When VR meets Underwater Robots: Creating an Underwater Hull Cleaning Simulator

Wei Jian Chan* Po Yi Wu† Abu Bakr Azam‡ Yiyu Cai§

School of Mechanical and Aerospace Engineering
Nanyang Technological University

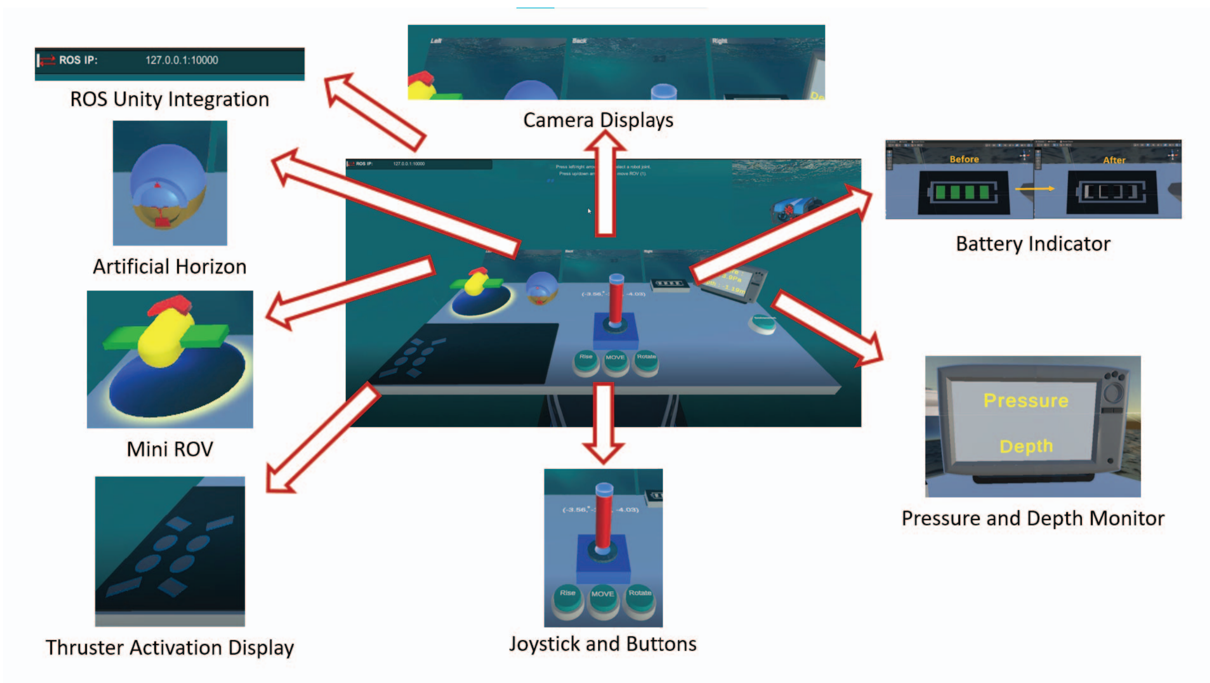

Figure 1: Various VR UI Elements of the Underwater ROV Simulator

Abstract

Shipping is one of the most important transportation methods used in global trade. Underwater hull cleaning increases the efficiency and decreases fuel consumption by 9.6% due to the removal of marine foul which reduces friction and improves vessel hydrodynamics. With the advancement of technologies, underwater cleaning robots are used to reduce the need of the human diver to clean the hull. While the use of cleaning Remotely Operated Vehicles (ROVs) may have its merits, there is a difficulty in training people that can skilfully control the cleaning ROV. This work attempts to develop a novel simulation technology combining VR and underwater robotics for the training of hull cleaning operators. Specifically, the work investigates the integration of Unity and the Robot Operating System (ROS).

A proof-of-concept VR simulator is implemented to control an underwater ROV, model the underwater environment and the ship hull with the aid of important sensors such as a SONAR and Inertial Measurement Units.

Index Terms: Hull Cleaning—Remotely Operated Vehicles—Robot Operating System—Virtual Reality

1 Introduction

Ship hull cleaning is a very crucial process in the maritime industry. It is an effective way for operators to increase energy efficiency and reduce fuel and carbon emissions. The bio-films attached to the hull are microorganisms that clump together and attach themselves as a group. This leads to increased drag resulting in a fuel penalty of approximately 20%. Therefore, the underwater surfaces of ship vessels need to be kept clean for efficient operation, ultimately reducing cost and carbon footprint [27].

Traditionally, ship hull cleaning was done manually. This involves the direct removal of marine growth by divers. Hand scrapers, chippers, and nylon-bristled brushes are the main tools for this type of work. Manual cleaning has several drawbacks such as low efficiency and limitations of divers. A survey conducted by Song and Cui [29] indicates there may still be around 40% of species attached to the hull even after cleaning. Apart from fatigue, divers may only be able to clean when the conditions of diving are safe, i.e. calmer sea and weather conditions. Hence, the use of an underwater Remote Operated Vehicle (ROV) for ship hull cleaning can be a better

*e-mail: chan0898@e.ntu.edu.sg
†e-mail: pwu008@e.ntu.edu.sg
‡e-mail: abubakr002@e.ntu.edu.sg
§e-mail: myycai@ntu.edu.sg

2771-1110/22/$31.00 ©2022 IEEE
DOI 10.1109/ISMAR-Adjunct57072.2022.00082

2022 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct) | 978-1-6654-5365-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISMAR-ADJUNCT57072.2022.00082

alternative.

An ROV can offer cost-efficient and eco-friendly hull cleaning services. Australia has strict bio-fouling management standards has approved the use of the HullWiper ROV to clean the hull of the ships [6]. This ROV reduces the risk of damaging the anti-fouling coatings by using adjustable seawater jets under variable pressure as the means for cleaning [10]. Using ROVs for hull cleaning purposes brings several advantages. Firstly, it is cost effective as compared to providing the right equipment for divers [2]. Secondly, it comes with recording capabilities which can capture operations footage. Thirdly, an ROV can provide extended dive time as compared to sending a human diver. Finally, the design of an ROV comes in all shapes and sizes with minimum maintenance operable on areas which are deemed dangerous for human divers.

VR uses innovative technologies in both software and hardware to create a digitally simulated environment. Statistics have shown that VR is a growing at a very fast pace with a market size of US$ 5 billion in 2019 to a predicted increase to US$ 12 billion in 2024 [20]. Leading software companies like Meta, Apple, Microsoft, and Google has also been investing heavily in VR [12].

There are many industries that have adopted the use of VR in several sectors like education, real-estate and education amongst others [22]. This is because of the enormous benefits VR brings, such as virtual test drives for the automotive industry [21] and virtual house tour for real estate viewing [5].

While there are highly advanced control and communication-based software for underwater ROVs like ArduSub, according to our knowledge, VR-level visualization may not be currently possible using its built-in capabilities.

A survey of articles confirms the benefits of using head mounted VR as opposed to using a multiple monitor setup, but only some works place an emphasis on the underwater ROV control cockpit design [24, 26]. Cruz et. al.'s work [24] is based on the use of VR for intervention tasks, while Garcia et. al.'s work [26] is centered on the creation of a natural user interface (UI). Elor et. al.'s work [25] focuses on devising the best method of displaying camera footage from ROVs to the operator but not on the development of the VR UI itself.

The objective of this study is to develop a VR UI for controlling virtual ROVs while also creating a framework to seamlessly connect to a real ROV using the Robot Operating System (ROS). Additional work with regards to sensor simulation, like that of a SONAR will also be discussed. The scope of this study will be limited to ship hull cleaning applications.

2 Fundamentals

2.1 Unity for Immersive and Interactive VR

Unity is cross-platform game engine first released in 2005. It can also be used in developing realistic simulations. It has a relatively mature ecosystem, and the assets store several assets available for developers. The physics engine in Unity is Nvidia PhysX for 3D physics and Box2D for 2D physics. Unity 2021 is currently used for the development of this study.

Unity supports the Mixed Reality Toolkit (MRTK). It is a project driven by Microsoft that provides components and features in assisting the development of VR and augmented reality (AR) projects [13]. Some of its functions include providing the cross-platform input system and building blocks for spatial interactions and UI enabling rapid prototyping via in-editor simulation. It operates using an extensive framework that allows developers the choice to swap out core components. MRTK also provides us with example scenes to interact and test out all the different UI blocks it provides.

2.2 Ocean Simulation

The ocean simulation package implemented in the application is Crest [1]. It is an advanced ocean renderer which uses the Gerstner wave model.

Gerstner waves are also known as trochoidal waves. It forms sharper crests by moving the vertices toward each crest [1]. This can represent the wave movement more realistically as compared to other waves, like the sine wave whose points are too round and not steep enough to represent a realistic ocean wave, causing repetitive issues in simulating large areas.

2.3 ROS-Unity Integration

ROS is an open-source framework for building robot applications, it is widely used by many companies in robotics and automation. Unity also officially announced the support of ROS [19]. ROS Noetic is currently used for this study.

Unity can either use the publishers and subscribers to send or receive continuous streams of data to and from ROS, or it can use services which send or receive data on request. For example, sensory data may be received continuously using subscribers, while commands to thrusters may be sent using services.

3 VR Simulator Design

The application is designed to operate an inspector ROV overseeing a cleaning ROV. Due to large tethers of the cleaning ROV, and the difficulty to steer it using the onboard sensors, the inspector ROV is placed at a distance, capable of overseeing the cleaning ROV progress. A simple visualization of the setup is shown in Fig. 2

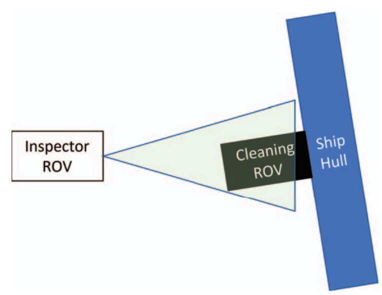

Figure 2: Illustration of the Simulator Setup

The main devices used for this are the Oculus Rift CV1 [15] for the display and the Leap Motion camera [18] for hand-tracking.A high-level system architecture is shown in Fig. 3.

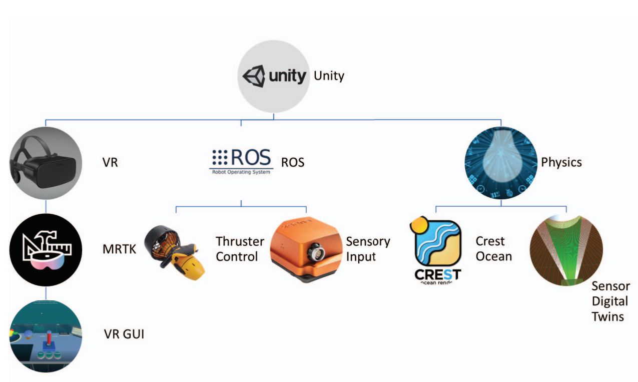

Figure 3: High-Level Overview of the VR Simulator

3.1 VR UI

The VR UI components are key in providing a realistic training experience that resembles the real operating conditions. However, given the immersive capabilities and the space a VR environment provides, there is a possibility to add new features (in a clock-wise order from Fig. 1) to improve the simulator functionality. Some of these features are:

1. Camera Displays: The operators can either prefer to turn around or look at the displays in front to steer the ROV. The "Render Texture" function is used to display several views on a selected geometry.

2. Battery Indicator: It gives a visual representation to the operator how much power is left in the ROV. A simple ROS subscriber may be scripted to read the battery percentage from the real ROV.

3. Depth and Pressure Monitor: It is a display to remind the operator of the depth and the pressure the ROV is experiencing in the water to prevent any issues. These measurements can be shown either using data from Crest and Unity or by using another ROS subscriber script.

4. Virtual Joystick-button based Combination: This type of an UI is more intuitive than a roller and keyboard-mice interface commonly used. There are 3 buttons integrated inside to switch between different modes, they are "Rise", "Move" and "Rotate button. When the joystick is toggled, the degree of joystick rotation is calculated, and the appropriate forces are applied on to the ROV to move it.

5. Thruster Information Screen: This panel provides the information of the thursters being activated in an ROV, due to the registered movement from the joystick.

6. Mini ROV: It has a simple ROV-like design from which the operators can quickly discern the orientation of the actual ROV and proceed accordingly. The mini ROV display was created using Unity's primitive models and features.

7. 3D Artificial Horizon: Since many ROV operators maybe familiar with this device, a 3D adaption of this sensor is developed.

8. ROS-Unity Integration: This is an indicator of connectivity between ROS and Unity.

A physical joystick is also integrated inside this simulation on top of the virtual version. This is to ensure improved accessibility amongst a variety of operators. The joystick used is Logitech Attack 3 [11] (Fig. 4).

Apart from surging (moving forward and back) and swaying (moving left and right) using the joystick, the various buttons on the joystick are also used to provide more control the ROV, these are:

1. "Right": For rotations on the right, i.e., yawing in the clockwise direction.

2. "Left": For rotations on the left, i.e., yawing in the anti-clockwise rotation.

3. "Float": For moving above, i.e., heaving up.

4. "Sink": For moving below, i.e., heaving down

5. "Stop": To stop all motion

Figure 4: Physical Joystick Controls

Currently, using the ROS-Unity integration package, the readings from the joystick can be published to a ROS node which will ultimately be used to control individual thrusters.

A cockpit-like interface may not be the only way of interaction. Certain UI elements maybe placed on the hand which can perform a plethora of operations. An example of this is the "full-stop" button implemented below the palm for an operator to quickly quit the application if there is any discomfort. Fig. 5 is an example of the button's implementation.

Figure 5: Illustration of Hand-based UI

Hand-based UI may incorporate several important functions, which can further reduce cognitive load by reducing the number of features on the cockpit itself.

3.2 Observer ROV

FilmBox (FBX) and Unified Robot Description Format (URDF) versions of a BlueROV, a popular ROV from Blue Robotics is used in this simulator. Fig. 6 shows the BlueROV.

The FBX file format is used for animation and file exchange. However, with the issues in the geometry, it was not possible to manipulate the ROV using the thrusters, it was instead treated as a single object.

URDF represents a robot in the Extensible Markup Language format. It defines kinematic and dynamic attributes as well as the visual meshes of the robot, i.e. a robot's physical description to ROS. The URDF file of the ROV provides with greater control, especially with the thrusters. Hence, the ROV can now be moved in a more realistic fashion, i.e. by the aid of thrusters.

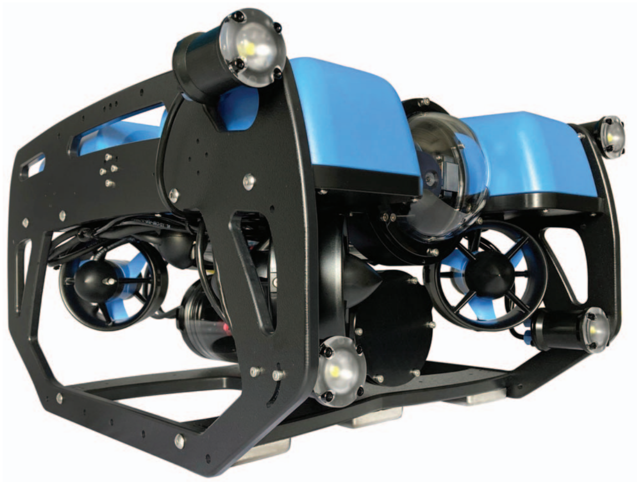

Figure 6: The BlueROV [8]

Only certain propellers will be rotated to move the ROV. Therefore, it is crucial to know which propellers will be rotating in the specific maneuver. When the ROV is moving forward or back (surging), only propellers 1, 2, 3 and 4 will be moving. Fig. 7 shows how the forward motion is possible. The red arrow will cancel each other out leaving the blue arrows. The thruster information screen will light up the respective propellers that are rotating.

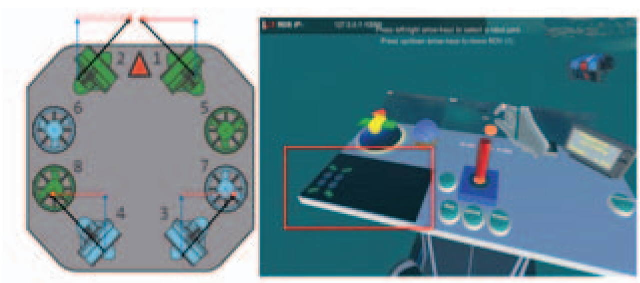

Figure 7: Surge/Forward Movement Illustration [4]

When the ROV is moving upward or downwards (heaving), a different set of propellers will be used. Only the propellers 5, 6, 7 and 8 will be rotating for this motion. Fig. 8 shows how the motion of the respective propellers if the ROV is moving upwards, and the rotating propellers will light up in the thruster information screen.

Figure 8: Heave Movement Illustration [4]

Upon established communication between ROS and Unity, the next step is to publish necessary data using ROS. Like the BlueROV, the thrusters used in this ROV are the T-200 thrusters [17]. Performance data can be obtained from the manufacturer so the necessary current and voltage for the thrusters can be published to a topic when the joystick is toggled. The data published can be subscribed by the real ROV to control the actual thrusters.

3.3 Cleaning ROV and Ship Hulls

A generic ship having a V-shaped hull is used in the simulator, as well as a custom-made ship with a flat-bottom hull. There is no consensus on the amount of cleanliness to call a ship hull to be "clean" or a fixed amount of bio-fouling a single pass of cleaning using an ROV can remove, apart from visual change. Hence, a short script is written for the bio-fouling to be "removed" when a cleaning ROV moves across bio-fouling. Fig. 9 is the isometric of the cleaning ROV, which is a simplified version of the an ROV used by a hull cleaning company [9]. Fig. 10 is a view of the cleaning ROV on a flat-bottom hull and Fig. 11 is a view of the cleaning ROV on a V-shaped hull.

Figure 9: Cleaning ROV Model

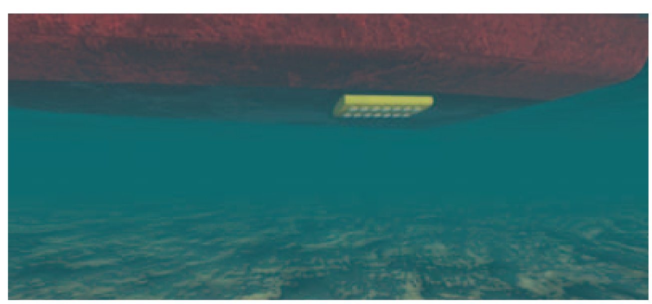

Figure 10: Cleaning ROV on a Flat Hull

3.4 Environment

A core advantage of any simulator is the ability to depict different operating conditions with ease. An example of this is the function to change the time of day by manipulating the brightness and "skybox" conditions in a scene by the touch of a button (Fig. 12 and Fig. 13).

Figure 11: Cleaning ROV on a V-shaped Hull

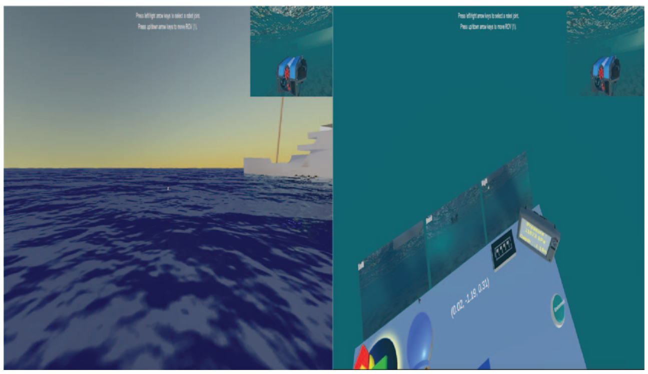

Figure 12: Simulator Environment during Daytime

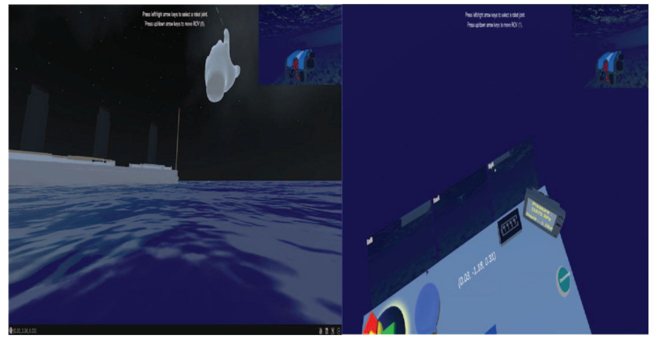

Figure 13: Simulator Environment during Nighttime

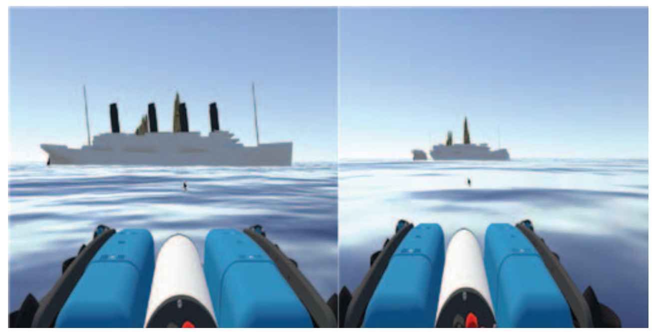

Figure 14: Illustration of Different LODs: High LOD (Left) and Low LOD (Right)

The terrain building-system in Unity helps in creating realistic terrain in the scene. It becomes more useful during depth perception as it acts a reference for the operator's eyes if there is no other object around. A terrain also needs a terrain collider for objects to interact with it. This may be used for ROV to interact with when it sinks to the bottom.

To improve the realism of the simulator, 2 different variants of fish are used. A Fish Flock Controller script is used for random fish generation and movement. This allows them to have a continual movement in the sea. However, due to noise and frequent movement of ships, it is unlikely to find many fish nearby.

Mesh tessellation is standard in game engines like Unity. The smaller the individual elements are, the closer it is to the actual mesh of an object. Level of Detail (LOD) reduces processing load by rendering objects with various details at different distances. Each LOD level exists separately and is activated based on its distance from the viewing camera. The farther the camera is, the lesser the requirement is for high LOD. For the ship in the scene, four LODs were introduced. The original version was imported to a 3D modelling software to decrease the number of faces of the model. Four different versions with various LODs were created and imported back to Unity. Different game objects with different LODs are grouped together, and component LOD group is added to set the distance required to render different LODs. (Fig. 14).

3.5 ROV Sensors

Digital twins of two popular sensors in any underwater ROV, a SONAR and an Inertial Measurement Unit (IMU) are created. Taking advantage of the ROS-Unity integration, we can have the sensor information available with the simulator cockpit that will be convenient for the trainees.

An IMU's digital twin has to manipulate the 3D position of a given object. An IMU usually provides noisy readings and is prone to drift. The current simulator is based on Xsens MTi 100 [14] which is an advanced IMU and the noise addition scripts is a C# implementation of the Aceinna GNSS-SIM python scripts [3]. A simple ROS subscriber could have been used via python scripts to directly change the transforms. However, this simulator uses the C# implementation taking the advantage of Unity with faster update rates. Preliminary experimental validation showed the similarity of the simulated and real IMU readings, shown by the overlap of the probability distributions of the acceleration from both cases (Fig. 15).

A SONAR is a complex device, given its many physics concepts with regards to reflection, transmission losses etc. Based on the Oculus multi-beam echosounder [7] for imaging and the Blue Robotics Ping SONAR from, a single beam echo-sounder [16] our simulator is implemented using the ray-tracing method [23, 28]. However, the limiting factor is the number of rays which is ultimately decided by the hardware capabilities of the device being used. Fig. 16 shows the use of ray-based SONAR simulation as observed in the secondary simulator.

4 Conclusions

This study proves the possibility of integrating VR and robotics technology for simulated training of underwater ship hull cleaning applications. This work emphasizes on the combination of Unity and ROS for underwater dual ROV collaboration. A complete simulator/control package is under development. Initial validation is done with the simulation system. More experiments and optimization will be carried out as part of future work.

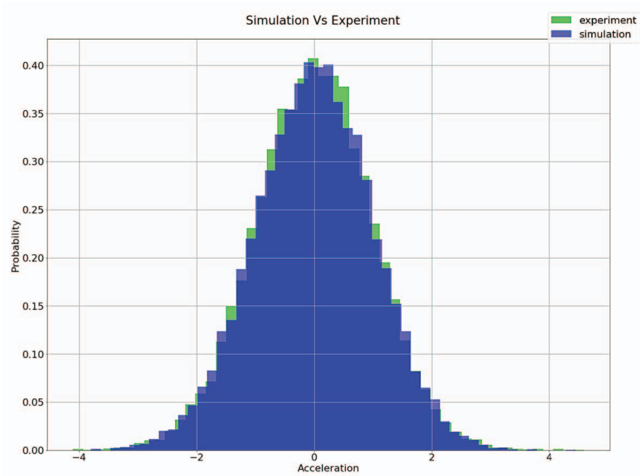

Figure 15: Normalized Probability Distribution of Real and Simulated IMU data

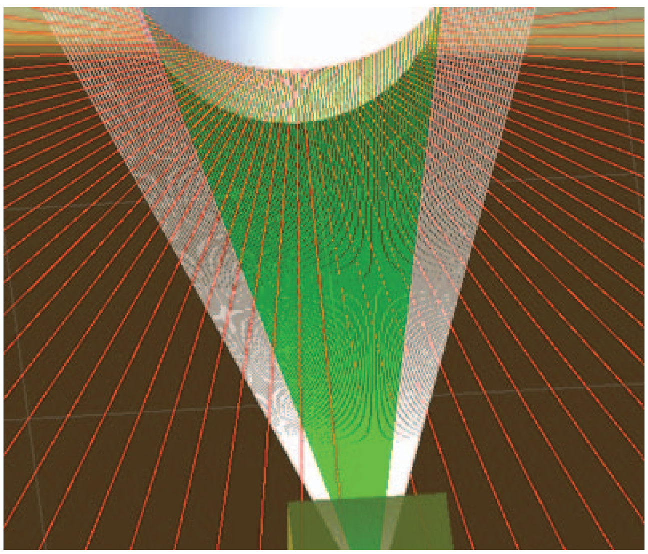

Figure 16: Ray-based SONAR Simulation

References

[1] 1. Introduction - Crest Ocean System. (Date Last Accessed 01-June 2022).

[2] 7 Reasons to use an underwater ROV – Blue skies Drones. (Date Last Accessed 01-June 2022).

[3] Aceinna/gnss-ins-sim: Open-source GNSS + inertial navigation, sensor fusion simulator. Motion trajectory generator, sensor models, and navigation. (Date Last Accessed 01-June 2022).

[4] Ardusub Features. (Date Last Accessed 01-June 2022).

[5] AR/VR technology connects Buyers with Homes in the Real estate market. (Date Last Accessed 01-June 2022).

[6] Australia - New mandatory Biofouling Management Requirements for international Vessels. (Date Last Accessed 01-June 2022).

[7] Blueprint Subsea — Oculus M-series. (Date Last Accessed 01-June 2022).

[8] BlueROV2 - Affordable and Capable Underwater ROV. (Date Last Accessed 01-June 2022).

[9] Hull Cleaning — C-LEANSHIP. (Date Last Accessed 01-June 2022).

[10] K-ROV to use hull-cleaning technology at Port of Townsville - Ship Technology. (Date Last Accessed 01-June 2022).

[11] Logitech® Attack™ 3 Joystick. (Date Last Accessed 01-June 2022).

[12] The Metaverse is already here: 5 companies building our virtual reality future - CBS news. (Date Last Accessed 01-June 2022).

[13] MRTK-Unity Developer Documentation - mixed Reality toolkit — Microsoft Docs. (Date Last Accessed 01-June 2022).

[14] MTi-100 IMU. (Date Last Accessed 01-June 2022).

[15] The Oculus Rift, oculus touch, and Vr Games at E3. (Date Last Accessed 01-June 2022).

[16] Ping sonar Echosounder for underwater Distance Measurement. (Date Last Accessed 01-June 2022).

[17] T200 Thruster for ROVs, AUVs, and marine robotics. (Date Last Accessed 01-June 2022).

[18] Tracking — Leap Motion Controller — ultraleap. (Date Last Accessed 01-June 2022).

[19] Unity Announces support for ROS 2. (Date Last Accessed 01-June 2022).

[20] Virtual reality (VR) - statistics facts — Statista. (Date Last Accessed 01-June 2022).

[21] Virtual Reality (vR) in the Automotive Industry — Overview. (Date Last Accessed 01-June 2022).

[22] VR Applications: 23 Industries using Virtual Reality. (Date Last Accessed 01-June 2022).

[23] W. S. Choi, D. R. Olson, D. Davis, M. Zhang, A. Racson, B. Bingham, M. McCarrin, C. Vogt, and J. Herman. Physics-based Modelling and simulation of Multibeam Echosounder Perception for Autonomous Underwater Manipulation. *Frontiers in Robotics and AI*, 8:279, 9 2021. doi: 10.3389/FROBT.2021.706646/BIBTEX

[24] M. de la Cruz, G. Casañ, P. Sanz, and R. Marín. Preliminary work on a Virtual reality Interface for the Guidance of Underwater Robots. *Robotics 2020, Vol. 9, Page 81*, 9:81, 10 2020. doi: 10.3390/ROBOTICS9040081

[25] A. Elor, T. Thang, B. P. Hughes, A. Crosby, A. Phung, E. Gonzalez, K. Katija, S. H. Haddock, B. E. Erwin, E. J. Martin, and L. Takayama. Catching jellies in immersive virtual reality: A comparative teleoperation study of ROVs in underwater capture tasks. *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST*, 12 2021. doi: 10.1145/3489849.3489861

[26] J. C. García, B. Patrão, L. Almeida, J. Pérez, P. Menezes, J. Dias, and P. J. Sanz. A Natural interface for Remote Operation of underwater Robots. *IEEE Computer Graphics and Applications*, 37:34–43, 1 2017. doi: 10.1109/MCG.2015.118

[27] O. M. Group. Why Do ships Need Hull Cleaning? Offshore Technology, 3 2021.

[28] C. Morency, D. J. Stilwell, and S. Hess. Development of a simulation environment for evaluation of a forward looking sonar system for small AUVs. *OCEANS 2019 MTS/IEEE Seattle, OCEANS 2019*, 10 2019. doi: 10.23919/OCEANS40490.2019.8962650

[29] C. Song and W. Cui. Review of Underwater Ship Hull Cleaning Technologies. *Journal of Marine Science and Application*, 19:415–429, 9 2020. doi: 10.1007/S11804-020-00157-Z/FIGURES/15