**SIMULAÇÃO DE UM VEÍCULO SUBMARINO TELEOPERADO REMOTAMENTE EM UM AMBIENTE SIMULADO COM O USO UNITY E ROS2**

**Matheus Anselmo da Silva** 1; Bruno Schettini Soares Perreira2

1 Bolsista; matheusanselmo97@gmail.com

2 Centro Universitário SENAI CIMATEC; Salvador-BA; bruno.pereira@fieb.org.br

**RESUMO**

O objetivo deste material é apresentar o desenvolvimento parcial da reprodução do veículo submarino BlueROV2 em um ambiente simulado. Os desenvolvimentos foram feitos usando as plataformas ROS2 e Unity. A primeira plataforma é um middleware bastante usado para desenvolvimento de sistemas robóticos, e a segunda é famosa pela concepção de jogos eletrônicos. São apresentadas as estratégias usadas para simular a dinâmica dos propulsores (*thrusters*), elementos responsáveis pela locomoção do veículo, assim como o uso da matriz de alocação juntamente com a estratégia para realizar a tele operação. Para validar a execução do veículo com seus graus de liberdade, são apresentados gráficos que coletam a posição linear do veículo. Dados visuais da câmera também são apresentados, já que se considera replicar a capacidade de sensoriamento do veículo. O objetivo da reprodução do BlueROV2 em ambiente simulado é que ele seja usado em aplicações que têm por alvo estratégias de controle.

**PALAVRAS-CHAVE:** ROS2; UNITY; Tele Operação; ROVs.

**1. INTRODUÇÃO**

A simulação de sistemas robóticos e outros sistemas autônomos têm se tornado uma prática fundamental para a reprodução das dinâmicas desses veículos em diferentes ambientes operacionais. A simulação de sistemas robóticos, especialmente aqueles utilizados em aplicações submarinas, como ROVs (veículos submarinos operados remotamente) e AUVs (veículos submarinos autônomos), pode ser dividida em três principais grupos: a reprodução de elementos sensoriais, como sonares e câmeras; a simulação de propulsores e atuadores; e a implementação de estratégias e ações de controle. Além disso, é necessário simular o ambiente em que o sistema robótico será utilizado, reproduzindo características visuais e físicas semelhantes às encontradas no ambiente real.

O simulador Gazebo é uma das principais ferramentas utilizadas para a simulação de sistemas robóticos submarinos, fornecendo plugins capazes de reproduzir forças físicas, como arrasto (*drag force*), empuxo (*buoyancy*) e forças produzidas pelos elementos de propulsão. Essa reprodução fenomenológica é um dos principais benefícios do uso do Gazebo. Estes plugins são usados no UUV simulador, uma famosa plataforma para simulação de veículos submarinos juntamente com o ambiente submarino. No entanto, em termos de aplicação de visão computacional, o Gazebo tem algumas limitações em comparação a outras ferramentas, como a Unity. O Unity possui uma grade variedade cores, texturas e efeitos visuais que podem ser aplicados em diversos objetos.

A execução de uma ação de teleoperação de um veículo submarino em uma plataforma que possa reproduzir a física do ambiente e os aspectos visuais com grau de similaridade com o ambiente real pode ser de grande para o desenvolvimento de um sistema robótico submarino. Esses aspectos visuais podem ser direcionados para aplicações de visão computacional, enquanto a teleoperação pode ser usada para implementar estratégias de controle nos veículos.

O objetivo deste trabalho é apresentar a simulação do veículo submarino BlueROV2, representando na Figura 1, utilizando como ferramentas o ROS2 (Robot Operating System) e a plataforma de desenvolvimento de jogos Unity. O material apresenta o desenvolvimento tanto da dinâmica submarina quanto dos aspectos visuais do ambiente simulado.

Figura 1: Veículo submarino BlueROV2

Moto azul e preta

Descrição gerada automaticamente com confiança média

Fonte: https://bluerobotics.com/

**2. METODOLOGIA**

Os propulsores são elementos fundamentais para proporcionar movimento aos veículos submarinos não tripulados. A virtualização destes elementos no Unity, visando à reprodução da geração de forças, foi fortemente baseada na aplicação realizada para o UUV Simulator. Para reproduzir a dinâmica dos propulsores, optou-se pelo modelo de ordem zero, onde o sinal de entrada é igual à saída. O sinal de entrada corresponde à velocidade, enquanto a saída corresponde à velocidade de rotação angular. Para determinar a força gerada pelos propulsores, utilizou-se o modelo de conversão básico, proposto por Youguer, segundo o qual a potência produzida é o produto da velocidade angular da hélice do propulsor pelo seu valor em módulo. Além da virtualização dos propulsores, empregou-se uma matriz de alocação de propulsores, baseado no trabalho de 2, para selecionar a força ou torque necessários para o veículo simulado realizar os movimentos.

O BlueROV2 apresenta quatro graus de liberdade: surge, responsável pelo movimento linear no eixo x; *sway*, movimento lateral linear no eixo y; *heave*, aplicado aos deslocamentos no eixo z; e *yaw*, correspondente à rotação no eixo do veículo. Visando realizar a teleoperação, permitindo que o operador utilize os quatro graus de liberdade para a navegação do veículo, desenvolveu-se um nó em ROS2 para receber os comandos do operador e utilizá-los como entrada para a matriz de alocação de propulsores. A Figura 2 ilustra o esquemático da comunicação entre o ROS2 e o Unity, sendo apresentado o ROS TCP-CONNECTOR 3, pacote de softwares responsável pela troca de dados entre as plataformas.

Com o objetivo de reproduzir a dinâmica submarina, com a presença das forças de arrasto e empuxo foi utilizado o pacote de *softwares, DWP* 4.

Figura 2: Esquemático de comunicação entre o ROS2 e o Unity.

Diagrama

Descrição gerada automaticamente

Fonte: Própria

**3. RESULTADOS E DISCUSSÃO**

A simulação do BlueROV2 foi executada no ambiente Unity. Os movimentos de *surge* e *sway* foram executados com sucesso, porém a dinâmica das forças de empuxo reproduzidas pelo *asset* DWP não permitiu a execução adequada do movimento de *heave*. O gráfico da Figura 3 apresenta a posição do veículo após receber entrada para os graus de liberdade de surge, *sway* e *heave*. É importante destacar que no Unity, o eixo z é associado aos movimentos laterais, enquanto o eixo Y está associado ao movimento de *heave*.

Figura 3: Gráfico da posição linear para os movimentos surge, *sway* e *heave*.

Gráfico, Histograma

Descrição gerada automaticamente

Fonte: Própria.

A realização do movimento *heave* do veículo simulado não foi bem-sucedida devido às características do ambiente em relação às forças de empuxo. No Unity, o asset DWP não apresenta uma hidrodinâmica submarina coerente o suficiente para executar simulações de ROVs. Para a representação visual, a simulação apresentou bons resultados e uma textura bastante semelhante aos ambientes reais, que pode ser notado na Figura 4, onde a imagem de barris, que estão posicionados no ambiente submarino é capturada.

Figura 4: Imagem capturada em ambiente submarino.

Uma imagem contendo barril, cheio, comida, garrafa

Descrição gerada automaticamente

Fonte: Própria.

**4. CONSIDERAÇÕES FINAIS**

A aplicação da teleoperação para o veículo submarino BlueROV2 foi realizada com sucesso para dois graus de liberdade, surge e *sway*. Contudo, a reprodução do movimento *heave* não obteve êxito devido às limitações do *asset* DWP em reproduzir a dinâmica dos ambientes submarinos com fidelidade adequada. A adequação deste *asset* pode ser considerada para simulações com um nível de fidelidade mais elevado.

Os graus de liberdade disponíveis podem ser empregados para simular operações de teleoperação do veículo em proximidade da superfície. Além da teleoperação, técnicas de controle também podem ser testadas, ainda que a ausência de ações de controle para o movimento *heave* deva ser levada em consideração.

**5. REFERÊNCIAS**

1MANHÃES, Musa; SCHERER, Sebastian; VOSS, Martin; DOAUT, Luiz; RAUSCHENBACH, Thomas. (2016). **UUV Simulator: A Gazebo-based Package for Underwater Intervention and Multi-Robot Simulation**. 10.1109/OCEANS.2016.7761080.

2FOSSEN, T, I. Handbook of Marine Craft Hydrodynamics and Motion Control, Wiley, 2011.

3YOUNESY, **Hamid. Unity-Technologies / ROS-TCP-Connector** Disponível em: https://github.com/Unity-Technologies/ROS-TCP-Connector. Acesso em 08/04/2023.

4DYNAMIC WATER PHYSICS. **Dynamic Water Physics Documentation.** Disponível em: http://dynamicwaterphysics.com/doku.php. Acesso em: 08/04/2023.