

BUILDING A SIMULATION ENVIRONMENT FOR AUTONOMY VERIFICATION AND VALIDATION

Wong Xi Lei Celeste¹, Chew Yihang², Wong Yu Hin²

¹CHIJ St Nicholas Girls' School, 501 Ang Mo Kio Street 13, Singapore 569405

²Defence Science and Technology Agency, 1 Depot Road, Singapore 109679

Abstract

This paper explores the use of simulation environments in the verification and validation (V&V) of autonomous systems through the processes of setting up an environment and navigating how to teleoperate an Unmanned Surface Vehicle (USV) within the environment. The aforementioned simulation environment is the Virtual RobotX (VRX) environment and 3 different approaches to setting up the VRX environment are considered in this paper. The final simulation environment is set up by using Robot Operating System 2 (ROS 2) and Gazebo Sim to create the VRX environment in a Virtual Machine (VM). The Real Time Factor (RTF) of the final simulation environment fluctuates within a range of 7.0-9.0%. With the final environment set up, teleoperation of the USV is configured.

1. INTRODUCTION

Autonomy V&V is the testing of autonomous systems in USVs, whereby verification ensures that the USV has been built correctly, and validation checks that the USV is safe and able to fulfil its purpose. [1] Traditionally, autonomy V&V for USVs is done after the shipcraft has been built and it is then made to travel long distances. This process can only be done later on in the project phase and takes up to a few months to complete. Hence, it is an expensive and highly time consuming process.

Introducing the use of simulation environments to autonomy V&V solves many of the problems that come with traditional testing. Firstly, the use of virtual environments in autonomy V&V eliminates the need for a physical ship to be built for testing. Testing can therefore be conducted earlier on in the project phase, and problems can be detected and rectified sooner. Also, using simulation environments for autonomy V&V allows for accelerated testing to be conducted as simulations can be carried out faster than in real time.

This paper explores the application of simulation environments to autonomy V&V through the process of setting up the VRX simulation environment and teleoperating a USV in the environment. The VRX environment is introduced in Section 2 and in Section 3, various approaches are proposed to set it up. The results of the 3 methods are then discussed in Section 4 and the ability of the USV to teleoperate in the environment is evaluated in Section 5.

2. THE VIRTUAL ROBOTX (VRX) ENVIRONMENT

The VRX environment shown in Fig. 1 is an environment jointly created by Open Robotics and the United States of America (USA) Naval Postgraduate School. It is a Gazebo-based simulation that is designed to support the development, testing and evaluation of USVs operating in ocean environments. [2] The VRX environment is also used in the VRX competition hosted by RoboNation, the USA Office of Naval Research, the USA Naval Postgraduate School and Open Robotics, where participants have to set up the environment and innovate creative solutions to the challenges their USVs face within the environment.



Fig. 1. Picture of the VRX environment.

The VRX environment was chosen as it is open source and Gazebo Simulator, which the VRX environment is based on, is designed to accurately reproduce dynamic environments that robots may be faced with. Therefore, all models in the simulation have mass, velocity, friction and various elements that allow the environment to be more accurate and realistic. [3] When the VRX world is launched, it has numerous models such as islands, buoys and the USV in place. It is also able to support accelerated testing as Gazebo Simulator is able to conduct simulations faster than in real time. Hence making the VRX environment the most suitable simulation environment for this project.

3. METHODS TO SET UP THE ENVIRONMENT

There are 3 methods proposed to set up the VRX environment. The flowchart in Fig. 2 summarises the prerequisites for the individual methods, such as Robot Operating System 2 (ROS 2), and the steps to be taken for each. The methods are the Base Gazebo method, the method of using ROS 2 Humble and Gazebo Garden in a VM, and the method of using ROS 2 Humble and Gazebo Garden in a docker container. Similar to the method of using ROS 2 Humble and Gazebo Garden in a VM, the Base Gazebo method is able to be carried out in a VM.

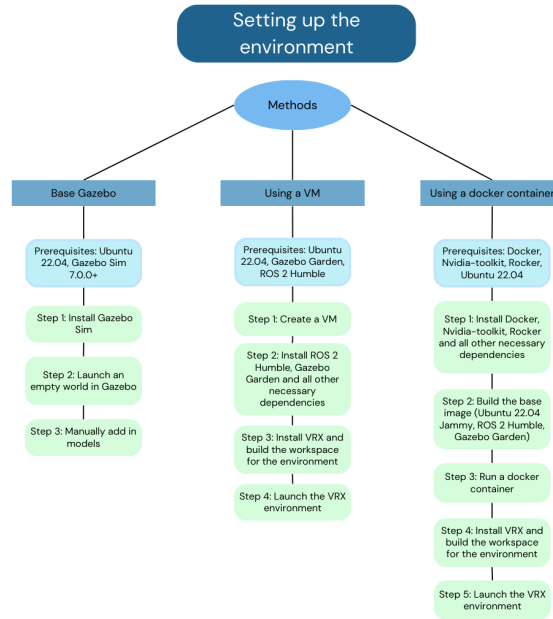


Fig. 2. The prerequisites and steps taken for each method.

3.1: Base Gazebo

The Base Gazebo method makes use of the Gazebo graphical interface that allows for the manipulation of the 3D simulated world. [1] It involves creating an empty world on Gazebo Simulator and manually adding in the models used in the VRX world. This method is the easiest to set up of the three as it has the least prerequisites and steps. In contrast with the other two methods, the Base Gazebo method does not require a specific version of Gazebo and does not require the installation or use of ROS 2. As the models have to be installed, sourced and added to the environment manually, the process can be tedious and time consuming.

3.2: Using ROS 2 Humble and Gazebo Garden in a Virtual Machine (VM)

This approach of setting up the environment is simple and can be done in a relatively short amount of time. As it is considered the default method of participants to set up the VRX environment for the VRX competition, instructions and troubleshooting information is easily accessible. However, the packages and other necessary dependencies have to be manually installed which can be tedious, and this method takes up a sizable amount of memory as VMs run on their own Operating System (OS). [4] It also requires specific versions of Gazebo Simulator and ROS 2.

3.3: Using ROS 2 Humble and Gazebo Garden in a docker container

The docker container method also requires specific versions of Gazebo Simulator and ROS 2, however unlike the previously mentioned method, the packages and dependencies do not have to be manually installed. Unlike VMs, containers run on the

host machine's OS and are therefore more lightweight and take up less memory. [4] The process of installing the docker is also simple, however this method can be complicated as there are both docker managed and third-party managed tools. When using this method, several complications were also faced when setting up the docker container.

4. EVALUATION OF THE RESULTS

Each method created an environment and the results have been compared in Table 1. The individual outcomes are explained in reference to the table in Sections 4.1, 4.2 and 4.3

	Base Gazebo *Can be done in a VM	Using ROS 2 Humble and Gazebo Garden in a VM	Using ROS 2 Humble and Gazebo Garden in a docker container
Operating system (OS)	<ul style="list-style-type: none"> Can run on either the VM or host machine's OS 	<ul style="list-style-type: none"> Runs on the VM's own OS 	<ul style="list-style-type: none"> Runs on the host machine's OS
Environment	<ul style="list-style-type: none"> Empty environment with no models Models have to be manually installed, sourced and added into the environment 	<ul style="list-style-type: none"> Launched with all the VRX environment models in place 	<ul style="list-style-type: none"> Launched with all the VRX environment models in place
Models	<ul style="list-style-type: none"> Models do not have the element of collision Models fall through the ground plane upon starting the simulation 	<ul style="list-style-type: none"> Models have the element of collision 	<ul style="list-style-type: none"> Models have the element of collision
Simulator plugins	<ul style="list-style-type: none"> Plugins are fully functional 	<ul style="list-style-type: none"> Plugins are fully functional 	<ul style="list-style-type: none"> Plugins are fully functional
Real time factor (RTF) *RTF \approx time elapsed in the simulation \div time elapsed in reality	<ul style="list-style-type: none"> Ranges from 38.0-42.0% Highest RTF achieved of the three methods 	<ul style="list-style-type: none"> Ranges from 7.0-9.0% After removing all other models except the USV and the ocean, the RTF maintains a range of 8.0-11.0% 	<ul style="list-style-type: none"> Ranges from 7.0-9.0%

Table 1. Comparison of the environments set up by each method.

4.1: Base Gazebo

The environment set up using the Base Gazebo method is most flexible as the user has control over the models added, however upon starting the simulation, the models would fall through the ground plane as they did not have elements of collision. Thus making the Base Gazebo simulation environment unable to support USV teleoperation. The plugins of the simulation such as the video recording function and world statistics were also fully functional. Despite the environment achieving the highest Real Time Factor (RTF), a RTF of 38.0-42.0% is considerably low as for accelerated testing to be achieved, the RTF of the simulation should be $>100\%$.

4.2: Using ROS 2 Humble and Gazebo Garden in a Virtual Machine (VM)

The environment was launched with all the models such as the islands, ground station and USV in place. Unlike the Base Gazebo method, the models have the element of collision and do not sink through the ground plane upon starting the environment. However, the RTF of the simulation fluctuates within a range that is extremely low at 7.0-9.0% and improves marginally to a range of 8.0-11.0% after removing the elements of light and all other models except the USV

and the ocean. Despite the low RTF, this method is the preferred method of the 3 due to the simplicity of the set up and the results.

4.3: Using ROS 2 Humble and Gazebo Garden in a docker container

The environment launched is similar to the one launched in the previously mentioned method. It has a similarly low RTF and the models have the element of collision. However, as the process of setting up the environment is more complicated than the previously mentioned method, it is not preferred for setting up the VRX environment.

5. TELEOPERATION OF THE UNMANNED SURFACE VEHICLE (USV)

With the final simulation environment set up, teleoperation of the USV was configured. To configure teleoperation, the ROS 2 Humble teleoperation dependencies have to be installed and launched. The process of setting up teleoperation is simple and movement of the USV can be controlled through the joysticks of a gamepad. The USV was able to teleoperate however, due to the low RTF of the simulation, the movement of the USV was extremely delayed. The position of the USV shown in Fig. 3 is the USV's starting point and it is then moved toward the buoy (circled in red). When the USV first made contact with the buoy, Fig. 4 was taken, the photographs were taken 88s apart, showing that the USV takes a significant amount of time to move from point to point.



Fig. 3. The USV at its starting point. The buoy is circled in red.



Fig. 4. The USV in contact with the buoy.

6. FUTURE IMPROVEMENTS

More can be done to explore the Base Gazebo method, especially in regard to the models in the simulation environment not having the element of collision. Due to the more favourable RTF achieved by the Base Gazebo method simulation environment, successfully adding an element of collision to the models would allow for more progress towards achieving the goal of accelerated testing.

Additionally, more work can be done to find a solution to the problem of the simulations having low RTFs. Removing elements of light and other models help to increase the RTF marginally but a solution to increase RTF while maintaining all elements of light and models will allow for a more accurate testing environment to be created.

7. CONCLUSION

The VRX environment has been set up successfully and the 3 methods have been evaluated. Teleoperation of the USV has also been configured, however the ability of the USV to teleoperate and the accuracy of the testing environment is limited by the low RTF of the simulation. The current simulation environment still has many flaws, and in order to achieve accelerated testing and a simulation environment that is able to accurately test the safety and efficacy of USVs, the environment will need to be optimized, for example by increasing the computation step size, and removing unnecessary elements such as models on the landmass, which do not affect USV navigation.

Due to the short duration of the project, there was not enough time to develop a control logic for the navigation autonomy.

REFERENCES

- [1] Araujo, Hugo, Mohammad Reza Mousavi, and Mahsa Varshosaz. 2023. "Testing, Validation, and Verification of Robotic and Autonomous Systems: A Systematic Review." *ACM Transactions on Software Engineering & Methodology* 32 (2): 1–61. doi:10.1145/3542945.

- [2] B. Bingham et al., "Toward Maritime Robotic Simulation in Gazebo," OCEANS 2019 MTS/IEEE SEATTLE, Seattle, WA, USA, 2019, pp. 1-10, doi: 10.23919/OCEANS40490.2019.8962724.
- [3] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), Sendai, Japan, 2004, pp. 2149-2154 vol.3, doi: 10.1109/IROS.2004.1389727.
- [4] Yadav, Anuj Kumar, Madan Lal Garg and Ritika. "Docker Containers Versus Virtual Machine-Based Virtualization." *Advances in Intelligent Systems and Computing* (2018): n. pag.