Volans: a Physics Simulator for Evaluating UUV and USV Autonomy at Scale

Mark Moll
Metron, Inc., Reston, VA, USA, mollm@metsci.com

***Abstract*—We present Volans, a Gazebo-based physics simulator focused on evaluating autonomous behavior of maritime vehicles. The simulator has been used successfully to test our autonomy stack before in-water testing of several vehicles.**

***Index Terms*—Simulation, UUVs, USVs, autonomy, sim-to-real.**

I. Introduction

Metron has developed a physics-based maritime simulation framework called Volans[1] to evaluate and test its autonomy software. Development started more than 13 years ago as a fork of UUV Simulator [1], which is built on top of Gazebo [2]. Over time Volans has significantly diverged from UUV Simulator. Its main features can be categorized as follows:

Environment: Volans uses a geodetic coordinate system and has models for bathymetry, time-varying water currents, and many other environmental information.

Vehicle dynamics Volans includes different dynamics models, where model complexity can be selected based on available in-water data available for fitting and desired speed/accuracy trade-offs. Dynamics include the effects of currents, buoyancy, gravity, and Coriolis effects.

Sensing & perception: A broad range of sensors have been implemented, including Doppler velocity log, GPS, USBL, and an IMU.

Interoperability: Volans has been integrated with various complementary software frameworks using a variety of messaging frameworks.

Below, we will expand on each category.

II. Environmental Modeling

Volans is able to use a broad range of geospatial data in NetCDF format. This includes bathymetry, water currents, surface wind, water temperature, and salinity. All of these (except bathymetry) can be time-varying. Within the NetCDF format this data is stored in an n-dimensional grid. Internally, Volans uses linear or cubic interpolation to compute data values between grid points.

Internally, Volans maintains a geodetic coordinate system and remaps data to a collection of two-dimensional patches (similar to an atlas) using the S2 Geometry Library[2] Bathymetry data that maps to a given patch (or *cell* in S2 Geometry terminology) is (un)loaded on demand as vehicle enter or leave an area. From the bathymetry data for a cell a uniformly interpolated mesh is constructed and inserted into the simulated world.

[1]Volans is named after the constellation of Piscis Volans, "flying fish."
[2]See http://s2geometry.io.

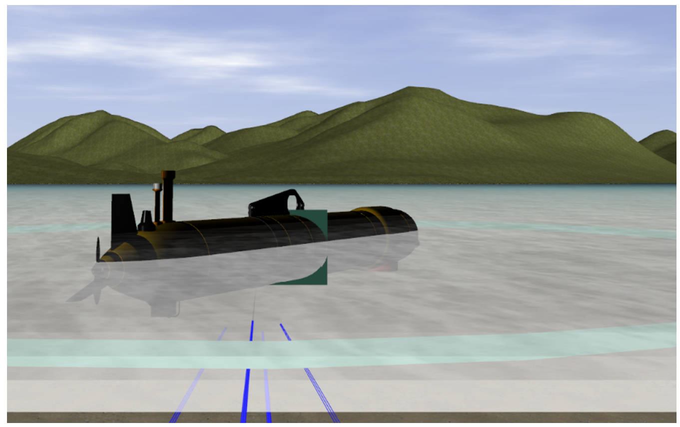

Fig. 1. Screenshot of a simulated vehicle in Volans. The blue rays below the vehicle are a visualization of DVL beams.

III. Vehicle Dynamics

We are focused on simulations to evaluate autonomous vehicle missions that last anywhere from hours to weeks, with vehicles covering distances of up to a few hundred miles. We routinely simulate multiple vehicles with heterogeneous capabilities at faster than real-time speed. The desired spatiotemporal scale of simulation makes detailed simulation of, e.g., fluid dynamics impractical. Instead, we focus on simulation at a resolution that (1) affects autonomous decision making and (2) can be validated via in-water testing. This means that we can take some shortcuts when simulating sensors, controllers, and communication devices. We can control the degree of realism (and accuracy/speed trade-offs) as needed. For example:

- For long transitions a simple controller might be sufficient, while for docking a UUV to a USV a more precise model could be used.
- For an INS or other sensor-dependent capabilities sometimes a "qualitatively correct" approximation is sufficient, while sometimes we may want to use a more detailed model.
- Similarly, for basic communications a simple abstraction might be sufficient, while it is sometimes beneficial to include environmental factors (like salinity and water temperature) that affect speed of communication.

We support three different types of vehicle dynamics, which can be selected per vehicle (as opposed to globally for all simulated vehicles):

1) A detailed Fossen model [3]. While this offers potentially a good fit to many maritime vehicles, it includes over 50 parameters, which can be difficult to fit to collected in-water data for a vehicle of interest.
2) A reduced closed-loop model with “only” about 20 parameters. For many vehicles we have been able to achieve good agreement with in-water data by creating initial estimates for many parameters from manufacturers’ specification sheets followed by a parameter tuning stage to fit parameters to in-water data.
3) Purely scripted motion. To simulate non-reactive traffic in proximity of vehicles of interest, we also have a way to interpolate the states for this traffic from a series of keyframes. Each keyframe contains a waypoint and timestamp.

Volans has been used to simulate primarily underwater vehicles ranging from small vehicles like the REMUS 600 to large vehicles like the Cellula Guardian, a shipping container-sized vehicle with a 5,000 km range. In the last few years, we have expanded into simulation of surface vehicles as well such as the widely-used WAM-V and an experimental surface vehicle developed in-house.

IV. Sensing & Perception

As mentioned in the introduction, there are sensor models for most commonly used sensors for maritime vehicles. On top of these sensors we have implemented different INS models to maintain state estimates. The reason for having different INS models is the same as for different vehicle dynamics models: it allows us to trade off accuracy and speed as desired. All INS models can take in IMU readings, position/velocity fixes, depth fixes and DVL beam fixes and perform state updates using, e.g., an unscented Kalman filter. There is also an abstracted target recognition pipeline that can be used for target identification, target classification, and target tracking. Finally, we also have some introspective sensing capabilities: power consumption is explicitly modeled and includes hotel power, power used by sensors and propulsor power.

V. Interoperability

While the original use case for the simulator was to test out own autonomy, we have since build software bridges for a number of third-party autonomy stacks using different messaging frameworks. The typical structure of such software bridges is to subscribe to Volans topics (which uses the same Protobuf-over-ZeroMQ message infrastructure as Gazebo) and republish them using a third party’s messaging framework such as MOOS [4].

Besides support for autonomy software, we have also integrated Volans with complementary software. For instance, we have developed an extension that enables realistic simulation of acoustic communication with the ros_acomms package [5]. Here, Volans is responsible for computing the speed of sound at different depths, which can be derived from the environmental data for water current and salinity at the time of communication. Volans publishes this information on a ROS topic that ros_acomms subscribes to. Similarly, Volans has been integrated with high-level command and control frameworks and post-mission analysis frameworks, where typically Volans listens for vehicle commands and publishes vehicle state and high-level perceptions.

VI. Related Work

In recent years, Gazebo has added specific features to support maritime applications. Some of these improvements were developed in the context of virtual maritime robotics competitions such as MBZIRC and the Virtual RobotX competitions. There is a long history of research groups building their own Gazebo-based maritime simulator (often starting from one from another group) including UWSim, WHOI’s ds_sim, MBARI’s LRAUV, and the Naval Postgraduate School’s DAVE [6]. Of those maritime simulators, only DAVE appears to be still actively maintained. HoloOcean [7] is a relatively new maritime simulator that, unlike the other simulators, is built on top Unreal Engine 4. While Volans definitely has some overlap with these other efforts, our focus is more driven by needs for applications that require long-duration navigation.

Acknowledgments

This work was supported in part by the Office of Naval Research (ONR) Code 32.

The author would like to thank the rest of the Unmanned Maritime Systems Division at Metron for the contributions to Volans over the years.

References

[1] M. M. M. Manhães, S. A. Scherer, M. Voss, L. R. Douat, and T. Rauschenbach, “UUV Simulator: A Gazebo-based package for underwater intervention and multi-robot simulation,” in *OCEANS 2016 MTS/IEEE Monterey*. IEEE, Sep. 2016.

[2] N. Koenig and A. Howard, “Design and use paradigms for Gazebo, an open-source multi-robot simulator,” in *Proc. 2004 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 2004, pp. 2149–2154.

[3] T. I. Fossen, *Marine Craft Hydrodynamics and Motion Control*, 2nd ed. Wiley & Sons, 2021.

[4] P. M. Newman, “MOOS - Mission Orientated Operating Suite,” Department of Engineering Science, University of Oxford, OUEL Report 2299/08, 2008.

[5] E. Gallimore, D. Giaya, B. Miller-Klugman, C. Fitzgerald, K. Griffen, L. Lindzey, and L. Freitag, “ROS message transport over underwater acoustic links with ros_acomms,” in *2022 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV)*, Sep. 2022, pp. 1–6.

[6] M. M. Zhang, W.-S. Choi, J. Herman, D. Davis, C. Vogt, M. McCarrin, Y. Vijay, D. Dutia, W. Lew, S. Peters, and B. Bingham, “DAVE Aquatic Virtual Environment: Toward a General Underwater Robotics Simulator,” in *2022 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV)*, Sep. 2022, pp. 1–8.

[7] E. Potokar, S. Ashford, M. Kaess, and J. Mangelson, “HoloOcean: An Underwater Robotics Simulator,” in *Proc. IEEE Intl. Conf. on Robotics and Automation, ICRA*, Philadelphia, PA, USA, May 2022.