Towards Design of an Autonomous Navigation Framework for Unmanned Surface Vessels using Marine Robotics Unity Simulator

SANGHAR, MS, TEMILOLORUN, A., KARA, F. and SINGH, Yogang

Citation:

SANGHAR, MS, TEMILOLORUN, A., KARA, F. and SINGH, Yogang (2024). Towards Design of an Autonomous Navigation Framework for Unmanned Surface Vessels using Marine Robotics Unity Simulator. In: International Ship Control Systems Symposium (iSCSS), Liverpool, UK, 05-07 Nov 2024. [Conference or Workshop Item]

Towards Design of an Autonomous Navigation Framework for Unmanned Surface Vessels using Marine Robotics Unity Simulator

Mohammed Shahid Umar Sanghar [a], Temilolorun Aiyelari[a], Fuat Kara [a], Yogang Singh [a]*

[a]*Department of Engineering and Mathematics, Sheffield Hallam University, Sheffield, S1 1WB, United Kingdom*
*Corresponding author. Email: y.singh@shu.ac.uk

Synopsis

The rising importance of Unmanned Surface Vessels (USVs) in diverse maritime contexts, such as coastal monitoring and oceanographic applications, underscores the necessity for a robust simulation environment framework. This need arises from the limited availability of sea trials and demands a high-quality simulation setup to test USV autonomy across multiple scenarios with enhanced visual fidelity. This paper describes an autonomous navigation framework for USV in a MArine Robotics Unity Simulator (MARUS), a high-fidelity simulation environment based on Unity3D to close the gap. Our approach works on coupling the MARUS simulator offering water physics and a wide array of sensors commonly used in the maritime domain with a Robot Operating System (ROS) to facilitate remotely controlled USV behaviour and motion for different navigation scenarios to assess the vessel's maneuverability and overall performance characteristics.

Keywords: Unmanned Surface Vehicle, Robot Operating System, Marine Robotics Unity Simulator, Autonomous Navigation, Unity3D

1 Introduction

Simulations, utilizing physical equations and computer algorithms, offer a means to analytically solve complex real-world problems safely and efficiently. These simulations speed up the engineering design cycle, cut costs, ensure safe and controlled testing conditions, produce extensive training data for machine learning, aid in developing intelligent robots, and allow for the supervision of real robots via digital twins.

In the marine environment, the advantages of using simulators for developing vehicle control algorithms are even more pronounced due to the high costs associated with testing vehicles at sea. The complexity and logistical challenges of conducting trials at sea necessitate high-quality simulation tools. Existing simulators, however, often fall short in providing the necessary visual fidelity and support for human-robot interaction. Gazebo, one of the most widely used general-purpose robotics simulators, offers various perception sensors through modular plugins (Koenig and Howard, 2004). Despite its extensive use in the robotics community, Gazebo lacks the realistic visual data support required for underwater and surface robotics applications (DeMarco et al., 2015). This gap is significant as the current trend in computer vision research emphasizes the need for visually realistic simulation environments.

Among the specialized simulators for marine robots, the UUV (Unmanned Underwater Vehicle) Simulator and UWSim (UnderWater Simulator) are notable (Cook et al., 2014). The UUV Simulator, integrated with the ROS framework, provides robust models for underwater vehicle dynamics and sensor simulation. However, it has not been actively maintained, resulting in numerous unresolved issues. Similarly, UWSim, while offering underwater simulation capabilities, does not fully meet the needs for high-fidelity visual data. Recently, several marine simulators like NetMarSyS (Garg et al., 2020), Simu2VITA (de Cerqueira Gava et al., 2022), VR-based simulator (Xu et al., 2021) along with a few simulation environments based on Unity3D (Katara et al., 2019; Osa and Orukpe, 2021; Szleg et al., 2022) have been proposed by research groups worldwide. However, these simulations have certain limitations, such as insufficient documentation, lack of maintenance activity, limited access to a comprehensive asset store with easy-to-interface plugins, and the absence of cross-platform deployment capabilities.

With an emphasis on such requirements in mind, this study adopts MARUS, an open-source marine simulator developed by Laboratory for Underwater Systems and Technologies (Lončar et al., 2022) based on modular

---

Authors' Biographies

Mohammed Shahid Umar Sanghar currently is a Research Assistant in the Department of Engineering and Mathematics at Sheffield Hallam University working on the development of simulation frameworks for autonomous marine systems. His research interests include Automation, Control and Operations of Autonomous Systems.

Aiyelari Temilolorun currently is a Research Assistant in the Department of Engineering and Mathematics at Sheffield Hallam University working on the design and development of autonomous marine robots. Aiyelari Temilolorun's research interests include Marine Robotics, Control of Autonomous Systems and Robotics.

Fuat Kara is an Associate Professor of Mechanical Engineering (Ocean Engineering) at the Department of Engineering and Mathematics, Sheffield Hallam University. Dr.Fuat's research interests include Naval Architecture, Mechanical Engineering, Ocean Engineering and Marine Hydrodynamics.

Yogang Singh is a Lecturer (Assistant Professor) in Automation, Control and Systems Engineering at the Department of Engineering and Mathematics, Sheffield Hallam University. Dr. Singh's research interests include Autonomous Shipping and Autonomous Marine Systems with a specific interest in USVs.

software framework - ROS and a cross-platform real-time game engine - Unity3D to perform the virtual captive model tests, namely the Turning Circle test on a USV platform validated by experiments. MARUS provides a set of significant features important for such testing such as accurate lighting simulation, simplicity of creating surrounding scenery, and ocean simulation with dynamic fluid-structure interaction. The current work extends the existing capabilities of MARUS by developing a new captive test module and a new control module to perform Turning Circle tests in the MARUS simulation environment validated by experiments. The paper is organized as follows: Section 2 outlines the specifications of the captive model test and discusses the performance metrics used to evaluate the USV. Section 3 elaborates on the MARUS architecture and presents simulation results for the captive tests conducted on the USV, which are detailed in Section 4. The manuscript concludes with a discussion of the results and potential future developments for the simulation framework.

2 Captive Model Tests

International Maritime Organisation (IMO) standards for captive model tests serve as a crucial step in validating the performance of marine vehicles to make a controlled assessment of their seakeeping and maneuvering capabilities (Kim et al., 1996). Captive model tests offer a cost-effective and practical means of identifying potential issues before full-scale deployment of USV, thereby saving time and resources while enhancing the overall reliability and efficiency of the USV. There are various methods for performing captive model tests, including Computational Fluid Dynamics (CFD), Planar Motion Mechanism (PMM) tests, and open-water free-running tests, to evaluate the maneuvering capabilities of USVs in different operating environments (BAKAR, 2015). In the current study, the maneuvering capabilities of the USV are assessed by a known captive model test, namely, the Turning Circle test through a set of open-water free-running tests conducted in the simulation environment of MARUS. The Turning Circle test evaluates the ship's ability to turn within a specific radius, responsiveness and stability during course changes ensuring that the vessel meets performance metrics as shown in Figure 1.

2.1 Performance Metrics

The virtual Turning Circle test are performed in the current study on the simulation environment of MARUS to determine the following performance metrics:

- Tactical diameter
- Advance
- Transfer
- Turning Radius
- Time to change heading 90° and 180°

2.2 IMO Criteria for Virtual Free Running Tests

To perform the virtual free-running maneuvering experiments in the open water simulation environment of MARUS, International Maritime Organisation (IMO) Resolution A.751 stating interim standards for ship maneuverability(Kim et al., 1996; BAKAR, 2015) was adopted with suggested modifications for differential drive propulsion vessels as follows:

1. The approach speed should be 90% of the vessel speed corresponding to 85% of the maximum engine output.
2. Before executing the maneuver, the vessel must maintain a steady course at a constant setting for a minimum of 10 seconds.

To test the performance of the USV in the simulated environment, the USV maneuverability in terms of turning circle parameters was compared against the experimental values from the test conducted in the (Aiyelari Temilolorun, 2024) whose details are explained in the separate manuscript presented in this conference.

3 MARUS Architecture and Capabilities

3.1 MARUS Architecture

The advanced simulation framework of MARUS uses gRPC networking technology, an open-source remote procedure call (RPC) framework that uses HTTP/2 for transport, Protocol Buffers as the interface description language, and provides features such as authentication, load balancing, and more, for bi-directional communication between Unity3D and ROS system. The cross-platform compatibility of Protocol Buffers has been utilised in the

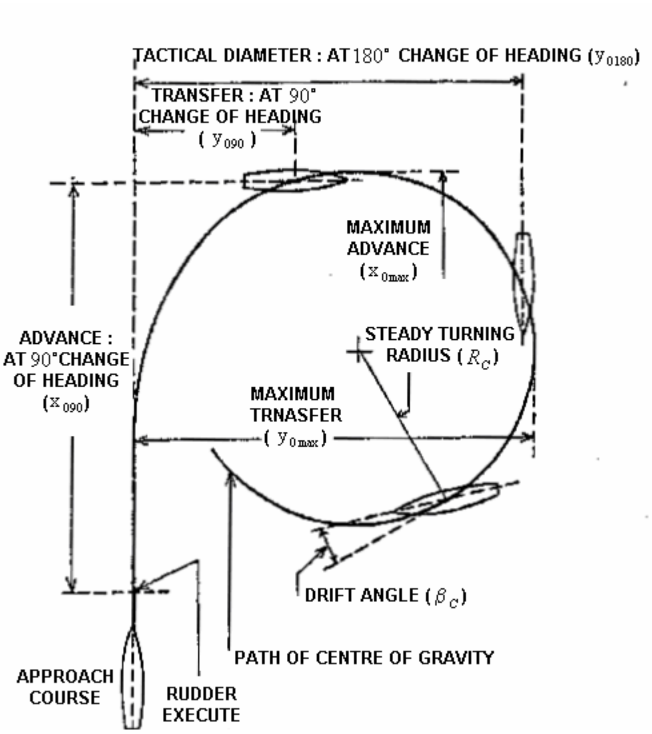

Figure 1: Schematic of Turning Circle Test with performance metrics

current work to define services and messages, due to its language-neutral, platform-neutral, extensible mechanism for serializing structured data. MARUS leverages Unity's powerful real-time 3D rendering capabilities and physics engine to create realistic marine environments through the uncoupled incorporation of water physics (buoyancy, currents, waves, etc.), sensors, robotic tools and actuators into the application backend of controllers and situational awareness. MARUS architecture with ROS backend is shown in Figure 2 where MARUS receives inputs for actuators, mission control variables, and simulation environment requests from ROS. The current study enhances the MARUS architecture by integrating a captive test module, which includes the Turning Circle behaviours, into the MARUS environment. Additionally, it incorporates a path-following control script to execute these tests, as depicted in red in Figure 2.

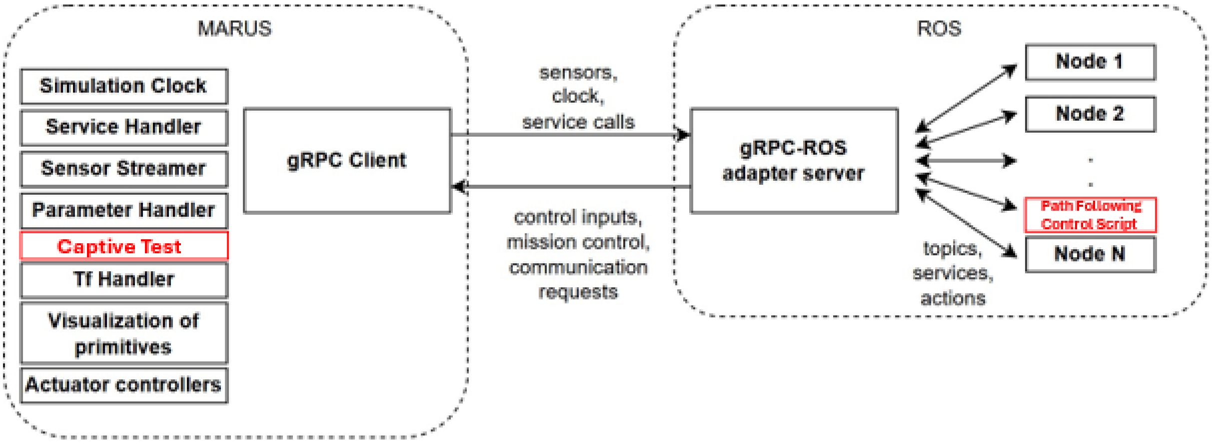

Figure 2: MARUS architecture with ROS backend (Modified from Lončar et al. (2022)) with red elements showing the new incorporated modules

3.2 MARUS Capabilities

MARUS uses Unity, a game engine for creating robotic test scenes due to its widespread popularity for offering rendering, lighting and physics solutions in 3D virtual environments. In addition to that, several virtual sensors have been integrated for the simulation of marine robots such as IMU, GNSS, Camera, 3D Lidar, Sonar and several others required for performing critical virtual simulations. Additionally, for USV actuation, multiple controllers namely, Keyboard controller, Velocity PID controller, Go To Point controller, Open Loop controller and Thruster have been developed. The vessel's physics are approximated, considering buoyancy and hydrodynamic forces in the virtual 3D scene set up in the ocean environment as shown in Figure 3 where virtual Turning Circle tests are performed.

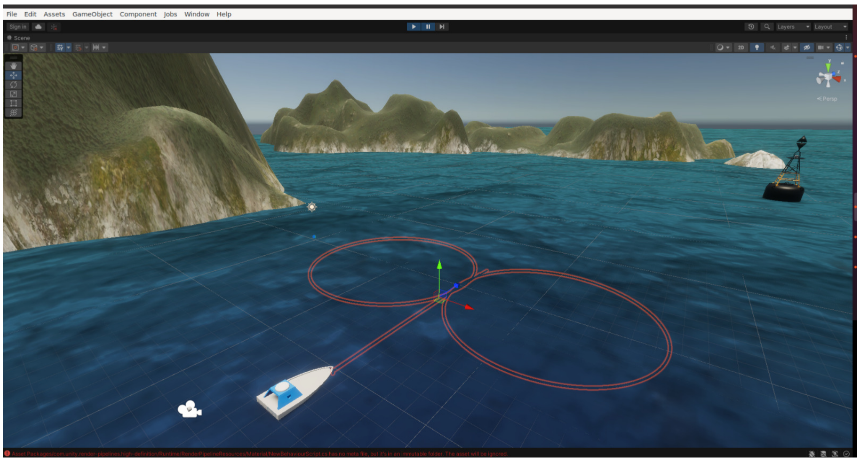

Figure 3: MARUS 3D simulation environment developed using Unity showing turning circle path in red for port and starboard side

In this study, we used Ubuntu 20.04 for running Unity3D and ROS simulations, benefiting from the stability and compatibility of the operating system. The computational demands primarily arise from Unity3D's real-time rendering and physics simulations, which require a multi-core processor, at least 16 GB of RAM, and a dedicated GPU (e.g., Nvidia GTX 960 or higher). ROS handles communication between nodes and real-time sensor data, scaling efficiently across multiple cores. The system is scalable, with ROS distributing workloads and Unity3D offering optimization options to manage graphical and physics performance. For this experiment, we utilized Ubuntu 20.04 running on a system equipped with an Nvidia GTX 960M GPU, 16GB of RAM, and 500GB of storage. Ubuntu 20.04 was chosen for its stability and compatibility, as it is the last release to officially support both ROS1 and ROS2, which are essential for setting up the docker installation of MARUS. This configuration ensured smooth integration of both ROS versions, which was a critical requirement for the simulation environment.

3.3 Thruster Controller

The MARUS architecture has several in-built controllers for the USV actuation as follows:

- Keyboard controller- keyboard-based speed generator.
- Velocity PID controller - uses position reference to generate PID parameters for velocity control.
- Go To Point controller - drives USV to a given reference point using primitive control.
- Open Loop controller - time-dependent control input.
- Thruster controller- generates force based on the datasheet available in ROS or MARUS.

The current study adopts the Thruster controller due to its ability to generate force as required for the differential drive propulsion to execute the IMO criteria for free-running turning circle tests stated in section 2.2. The current study adopts the datasheet of the T200-V16 thruster installed on the USV in the simulation framework with a maximum forward thrust of 14.65kgf and a maximum reverse thrust of 11.65kgf being utilised. This also aligns closely with the experimental set-up to validate the simulation results. This datasheet can be accessed from the docker using *Assets > marus-core > Datasheets > T200*. The schematic of the arrangements of the thrusters on the USV in the current simulation framework is shown in Figure 4.

4 Simulation Results

In the current study, the virtual turning circle test was conducted in a simulation framework for both the port and starboard sides by controlling the USV's thrusters, and the USV's path was recorded using a virtual Inertial Measurement Unit (IMU) embedded within the MARUS framework. The starboard results were validated against

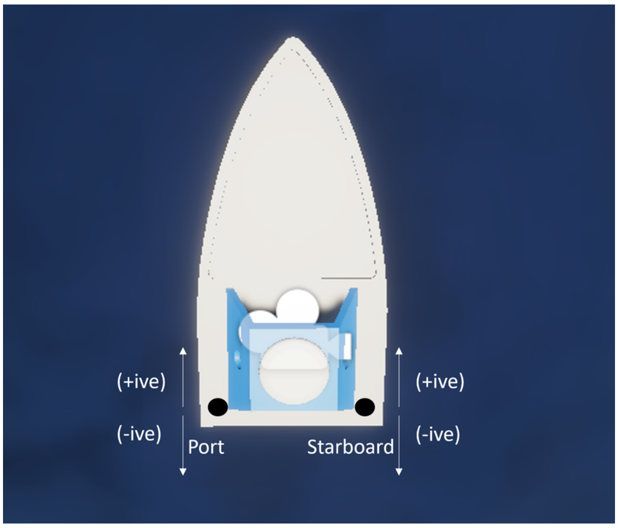

Figure 4: Schematic of the arrangement of thrusters on the USV in the current simulation framework. The positive and negative labels on the port and starboard side represent the directions of the forward and reverse thrust for the USV actuation.

the experimental results from the test conducted in the Aiyelari Temilolorun (2024). Figure 5 presents the simulation results for the port and starboard sides during the virtual Turning Circle test, alongside a comparison with the experimental results for the starboard side. Table 1 presents the performance metrics for both the port and starboard sides during the virtual Turning Circle test and the experimental results for comparison.

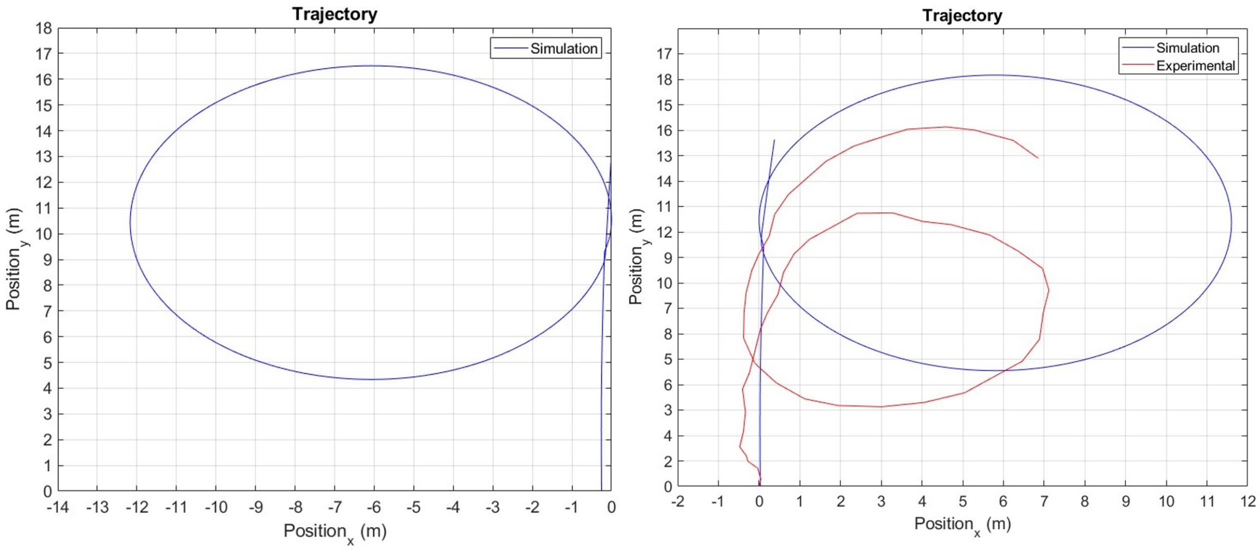

Figure 5: Simulation Turning Circle test for the port (left image) and starboard side (right image)

Parameters	Simulation(Port)	Simulation(Starboard)	Experimental(Starboard)
Advance	7.15 m	7.1 m	8.42 m
Transfer	6.11 m	5.9 m	3.8 m
Tactical Diameter	12.2 m	11.3 m	7.07 m
Time to change heading 90°	62.5 s	60.5 s	45 s
Time to change heading 180°	85 s	79 s	56 s

Table 1: Comparison of performance metrics for the virtual Turning Circle test and experimental results

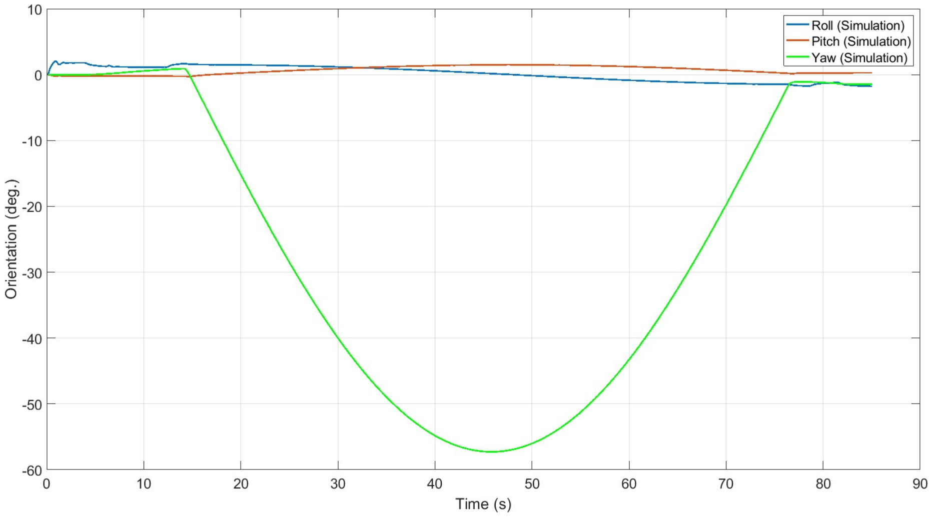

Figure 6: 3D orientation of the vessel in virtual Turning Circle test for the port side

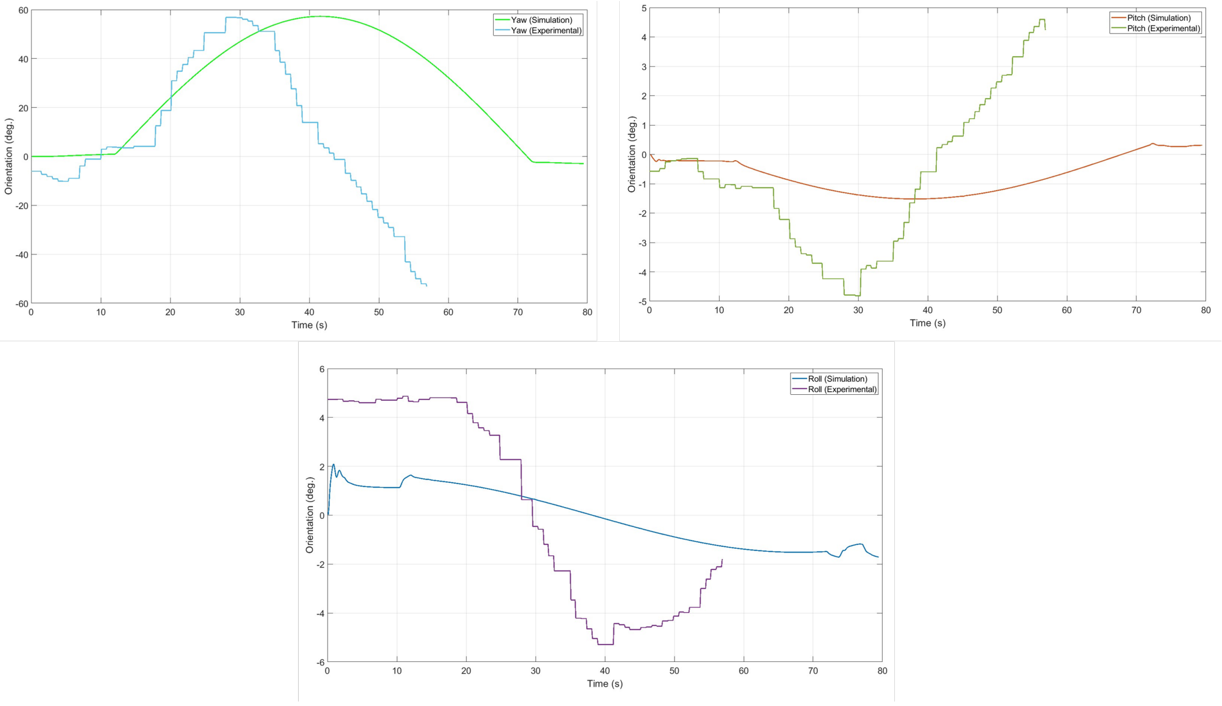

Figure 7: 3D orientation of the vessel in virtual Turning Circle test for the starboard side with comparison against the experimental values

From table 1 it is evident that manoeuvring parameters in both simulation conditions are quite similar, indicating consistency in the model's behaviour for turns in either direction. The starboard turn has a slightly lower value with a larger discrepancy from the experimental one. This discrepancy indicates that in real-life conditions vessel completes the turn in a significantly smaller area than predicted by simulations. This discrepancy arises from the fact that real-time conditions like wind and surface currents may not be accurately modelled in simulations with simulation not fully capturing all the dynamics of the vessel, such as hull interactions with water and real-time adjustments by the humans for remotely controlled vessels. Figure 6 shows the orientation of the vessel during a simulated Turning Circle test to the port side where roll and yaw remain relatively stable for the duration of the turn and the pitch relatively aligns with the behaviour of the dynamics of turning circle test. Similarly, Figure 7 depicts the 3D orientation of a vessel during a simulated and experimental turning circle test to the starboard side. The simulation predicts minor roll oscillations, while the experimental results show significant roll, with the vessel tilting much more from side to side. This suggests that the real-life vessel experiences greater lateral instability during a starboard turn than the simulation predicts. The simulation shows a significant bow-up pitch, reaching 55 degrees, whereas the experimental results show minimal change in pitch, staying around 5 degrees. This discrepancy suggests that the simulation may overestimate the vessel's pitching behaviour during a starboard turn. Both simulation and experimental results show a similar yaw pattern, with a steady counterclockwise rotation. The experimental yaw is slightly more stable, indicating that the vessel's heading change in real-time is consistent but less variable than predicted by the simulation. The simulation appears to predict a more controlled and less dynamic turning behaviour than observed in real-life conditions. This could be due to simplifications in the simulation model or unaccounted factors in the real environment. The simulation for the port side can be accessed from https://www.youtube.com/watch?v=wJ05OrLqu$_M$ while the starboard side can be accessed from https://youtu.be/yFIsHnBdhZA.

5 Conclusions

In this study, a virtual free-running Turning circle test was performed using a MARUS simulation framework to evaluate the manoeuvering performance of a USV platform validated by experiments. The simulations provided a useful approximation of the parameters with a comparison against the experimental results providing a crucial understanding concerning the vessel's true manoeuvring capabilities. The differences highlight the need for continual refinement of simulation models to better match experimental results. The factors such as surface current, winds and hull design will be considered for the redesign of the simulation framework to reiterate on simulation results for a closer match with experimental data. A first step in this regard is to implement a non-linear feedback control strategy within MARUS to better capture the complex dynamics of vessel maneuvers during the Turning Circle test and expand the scope to more difficult manoeuvring scenarios.

Acknowledgement

This research is supported by the Sheffield Hallam Department of Engineering & Mathematics Research & Innovation Funding for the project titled "*Towards design and development of a Marine Robotics Unity Simulator (MARUS) using Artificial Intelligence (AI) tools*". For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising from this submission.

References

Aiyelari Temilolorun, Y.S., 2024. Development of a low cost unmanned surface vessel for autonomous navigation in shallow water, in: iSCSS 2024, Liverpool, Institute of Marine Engineering Science and Technology.

BAKAR, M.H.A.B.A., 2015. Manoeuvering analysis in free running scaled rc model on 60m multi purpose offshore support vessel (mposv).

de Cerqueira Gava, P.D., Nascimento Júnior, C.L., Belchior de França Silva, J.R., Adabo, G.J., 2022. Simu2vita: A general purpose underwater vehicle simulator. Sensors 22, 3255.

Cook, D., Vardy, A., Lewis, R., 2014. A survey of auv and robot simulators for multi-vehicle operations, in: 2014 IEEE/OES Autonomous Underwater Vehicles (AUV), IEEE. pp. 1–8.

DeMarco, K.J., West, M.E., Howard, A.M., 2015. A computationally-efficient 2d imaging sonar model for underwater robotics simulations in gazebo, in: OCEANS 2015-MTS/IEEE Washington, IEEE. pp. 1–7.

Garg, S., Quintas, J., Cruz, J., Pascoal, A.M., 2020. Netmarsys-a tool for the simulation and visualization of distributed autonomous marine robotic systems, in: 2020 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV), IEEE. pp. 1–5.

Katara, P., Khanna, M., Nagar, H., Panaiyappan, A., 2019. Open source simulator for unmanned underwater vehicles using ros and unity3d, in: 2019 IEEE Underwater Technology (UT), IEEE. pp. 1–7.

Kim, M.J., et al., 1996. Assessment of ship manoeuvrability based on imo resolution no. a. 751 .

Koenig, N., Howard, A., 2004. Design and use paradigms for gazebo, an open-source multi-robot simulator, in: 2004 IEEE/RSJ international conference on intelligent robots and systems (IROS)(IEEE Cat. No. 04CH37566), Ieee. pp. 2149–2154.

Lončar, I., Obradović, J., Kraševac, N., Mandić, L., Kvasić, I., Ferreira, F., Slošić, V., Na, ., Mišković, N., 2022. Marus-a marine robotics simulator, in: OCEANS 2022, Hampton Roads, IEEE. pp. 1–7.

Osa, E., Orukpe, P., 2021. Simulation of an underwater environment via unity 3d software, in: BOOK OF PROCEEDINGS, p. 384.

Szleg, P., Barczyk, P., Maruszczak, B., Zieliñski, S., Szymañska, E., 2022. Simulation environment for underwater vehicles testing and training in unity3d, in: International Conference on Intelligent Autonomous Systems, Springer. pp. 844–853.

Xu, F., Zhu, Q., Li, S., Song, Z., Du, J., 2021. Vr-based haptic simulator for subsea robot teleoperations, in: Computing in Civil Engineering 2021, pp. 1024–1032.