na base acadêmica *Scopus* e, assim, conseguir extrair os artigos.

Com o resultado das pesquisas coerentes, foi realizado o refinamento por título, resumo e introdução dos artigos, ao finalizar essa filtragem, alguns artigos foram descartados finalizando assim o segundo ciclo.

No terceiro ciclo, foi efetuado a leitura e a análise completa. Com a realização da leitura dos artigos, foram realizadas anotações, para efetivar interconexões dos principais conceitos que envolvem o tema e produzir anotações no Mendeley. Por fim, a elaboração do estudo do estado da arte foi realizada e resultou no conceito do projeto ao final desta fase.

3. RESULTADOS E DISCUSSÃO

3.1 SOFTWARES DE SIMULAÇÃO

3.3.1 Gazebo

Desenvolvido pela Open-Source Robotics Foundation (OSRF), o Gazebo é uma plataforma de simulação 3D open-source altamente versátil. Com sua ampla gama de funcionalidades, ele pode ser usado para simular uma grande variedade de ambientes, incluindo ambientes subaquáticos.³

O Gazebo utiliza as linguagens de programação, C++ e Python.⁴ Entretanto, ele também tem suporte para outras linguagens, que pode ser usado para fazer interface com o software.⁵

O simulador Gazebo é um recurso importante para a robótica subaquática, já que ele oferece aos usuários a possibilidade de projetar e testar modelos virtuais de robôs e seus respectivos ambientes. O simulador permite que pesquisadores realizem testes em um ambiente virtual, simulando forças hidrodinâmicas, gerando cenários altamente detalhados, suportando o controle de robôs subaquáticos e gerando dados de saída para análises estatísticas. Isso permite que sejam realizadas diversas iterações de testes, aperfeiçoando o produto final até que esteja perfeitamente ajustado às necessidades dos usuários.³

A utilização do Gazebo para realização de simulações subaquáticas é a integração entre o Gazebo e o ROS (Robot Operating System) foi concebida para operar de forma harmoniosa com o ROS, que é uma estrutura bastante utilizada no desenvolvimento de software para robótica. Essa integração possibilita aos usuários incorporar com facilidade pacotes e serviços do ROS em suas simulações no Gazebo, o que é especialmente valioso para simular os aspectos de controle e comunicação de robôs subaquáticos.⁴

O Gazebo é conhecido por sua alta fidelidade física, entretanto sua fidelidade visual pode ser considerada uma das limitações do simulador.³ Ele é popular para projetos de robótica e outras aplicações que exigem simulações precisas, devido à sua capacidade de reproduzir com precisão o comportamento físico dos objetos. No entanto, em comparação com outras ferramentas de simulação, o Gazebo pode não oferecer a mesma qualidade de renderização e detalhamento visual.

3.3.2 CoppeliaSim

CoppeliaSim, anteriormente conhecido como VRep, é um software de simulação multifuncional para robótica, desenvolvido pela Coppelia Robotics sendo utilizado também para a realização de simular robôs e ambientes. Ele suporta a simulação de uma variedade de robôs, incluindo braços robóticos, robôs móveis, veículos aéreos e subaquáticos, além de sensores, como câmeras, sensores de proximidade e sensores de força.⁶

O CoppeliaSim suporta uma ampla gama de linguagens de programação, como: C/C++, Python, Java, Lua, Matlab e Octave. Permitindo que o usuário desenvolva o código de controle na linguagem que preferir e integre facilmente com a simulação.⁶

O CoppeliaSim tem uma variedade de engines de física disponíveis para a simulação de corpos rígidos, incluindo a MuJoCo, Bullet, ODE, Vortex e Newton Game Dynamics. Cada uma dessas engines tem suas próprias características e benefícios, e os usuários podem escolher a que melhor se adapta às suas necessidades específicas.⁷

Com a capacidade de escolher a engine que melhor se adapta às suas necessidades e requisitos específicos, o CoppeliaSim oferece uma simulação de física precisa e confiável. No entanto, é importante observar que o CoppeliaSim não oferece fidelidade visual. Embora possa não ter gráficos de alta qualidade, o simulador é altamente eficiente em termos de simulação de física, permitindo que os usuários testem e desenvolvam seus projetos de robótica com confiança e precisão.

3.3.3 Unreal Engine

Desenvolvido pela Epic Games, Unreal Engine é um software para desenvolvimento de jogos e simuladores que exigem fidelidade e realismo.⁸

O Unreal Engine possui plugin nativo para criar rios, lagos e oceanos, o que permite interações físicas e simulação de fluidos. Através do Sistema de Água, é possível criar ondas realistas, modificar parâmetros oceânicos, simular uma visão subaquática através de pós-processamento e simular forças de flutuação em objetos.

Para o desenvolvimento de cenas de simulação é utilizado o visual scripting, chamado Blueprint, que pode ser usado para programação ou para ampliar as capacidades das classes C++.⁹

O Unreal Engine 4 utiliza o motor de física aberto PhysX, que pode simular com precisão a colisão de corpos rígidos e dinâmica.¹⁰ Já o Unreal Engine 5 apresenta o Lumen, um sistema de iluminação global totalmente dinâmico, e o Nanite, um sistema de geometria virtualizada altamente comprimido que pode melhorar o desempenho da simulação.¹¹ Esses recursos, juntamente com a fidelidade física e o fotorrealismo, tornam o Unreal Engine uma excelente escolha para simulação robótica.

3.3.4 Unity

O Unity, desenvolvido pela Unity Technologies, é um software que tem como objetivo criar jogos em várias plataformas, mas também pode ser utilizado para a criação de simulações interativas¹². Ele oferece uma ampla gama de recursos e ferramentas para o desenvolvimento de simulações, incluindo modelagem 3D, animação, física, iluminação e interação do usuário.

Possui um motor de física robusto, capaz de simular interações muito complexas tais como gravidade, flutuação, arrasto, bem como outras cinemáticas rígidas do corpo e dinâmica de fluidos. É também possível realizar atividades de visão por computador, dada a fidelidade visual do seu renderizador gráfico¹².

A sua implementação é principalmente na linguagem C#, e é possível criar vários sensores, atuadores e física do mundo real. A Unidade também tem o seu próprio armazém de ativos prontos a implementar para acelerar o processo de desenvolvimento quando necessário¹³.

A plataforma tem duas condutas de renderização: URP (Universal Render Pipeline) e HDRP (High Definition Render Pipeline), que devem ser escolhidas no início do projeto. O Unity utiliza o motor físico Nvidia PhysX, que suporta simulações de física multi-threaded, incluindo dinâmica corporal rígida e suave, ragdolls, dinâmica de veículos, partículas e simulação de fluidos volumétricos¹⁴.

4. CONSIDERAÇÕES FINAIS

A utilização de simulações na robótica subaquática é essencial para a criação precisa do comportamento do veículo aquático no ambiente e para testar algoritmos em cenários dinâmicos. Com o avanço dos simuladores, é possível obter resultados cada vez mais realistas, com gráficos e físicas mais precisos. É importante destacar, no entanto, que os simuladores não devem ser usados como substitutos completos para os testes práticos em ambientes reais, uma vez que esses testes são fundamentais para avaliar a efetividade do robô em condições reais.

Os softwares de simulação subaquáticas mais destacados atualmente são o Unity e o Unreal Engine, que oferecem uma variedade de recursos e ferramentas para a criação de mundos virtuais interativos e imersivos, com gráficos fotorrealistas e física precisa.

Com essas ferramentas, é possível desenvolver simulações mais sofisticadas e realistas, permitindo que os desenvolvedores trabalhem com maior precisão e eficácia.

5. REFERÊNCIAS

¹COLLINS, Jack et al. **A review of physics simulators for robotic applications**. IEEE Access, v. 9, p. 51416-51431, 2021.

²KATARA, Pushkal, et al. **Open Source Simulator for Unmanned Underwater Vehicles using ROS and Unity3D**. 2019 IEEE Underwater Technology (UT), Kaohsiung, Taiwan, 2019, pp. 1-7, doi: 10.1109/UT.2019.8734309.

³MANHÃES, Musa Morena Marcusso et al. **UUV simulator: A gazebo-based package for underwater intervention and multi-robot simulation**. In: OCEANS 2016 MTS/IEEE Monterey. IEEE, 2016. p. 1-8.

⁴THEJA, V. Banu et al. **Integration of Gazebo and ROS for Underwater Vehicle Environment**. In: OCEANS 2022-Chennai. IEEE, 2022. p. 1-6.

⁵**About Gazebo**. Gazebo. Disponível em: <https://staging.gazebosim.org/about>.

⁶CoppeliaRobotics. Disponível em: <https://www.coppeliarobotics.com/>.

⁷**CoppeliaSim User Manual**. CoppeliaSim. Disponível em: <https://www.coppeliarobotics.com/helpFiles/>.

⁸UNREAL. **Unreal Engine: The most powerful real-time 3D creation tool**. 2004. Disponível em: <https://www.unrealengine.com/en-US>.

⁹UNDERWATER BLUEPRINT. **UE5: Underwater Blueprint**. 2022. Disponível em: <https://karimibrahim.gumroad.com//raiev>.

¹⁰NVIDIA DEVELOPER. PhysX, s.d. Disponível em: <https://developer.nvidia.com/physx-sdk>.

¹¹UNREAL ENGINE. **Physics in Unreal Engine | Unreal Engine 5.0 Documentation**. 2022. Disponível em: <https://docs.unrealengine.com/5.0/en-US/physics-in-unreal-engine/>.

¹²Unity3d. Disponível em: <http://unity3d.com>.

¹³CHAUDHARY, Akash, et al. **Development of an Underwater Simulator using Unity3D and Robot Operating System**. OCEANS 2021: San Diego – Porto, San Diego, CA, USA, 2021, pp. 1-7, doi:10.23919/OCEANS44145.2021.9706012.

¹⁴UNITY DOCUMENTATION. **Unity User Manual 2023.1 (alpha)**. 2023. Disponível em: <https://docs.unity3d.com/2023.1/Documentation/Manual/index.html>.