Journal of the Korean Society of Marine Environment & Safety
Research Paper
Vol. 28, No. 3, pp. 441-450, May 31, 2022, ISSN 1229-3431(Print) / ISSN 2287-3341(Online) https://doi.org/10.7837/kosomes.2022.28.3.441

Faster-than-real-time Hybrid Automotive Underwater Glider Simulation for Ocean Mapping

Woen-Sug Choi[*†] · Brian Bingham[**] · Richard Camilli[***]

[*] Postdoctoral Researcher, Naval Postgraduate School, Monterey, CA 93943, United States
[**] Professor, Naval Postgraduate School, Monterey, CA 93943, United States
[***] Associate Scientist, Woods Hole Oceanographic Institution, Falmouth, MA 02543, United States

Abstract : *The introduction of autonomous underwater gliders (AUGs) specifically addresses the reduction of operational costs that were previously prohibited with conventional autonomous underwater vehicles (AUVs) using a "scaling-down" design philosophy by utilizing the characteristics of autonomous drifters to far extend operation duration and coverage. Long-duration, wide-area missions raise the cost and complexity of in-water testing for novel approaches to autonomous mission planning. As a result, a simulator that supports the rapid design, development, and testing of autonomy solutions across a wide range using software-in-the-loop simulation at faster-than-real-time speeds becomes critical. This paper describes a faster-than-real-time AUG simulator that can support high-resolution bathymetry for a wide variety of ocean environments, including ocean currents, various sensors, and vehicle dynamics. On top of the de facto standard ROS-Gazebo framework and open-sourced underwater vehicle simulation packages, features specific to AUGs for ocean mapping are developed. For vehicle dynamics, the next-generation hybrid autonomous underwater gliders (Hybrid-AUGs) operate with both the buoyancy engine and the thrusters to improve navigation for bathymetry mappings, e.g., line trajectory, are is implemented since because it can also describe conventional AUGs without the thrusters. The simulation results are validated with experiments while operating at 120 times faster than the real-time.*

Key words : *Autonomous Underwater Vehicle, Underwater Glider, Hybrid Underwater Glider, ROS-Gazebo Simulation, Software in the loop simulation*

1. Introduction

The advent of autonomous underwater vehicles has enabled significant advances in ocean floor mapping. However, the operation costs are impractical, typically exceeding 3,000 dollars per linear kilometer surveyed due to repositioning and launch/recovery. This low endurance is due to the power requirements of deep-sea operations. Autonomous underwater gliders (AUGs), incorporating the characteristics of both AUVs and autonomous drifters, scales down the power requirements and sizes to extend operation coverage and duration.

However, such long-duration, large-area missions of AUGs increase the cost and complexity of underwater testing of new approaches to autonomous mission planning and the development of hardware and software onboard. Therefore, a simulator that can support rapid design, development, and testing of autonomy solutions through software-in-the-loop simulation throughout the operation range, running at faster-than-real-time, is essential.

The ROS-Gazebo framework has become the de-facto standard environment for prototyping and testing robotic systems based on an open-source Gazebo robot simulator (Koenig and Howard, 2004). Many projects have exploited this framework to extend its capabilities for underwater simulations and maritime sensors for specific use cases (Prats et al., 2012; Kermorgant, 2014; Fernández, 2015; Gwon et al., 2017; Heriksen et al., 2016; Ciéslak, 2019; Suzuki and Kawabata, 2020; Jamieson et al., 2021). Among these projects, the UUV Simulator (Manháes et al., 2016), later inherited by Project DAVE (Dave, 2022), provides features for relatively general purposes. However, it lacks the critical features required for AUG simulation: glider dynamics, spatial/temporal ocean currents, faster-than-real-time simulation, hardware-in-the-loop interfaces, etc.

This paper explains AUG Sim, a simulator with extended capabilities customized to AUGs for ocean mapping. To support the mission characteristics of underwater gliders, critical features, including spatial/temporal-dependent ocean currents, dynamic assimilation of bathymetry, underwater glider dynamics, and faster-than-real-time capabilities, have been developed and demonstrated. Vehicle dynamics are formulated for next-generation hybrid underwater gliders (hybrid-AUGs), which operate in both the buoyancy engine of the AUGs and thrusters to improve

† Corresponding Author : woensug.choi.ks@nps.edu

navigation for bathymetry mappings, that is, line trajectory in ocean currents. The simulation results were validated with experiments while operating faster than real-time. This ability to test the autonomy solution of glider missions prior to deployment allows for rapid feedback early in the design cycle, demonstrating its utility in development.

2. Background and Objectives

In part of the development of EPIC-DAUG (Enhanced Propulsion Integrated Capability - Deep Autonomous Underwater Glider, Figure 1), which will incorporate significant innovation in propulsion, control, payload, communications, and navigation systems in development by the Woods Hole Oceanographic Institution (WHOI), a simulator capable of featuring various sensors, hybrid glider dynamics, and underwater environments for long-range missions operating faster-than-real-time was required.

The main objectives are to test high-level mission codes through low-level control of vehicle components, incorporating the following:

- Kinematics/dynamics of Hybrid-AUG physics
- Sensors (GPS, DVL, altimeter, USBL, Multi-beam sonar, etc.)
- Environmental influences (Ocean currents, Arbitrary Bathymetry)

Also, it should feature the following for mission planning:

- Multi-vehicle support, Faster-than-real-time, Live visualization
- Interactive interface with onboard driver for adaptive mission

There were no available simulators in the open-source community specific to Hybrid-AUG. However, relatively generic open-source underwater robotic simulator packages are available. AUG Sim is built on top of the simulator packages of Project DAVE, which is also based on the UUV simulator. The UUV simulator is a package containing necessary basic implementations for unmanned underwater vehicles (UUV), such as remotely operated vehicles (ROV) and autonomous underwater vehicles (AUV), for rapid prototyping and simulation in reproducible virtual environments. It is openly available for use as the base platform. Furthermore, Project DAVE added more features for manipulation and physics-based multibeam sonar (Choi et al., 2021). The key advantages of building on top of previous packages are that they inherit common underwater physics, sensors, utility plugins, and capabilities.

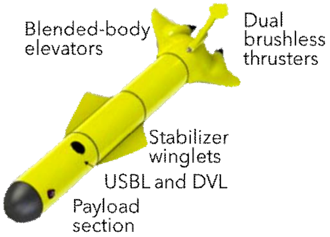


Fig. 1. Concept model rendering of the EPIC-DAUG glider.

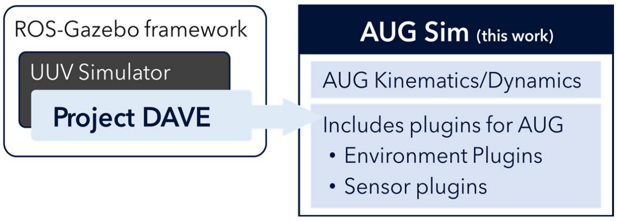


Fig. 2. Framework hierarchy of the AUG Sim.

3. Glider Dynamics

By taking advantage of the existing structure of the UUV simulator, modifications were conducted on the vehicle physics plugins to incorporate the glider's dynamic models, as shown in Figure 2. The distinctive characteristics of the gliders are the changes in the internal mass distribution and ballast volume, which correspond to the underwater object plugin and buoyant object plugin in the UUV simulator.

The coordinate frames to define the dynamics of the glider, two reference frames are defined: body-fixed and inertial. The body-fixed frame coincides with the glider's center of buoyancy (CB) when the foils are in a neutral position and is denoted by subscript cb. The inertial frame is fixed to the ocean surface and expressed by vector η. The x-axis of the body-fixed frame x_{cb} is aligned with the longitudinal body axis from the tail to the nose. The transverse axis y_{cb} is in its direction toward the starboard side of the body, and the z_{cb} is the normal direction axis with the bottom side as positive. For the inertial frame, z is fixed in the direction of gravity, x, and y are defined as perpendicular to each other, and z.

Notations for the positions and velocities of the motions are defined in the following vectors, consistent with Fossen's notation (Fossen, 1994).

$$\eta = \left[\eta_1^T, \eta_2^T\right]^T \quad \text{inertial position} \tag{1}$$

$$\eta_1 = [x, y, z]^T \quad \text{inertial transitional position}$$

$\eta_2 = [\phi,\theta,\psi]^T$	inertial orientation (Euler angles)
$\nu = \left[\nu_1^T,\nu_2^T\right]^T$	body-fixed velocity
$\nu_1 = [u,v,w]^T$	body-fixed linear velocity
$\nu_2 = [p,q,r]^T$	body-fixed angular velocity

In addition, the center of gravity(CG) and center of buoyancy(CB) are denoted as $r_{cg}= \left[x_{cg},y_{cg},z_{cg}\right]^T$ and $r_{cb}= \left[x_{cb},y_{cb},z_{cb}\right]^T$respectively in the inertial reference frame. The angle of attack α and sideslip angle β can be formulated as $\alpha=\tan^{-1}(w/u)$ and $\beta=\tan^{-1}(v/\sqrt{u^2+v^2+z^2})$.

The transformation matrix between the body-fixed and inertial reference frames is defined as follows:

$$\dot{\eta} = J(\eta)\nu,\ \dot{\eta}_1 = J_1(\eta_2)\nu_1,\ \dot{\eta}_2 = J_2(\eta_2)\nu_2 \tag{2}$$

where

$$J(\eta) = \begin{bmatrix} J_1 & 0_{3\times3} \\ 0_{3\times3} & J_2 \end{bmatrix}$$

$$J_1 = \begin{bmatrix} C\psi C\theta & -S\psi C\phi + C\psi S\theta S\phi & S\psi S\phi + C\psi C\phi S\theta \\ S\psi C\theta & C\psi C\phi + S\phi S\theta S\psi & -C\psi S\theta + S\theta S\psi C\phi \\ -S\theta & C\theta S\phi & C\theta C\phi \end{bmatrix}$$

$$J_2 = \begin{bmatrix} 1 & S\phi T\theta & C\phi T\theta \\ 0 & C\phi & -S\phi \\ 0 & S\phi/C\theta & C\phi/C\theta \end{bmatrix}$$

here, S, C, T are sine, cosine, and tangent respectively.

3.1 Equations of motion for hybrid glider

The complete non-linear 6-DOF equations of motion can be written in matrix form in the body-fixed reference frame as follows:

$$M\dot{\nu} + C\nu + D\nu + g(\eta) = \tau \tag{3}$$

where

$$\dot{\eta} = J\nu$$
$$M = M_{RB} + M_A,\ C = C_{RB} + C_A,\ D = D_f + D_h$$

here, M is the mass matrix including components of the rigid body M_{RB} and added mass M_A, C is the Coriolis-centripetal matrix with a rigid body and added mass components, D is the damping matrix considering glider hydrodynamics including lift, drag, and moment of the foil wings D_f and the hull D_h, $g(\eta)$ is the hydrostatic restoring force due to gravitational and buoyancy, and τ represents the external forces applied to the glider.

3.1.1 Rigid Body

The equations of motion for the rigid-body dynamics of the glider can be written as follows:

$$M_{RB}\dot{\nu} + C_{RB}\nu = \tau \tag{4}$$

where,$\tau = \tau_A + \tau_R + \tau_D + \tau_H$ represents the forces applied to the body, including added mass τ_A, damping τ_D, hydrostatic (restoring) τ_R, and hydrodynamic (propeller and foils) τ_H.

The mass and the Coriolis-centripetal matrix of the rigid body are defined as follows:

$$M_{RB} = \begin{bmatrix} M_{RB}^{11} & M_{RB}^{12} \\ M_{RB}^{21} & M_{RB}^{22} \end{bmatrix} = \begin{bmatrix} m(I_{3\times3}) & -mS(r_{cg}) \\ mS(r_{cg}) & I_0 \end{bmatrix} \tag{5}$$

$$C_{RB} = \begin{bmatrix} 0_{3\times3} & -S(M_{RB}^{11}\nu_1 + M_{RB}^{12}\nu_2) \\ -S(M_{RB}^{11}\nu_1 + M_{RB}^{12}\nu_2) & -S(M_{RB}^{21}\nu_1 + M_{RB}^{22}\nu_2) \end{bmatrix} \tag{6}$$

here, the superscripts represent 3×3 components, I_0 is the inertia matrix, and the operator $S(\cdot): R^3 \rightarrow R^{3\times3}$ is the skew-symmetric matrix operator defined as follows:

$$S(x) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \tag{7}$$

In rigid body matrices, m, which represents the mass of the body, is not consistent because of the change in volume in the buoyancy engine. The total mass m consists of fixed mass, body hull m_h and sliding mass m_s, and variable mass m_w dependent on volume displacement ∇.

$$m = m_h + m_s + m_w,\ m_w = \rho\nabla \tag{8}$$

In addition, the center of gravity vector r_{cg} changes as the sliding mass (battery pack) moves. Considering that the sliding mass moves only along the x-axis of the body-fixed reference frame x_b and the buoyancy engine is aligned along the same axis, it can be obtained as follows:

$$r_{cg} = \left[x_{cg},y_{cg},z_{cg}\right]^T,\ y_{cg} = z_{cg} = 0$$
$$x_{cg} = \frac{x_h m_h + x_s m_s + x_w m_w}{m_h + m_s + m_w} \tag{9}$$

where

$$x_h = -\frac{m_s}{m_h} x_{s,0},\ x_w = x_{w,0} + \frac{\nabla}{\pi r_w^2}$$

where x_s is the longitudinal distance to the sliding mass, x_w is the distance to the center of the water mass in the buoyancy engine, r_w is the radius of the buoyancy engine cylinder, and the subscript zero represents the initial value at the zero pitch condition.

For the inertia matrix I_0, it is assumed that the glider is symmetric ($I_{ij,i \neq j} = 0$) and is obtained by assuming a prolate spheroid as follows:

$$\begin{aligned} I_0 &= \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \\ I_{yy} &= I_{zz} = \frac{4}{15} m (a^2 + b^2),\ I_{xx} = \frac{8}{15} m b^2 \end{aligned} \tag{10}$$

here, a is half the length of the body, and b is the hull radius.

3.1.2 Added mass

The added mass force applied to the body τ_A can be expressed as follows (Isa et al., 2014):

$$\tau_A = -M_A \dot{\nu} - C_A \nu \tag{11}$$

where

$$M_A = \begin{bmatrix} I_{3\times 3} & S(r_{cg}) \\ 0_{3\times 3} & I_{3\times 3} \end{bmatrix} M_{A_{cg}} \begin{bmatrix} I_{3\times 3} & S(r_{cg})^T \\ 0_{3\times 3} & I_{3\times 3} \end{bmatrix}$$

$$C_A = \begin{bmatrix} 0_{3\times 3} & -S(M_A^{11}\nu_1 + M_A^{12}\nu_2 \\ -S(M_A^{11}\nu_1 + M_A^{12}\nu_2) & -S(M_A^{21}\nu_1 + M_A^{22}\nu_2) \end{bmatrix}$$

$$M_{A_{cg}} = -\mathrm{diag}\{X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}, K_{\dot{p}}, P_{\dot{q}}, N_{\dot{r}}\}$$

$$X_{\dot{u}} = -\frac{\alpha_0}{2-\alpha_0} m,\ Y_{\dot{v}} = Z_{\dot{w}} = -\frac{\beta_0}{2-\beta_0} m,\ K_{\dot{p}} = 0$$

$$N_{\dot{q}} = M_{\dot{r}} = -\frac{1}{5} \frac{(b^2 - a^2)^2 (\alpha_0 - \beta_0)}{2(b^2 - a^2) + (b^2 + a^2)(\beta_0 - \alpha_0)} m$$

$$\alpha_0 = \frac{2(1-e^2)}{e^3} \left(\frac{1}{2} \ln \frac{1+e}{1-e} - e \right)$$

$$\beta_0 = \frac{1}{e^2} - \frac{1-e^2}{2e^3} \ln \frac{1+e}{1-e}$$

here, $e = 1 - (b/a)^2$. The center of the added mass (CA) is assumed to be equal to CG to have the added mass inertia matrix diagonal $M_{A_{cg}}$ and transformed to the body-fixed reference frame.

3.1.3 Hydrostatic Forces

The gravitational and buoyant forces acting on the vehicle act as hydrostatic forces at the center of gravity and center of buoyancy, respectively. The hydrostatic restoring forces can be represented using the rotation matrix as follows:

$$\begin{aligned} f_G(\eta_2) &= J_1^{-1}(\eta_2)[0,0,W]^T \\ f_B(\eta_2) &= J_1^{-1}(\eta_2)[0,0,B]^T \end{aligned} \tag{12}$$

here, $W = mg$ is the gavitational force acting on a submerged vehicle, and $B = \rho g \nabla$ is the buoyancy force. For the glider, it is assumed that is neutrally buoyant and satisfies $W = B = mg = (m_{h+} m_s) g$.

Accordingly, the hydrostatic forces in the body-fixed reference frame are expressed as follows:

$$\begin{aligned} \tau_R = -g(\eta) &= \begin{bmatrix} \dfrac{f_G(\eta) + f_B(\eta)}{r_{cg} \times f_G(\eta) + r_{cb} \times f_B(\eta)} \end{bmatrix} \\ &= -\begin{bmatrix} (W-B)S\theta \\ -(W-B)C\theta S\phi \\ -(W-B)C\theta C\phi \\ -(y_{cg}W - y_{cb}B)C\theta C\phi + (z_{cg}W - z_{cb}B)C\theta S\phi \\ (z_{cg}W - z_{cb}B)S\theta + (x_{cg}W - x_{cb}B)C\theta C\phi \\ -(x_{cg}W - x_{cb}B)C\theta S\phi - (y_{cg}W - y_{cb}B)S\theta \end{bmatrix} \end{aligned} \tag{13}$$

3.1.4 Damping Forces

Damping forces may occur on the glider owing to skin friction acting on the surface. The damping forces depend on the relative velocity and angle of attack. Here, the coupled components of the damping terms were assumed to be insignificant. This can be represented as follows:

$$\tau_D = -\mathrm{diag}\{X_u, Y_v, Z_w, K_p, M_q, N_r\} \nu \tag{14}$$

here, the damping coefficients cannot to be determined analytically. Values were estimated from the iterative process and compared with the experimental data.

3.1.5 Hydrodynamic Forces

Hydrodynamic forces include lift, drag, and moment forces generated by the glider hull and foils. The forces are dependent on the relative velocity and the angle of attack. Because the lift, drag, and moment coefficients can be defined as constants in certain ranges of the angle of attack and velocity, they can be calculated using the following with prescribed coefficients: For the glider hull, the hydrodynamic drag, lift, and moment forces are represented as follows:

$$\boldsymbol{F_{D,h}} = 0.5\rho A_h C_{D,h} (\sqrt{u^2+v^2+w^2})^2 \tag{15}$$
$$\boldsymbol{F_{L,h}} = 0.5\rho A_h C_{L,h} (\sqrt{u^2+v^2+w^2})^2$$
$$\boldsymbol{M_h} = 0.5\rho A_h C_{M,h} (\sqrt{u^2+v^2+w^2})^2$$

where

$$C_{D,h} = 0.214 + 32.3\alpha_h^2 \tag{16}$$
$$C_{L,h} = 11.76\alpha_h + 4.6\alpha_h^2$$
$$C_{M,h} = 0.63\alpha_h$$

here, A_h is the cross-sectional area of the glider, and $C_{D,h}, C_{L,h}, C_{M,h}$ are the hydrodynamic drag, lift, and moment coefficients, respectively. $\alpha_h = \mathrm{atan}(w/u)$ is the angle of attack of the hull. The final calculation of hydrodynamic forces can be represented as follows:

$$\boldsymbol{\tau_H} = \begin{bmatrix} F_{L,h}\sin(\alpha_h) - F_{D,h}\cos(\alpha_h) \\ 0 \\ -F_{L,h}\cos(\alpha_h) - F_{D,h}\sin(\alpha_h) \\ 0 \\ F_{M,h} \\ 0 \end{bmatrix} + F_{propeller} \tag{17}$$

where $F_{propeller}$ is the force generated by the propeller propulsion (if zero, it is be a conventional underwater glider). Using the prescribed coefficients, the hydrodynamic forces due to foils can also be calculated using similar procedures. The forces due to the propeller should have their own interpolated formula that is dependent on the rotation speed and relative glider velocity.

3.1.6 Solver Schemes

Gazebo discretizes its physics equation in a simulated world using first-order Euler methods over time interval Δt. The equation is solved using an iterative Projected Gauss-Seidel algorithm implemented using the modified Open Dynamics Engine (ODE, Smite, 2015) to accelerate the convergence of the iterative solver (Hsu and Peters, 2014). Equation 1 is solved using the same Δt for every update of the simulated world using the Householder rank-revealing QR decomposition of a matrix with a column-pivoting linear solver, which can run much faster than the real-time using commonly available workstations. As the glider dynamics are calculated separately using first-order Euler methods, the accuracy depends on the Δt of the Gazebo world.

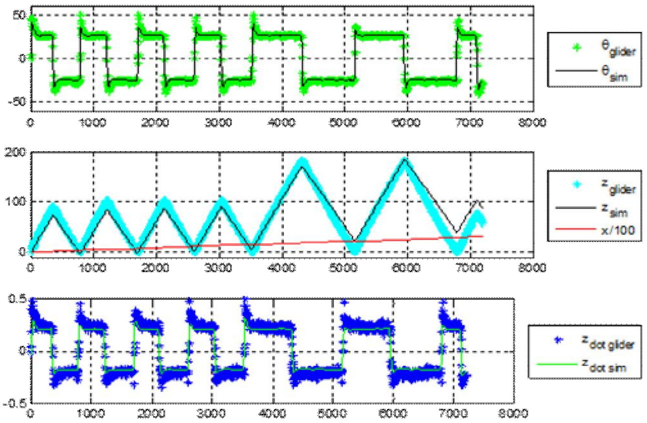


Fig. 3. Verification with field data using the in-house MATLAB simulator with predefined glider hardware parameters.

3.2 Validation and Verification

For the verification and validation of the vehicle dynamics formulated in Section 3.1, the in-house MATLAB code was validated with the model parameters defined using the knowledge of the physical system. The vehicle geometry and inertial properties are specified as follows:

$$y_{cg} = z_{cb} = y_{cb} = x_{cb} = 0,\ z_{cg} = 0.008 \tag{18}$$
$$x_{s,0} = 0.4,\ x_{w,0} = 0.75,\ r_w = 0.07,\ b = 0.105$$
$$m_h = 43.8,\ m_s = 7.8,\ m_w = 0.4$$
$$\delta x_v = 0.0001,\ g = 9.81,\ \rho = 1025$$

here, the offset distances $x_{s,0}$ and $x_{w,0}$ provide the positions of the associated masses for the zero pitch condition. The inputs m_s and ∇ are the deviations of the movable mass position and ballast volume, respectively, from the zero-pitch condition. In this case, the ballast was modeled as a cylindrical volume of radius r_w. The stationary vehicle mass offset δx_v was introduced to account for trimming errors.

Also, the added mass and damping terms were pre-defined. The damping coefficients cannot to be determined analytically. The added mass coefficients were calculated using the added mass calculations for a prolate ellipsoid and by adding the influence of the wings to the Z_w term. The estimates were generated by running the model several times with different values for each damping coefficient.

$$X_{\dot{u}} = -0.035,\ Z_{\dot{w}} = -0.934,\ M_{\dot{q}} = -0.1686 \tag{19}$$
$$X_u = 0,\ Z_w = -14,\ M_q = -12$$

The results obtained using the in-house MATLAB code with the measured ballast and mass distance from an actual mission were compared with the mission data in the Conception Bay South (CBS) in October 2010. The results well matched the data, as shown in Figure 3.

The Gazebo implementation was validated using the pre-validated MATLAB simulator. A glider dynamic model expression with inertial values using Fossen's hydrodynamic model is implemented on the Gazebo framework. The hydrodynamic model plugin source of the UUV simulator was modified to include the glider dynamics on the right-hand side of the external forces. The assumption applied is that the center of the added mass equals the center of gravity on the x-axis. For simplicity, the fins were removed in the simulator, and the thruster was assumed to force 1 N to the body frame x-axis at all times. The initial parameters that define the geometry of the model are as follows:

$$y_{cg} = z_{cb} = y_{cb} = x_{cb} = 0,\ z_{cg} = 0.0054 \tag{20}$$
$$x_{s,0} = 0.4,\ x_{w,0} = 0.711,\ r_w = 0.07,\ b = 0.105$$
$$m_h = 48.2,\ m_s = 7.8,\ m_w = 0.4$$
$$\delta x_v = 0.0001,\ g = 9.81,\ \rho = 1024$$

Cases for validation of the gazebo results are designed with various settings by controlling the position of the ballast tank x_w translated into the ballast tank pump volume and the position of the battery mass x_s as shown in Table 1. The results are well-matched to verify the implementation of underwater glider dynamics in the Gazebo, as shown in Figures 4 to 9.

Table 1. Cases for vehicle dynamics validation and verification

Cases	Battery position [mm]	Pump volume [mm^3]
Case 1	0	0
Case 2	0	-1.576E-4
Case 3	0	1.576E-4
Case 4	-0.018	0
Case 5	0.018	0
Case 6	sin(t/4), max=0.03	-1.576E-4

* For all cases, the thruster force is set to 1.0 [N]

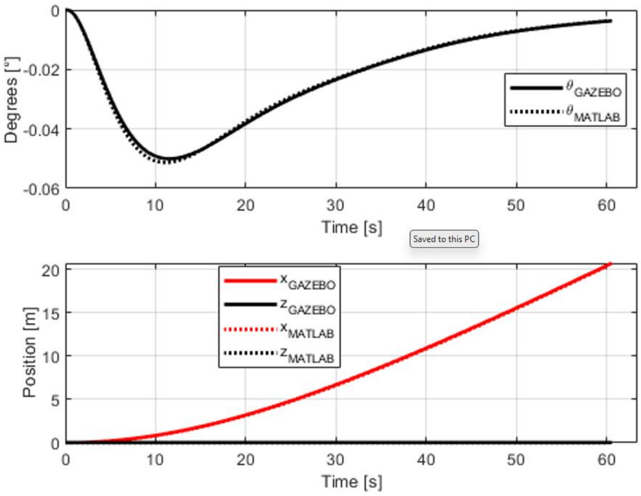


Fig. 4. Verification of gazebo implementation (Case 1).

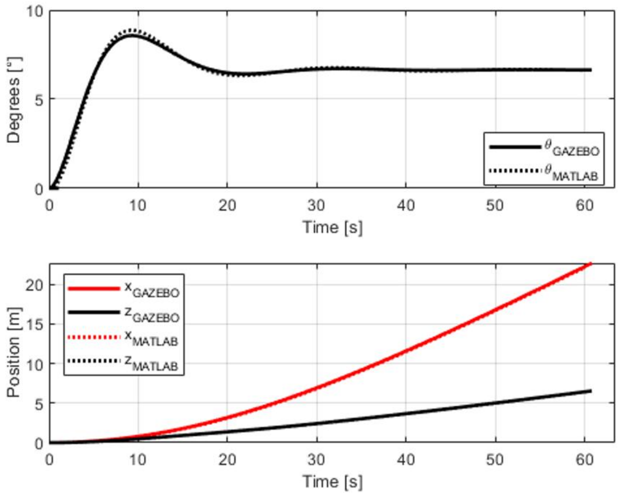


Fig. 5. Verification of gazebo implementation (Case 2).

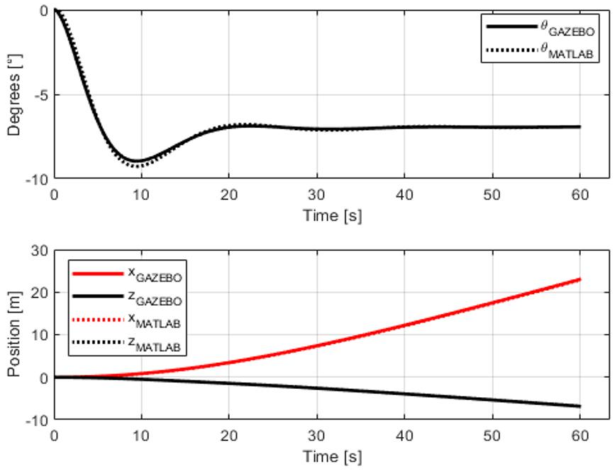


Fig. 6. Verification of gazebo implementation (Case 3).

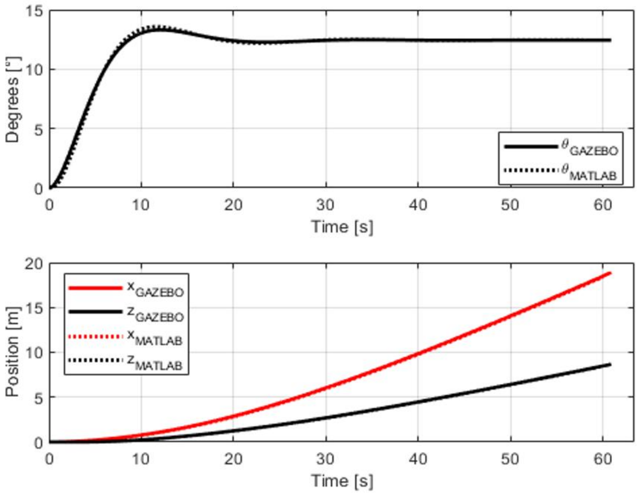


Fig. 7. Verification of gazebo implementation (Case 4).

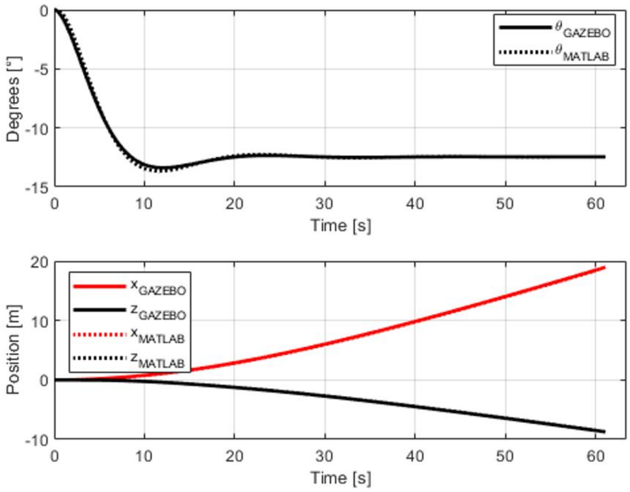


Fig. 8. Verification of gazebo implementation (Case 5).

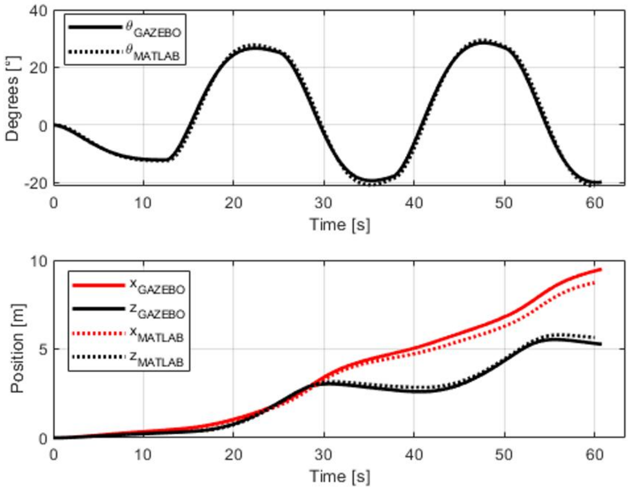


Fig. 9. Verification of gazebo implementation (Case 6).

4. Underwater Environments

Multiple vehicle supports were included in the simulator for the combined AUG-ASV and multiple AUG missions. The ocean currents and simulator-backseat driver interface should be defined with vehicle-wise ROS topics for multiple vehicle cases. The structure of the ROS nodes was designed to support multiple vehicles, including surface vehicles. Based on the stratified ocean current database, each vehicle is affected by ocean currents according to its depth. In addition, tidal oscillations were considered according to the position and time of the mission.

4.1 Ocean currents

4.1.1 Stratified ocean currents

The stratified ocean currents are definable in the simulator, with the stratified ocean current database of direction and magnitude at each depth. The developed plugin interpolates from the database to provide a vehicle with the appropriate ocean currents in the simulator at its depths. Using the interpolated value as the initial ocean current velocity, the ocean current, and the noise change can be defined using the Gauss-Markov process. Also, they can be changed on the fly or set to affect only specific periods.

4.1.2 Tidal ocean currents

Considering that the -100 to 100 cm/s ocean currents shift from ebb to flood, consideration of tidal ocean currents is essential. Using the database/prediction of the NOAA (National Oceanic and Atmospheric Administration), ocean currents oscillation for tidal periods were considered in the simulator. Tidal ocean currents can be defined in two ways: a database file downloaded from the NOAA website or harmonic constituents. Along with the simulation starting time specified at the GMT time frame and mean ebb/flood direction, the ocean currents were interpolated to include the tidal ocean currents at any time frame.

4.2 Dynamic bathymetry

The bathymetry of the operation region was assimilated into Gazebo simulator with the automatic spawning/removing of bathymetry tiles, as shown in Figure 10. The operation region of the glider is often a vast area; therefore the bathymetry digital elevation map (DEM) of the seabed cannot be loaded into the simulator at once. Therefore, bathymetry tiles are divided into multiple tiles (typically 1500 × 1500 m) and spawned on demand according to vehicle position (Bathymetry Converter, 2022). The overlapping regions also account for the smooth transition of the tiles. In addition, the number of tiles be kept in the scene is definable for the continuity of the sensory data stream from the sensors attached to the vehicle, for example the bottom tracking of the DVL sensor.

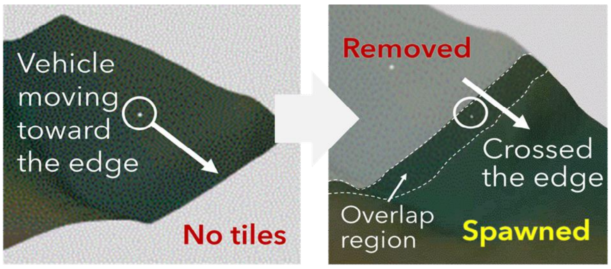


Fig. 10. Dynamic assimilation of bathymetry tiles according to the vehicle location.

5. Mission simulation

As the main objective of the simulator is to test high-level mission codes through low-level control of vehicles components, the simulator can run common mission profile definitions used for the SLOCUM glider generated using SFMC(Slocum Fleet Mission Control (Teledyne Webb Research), a mission specification code generator). The Slocum glider Simulator Driver (Slocum glider ROS interfaces, 2022) interprets the mission files and interacts with the AUG Sim through the ROS interface layer, which can be manipulated interactively during the run.

5.1 Interfaces

The interfaces between the Gazebo simulator and the backseat driver were structured as shown in Figure 11. The proposed plugin of the Gazebo simulator developed to support the required ROS topics for the back-seat driver to simulate missions.

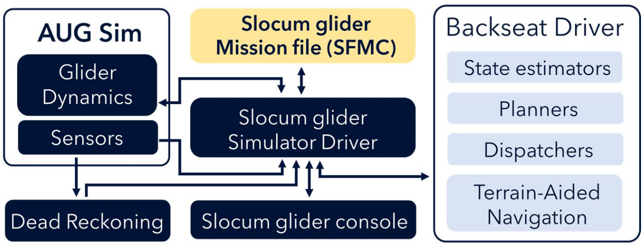


Fig. 11. Schematics of the simulation-In-Loop interfaces

5.1.1 Kinematics/Dynamics Control

There are two methods of utilizing a the control interface. Kinematic control is used the direct imposition of glider orientation and velocities in cases where the detailed configuration of the glider hardware and its hydrodynamic properties are not available. In this case, the induced velocity due to buoyancy propulsion was calculated using the flight model (Eichhron et al., 2020). On the other hand, the dynamic control fully calculates the glider rigid-body dynamics and hydrodynamics using Fossen's added mass/hydrostatic/hydrodynamic models with the vehicle dynamics formulation introduced in Section 3. The ROS nodes for the control interface receive a ROS message, including the following parameters:

- ■ Pitch control
  - Pitch can be controlled using the battery pack position
  - Change in pitch is defined by a first-order function with two coefficients for battery pack position ($Position = ax + b$)
- ■ Thruster control
  - The thruster force can be controlled using both the input voltage and power
  - Thruster power is defined by a second-order function with three coefficients for both voltage/power command values ($Power = ax^2 + bx + c$)
- ■ Rudder control
  - The rudder can be controlled by the vehicle heading angle or rudder rotation angle
- ■ Buoyancy engine control
  - The buoyancy pump volume is defined with the first-order function with two coefficients
  - The Flight model (Eichhron et. al., 2020) is adopted to change the attack angle according to the pumped volume to produce buoyancy-induced velocity and orientation change if hardware design including hydrodynamic properties are not determined

Also, the glider status is published as an ROS topic for backseat driver.

- ■ Glider status
  - Depth
  - Altimeter readings
  - Orientation (Roll, pitch, and yaw)
  - Magnetic heading
  - Estimated Latitude/Longitude with Dead Reckoning
  - Actual Latitude/Longitude

- Rudder angle
- Battery position
- Pumped volume

5.1.2 Sensors

Various sensors implemented in the physical glider were included in the Gazebo simulator, with corresponding sensor-specific parameters. IMU and GPS sensors for the AUG Sim were developed to calculate the reprojection between the geodetic coordinates and the ENU coordinate frame used inside the simulator. Other sensors, such as the altimeter, pressure, DVL, USBL, Multibeam sonar, and camera, are inherited from the UUV simulator and Project DAVE.

5.2 Results

A simulator can run the mission files with control and sensor interfaces to test high-level mission configurations providing simulated low-level control and sensory data. The results in Figure 12 show that more investigation is needed to obtain appropriate hydrodynamic parameters and a vertical ocean current database for simulation. However, the simulator could validate the mission configuration and autonomy of the controller from the start to the end location. These simulation-in-the-loop results imply that the simulator provided low-level glider status during the mission to test the autonomy solution to switch commanding states between ascending and descending equally to the field operations. In addition, the effects of ocean currents, ascending/descending angles, and GPS signals on the surface during the operation were well captured. The curved trajectory represent the effect of the ocean currents when the mission is to reposition the glider northbound in a straight line.

With the default sensors, which include an altimeter, pressure, IMU, GPS, and DVL sensors, the simulator was able to run 120.93 times faster than real-time at the i9-9900K (3.6GHz), GeForce RTX 2080Ti workstation, as shown in Figure 13 which is sufficient for simulating of AUG simulations for ocean mapping.

6. Conclusion

The advent of the Autonomous Underwater Glider has significantly reduced the operating costs for ocean floor mapping compared to other types of underwater vehicles. However, such long-duration, large-area missions of AUGs increase the cost and complexity of underwater testing. Therefore, a simulator that can support the rapid design, development, and testing of autonomy solutions through software-in-the-loop simulation throughout the operation range, faster-than-real-time is essential.

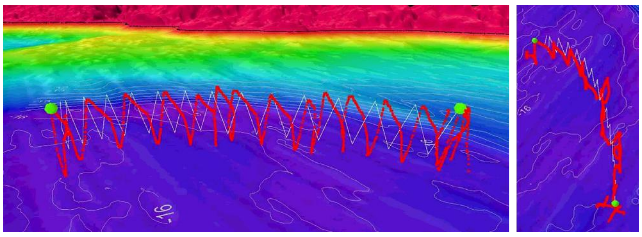
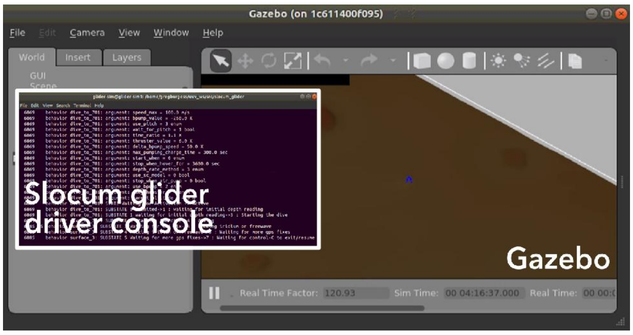


Fig. 12. Comparisons the simulated data with the field data (Post-processed using the Fledermaus, left: birds eye view, right: top view, red dots: field data, white line: simulated data).



glider-sim@glider-sim1: /home/pregburgess/uuv_ws/src/slocum_glider

```
6869    behavior dive_to_701: argument: speed_max = 100.0 m/s
6869    behavior dive_to_701: argument: bpump_value = -260.0 X
6869    behavior dive_to_701: argument: use_pitch = 3 enum
6869    behavior dive_to_701: argument: wait_for_pitch = 1 bool
6869    behavior dive_to_701: argument: time_ratio = 1.1 X
6869    behavior dive_to_701: argument: thruster_value = 0.0 X
6869    behavior dive_to_701: argument: delta_bpump_speed = 50.0 X
6869    behavior dive_to_701: argument: max_pumping_charge_time = 300.0 sec
6869    behavior dive_to_701: argument: start_when = 0 enum
6869    behavior dive_to_701: argument: stop_when_hover_for = 3600.0 sec
6869    behavior dive_to_701: argument: depth_rate_method = 3 enum
6869    behavior dive_to_701: argument: use_sc_model = 0 bool
6869    behavior dive_to_701: argument: stop_when_air_pump = 0 bool
6869    behavior dive_to_701: argument: use_bpump = 2 enum
6869    behavior dive_to_701: SUBSTATE 0 ...->1 : waiting for initial depth reading
6869    behavior dive_to_701: SUBSTATE 1 waiting for initial depth reading->3 : Starting the dive
6885    behavior surface_3: SUBSTATE 4 waiting for iridium or freewave
6885    behavior surface_3: SUBSTATE 4 ... freewave->5 : Waiting for more gps fixes
6885    behavior surface_3: SUBSTATE 5 Waiting for more gps fixes->7 : Waiting for control-C to exit/resume
```

Fig. 13. Screen capture of the simulator with the interactive driver console window interface.

AUG Sim, a simulator with extended capabilities customized to AUGs, is introduced. To support the mission characteristics of underwater gliders, critical features, including spatial/temporal dependent ocean currents, automatic assimilation of bathymetry, underwater glider dynamics, and faster-than-real-time, were developed. The vehicle dynamics formulation was verified using field experimental data.

Testing high-level mission codes generated using the SFMC became possible through low-level control of vehicle components and sensory data through the ROS interface layer using the mission simulation interface. The simulation-in-the-loop results are validated with experiments while operating 120 times faster than the real-time at a workstation. This ability to test the glider missions' autonomy solution before deployment enables feedback early in the design cycle to show its usefulness in developments.

In future studies, the adoption of FVCOM (a prognostic model library for ocean currents) using the Ocean Model Interfaces Library (JPL, 2022) will be planned. Also, continuous field data collection and fine-tuning of the parameters in the vehicle dynamics models are conducted. Migration to Ignition from the Gazebo Classic is also required for long-term support.

Acknowledgment

AUG Sim would have been impossible without the basis of existing open-source software, especially the Project DAVE(Dave, 2022), UUV Simulator(Manháes et al., 2016), and simulation from the WHOI Deep Submergence Lab(Vaughn and Suman, 2022). In addition, we would like to thank Gregory Burgess and Peter Ventola in MIT-WHOI Joint Program for testing and feedback during the developments; Michael Jakuba for providing initial template codes for bathymetry assimilation. This research was performed while the author held an NRC Research Associateship award at(Field Robotics Laboratory at NPS).

References

[1] Bathymetry converter(2022), https://github.com/Field-Robotics-Lab/Bathymetry_Converter (last accessed, 2022.5.10.).

[2] Choi, W. -S., D. R. Olson, D. Davis, M. Zhang, A. Racson, B. Bingham, M. McCarrin, C. Vogt, and J. Herman(2021), Physics-Based Modelling and Simulation of Multibeam Echosounder Perception for Autonomous Underwater Manipulation, Frontiers in Robotics and AI, Vol. 8, pp. 1-12.

[3] Ciéslak, P.(2019), Stonefish: An advanced open-source simulation tool designed for marine robotics, with a ROS interface, OCEANS2019.

[4] Eichhorn, M., D. Aragon, Y. A.Shardt, and H. Roarty(2020), Modeling for the performance of navigation, control and data post-processing of underwater gliders, Applied Ocean Research, 101, 102191.

[5] Fernández, J. J., J. Pérez, A. Penalver, J. Sales, D. Fornas, and P. J. Sanz(2015), Benchmarking using UWSim, Simurv and ROS: an autonomous free floating dredging intervention case study, OCEANS2015-Genova, IEEE.

[6] Fossen, T. I.(1994), Guidance and Control of Ocean Vehicles. John Wiley & Sons.

[7] Gwon, D. -H., J. Kim, M. H. Kim, H. G. Park, T. Y. Kim, and A. Kim(2017), Development of a side scan sonar module for the underwater simulator, IEEE International Conference on Ubiquitous Robots and Ambient Intelligence(URAI).

[8] Henriksen, E. H., I. Schjølberg, and T. B. Gjersvik(2016), UW MORSE: The underwater Modular Open Robot Simulation Engine, in IEEE/OES Autonomous Underwater Vehicles (AUV).

[9] Hsu, J. M. and S. C. Peters(2014), Extending open dynamics engine for the darpa virtual robotics.

[10] Isa, K., M. R. Arshad, and S. Ishak(2014), A hybrid-driven underwater glider model, hydrodynamics estimation, and an analysis of the motion control, Ocean Engineering, Vol. 81, pp. 111-129.

[11] Jamieson, S., K. Fathian, K. Khosoussi, J. P. How, and Y. Girdhar(2021), Multi-Robot Distributed Semantic Mapping in Unfamiliar Environments through Online Matching of Learned Representations, in ICRA2021.

[12] JPL(2022), Ocean model interfaces library https://github.com/nasa-jpl/ocean-model-interfaces(last accessed, 2022.5.10.).

[13] Kermorgant, O.(2014), A Dynamic Simulator for Underwater Vehicle Manipulators, International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR).

[14] Koenig, N. and A. Howard(2004), Design and use paradigms for Gazebo, an open-source multi-robot simulator, IROS 2004.

[15] Manháes, M. M. M., S. A. Scherer, M. Voss, L. R. Douat, and T. Rauschenbach(2016), UUV simulator: A gazebo-based package for underwater intervention and multi-robot simulation, in OCEANS 2016 MTS/IEEE Monterey.

[16] Prats, M., J. Pérez, J. J. Fernández, and P. J. Sanz(2012), An open source tool for simulation and supervision of underwater intervention missions, IROS 2012.

[17] Project dave(2022), https://github.com/Field-Robotics-Lab/dave (last accessed, 2022.5.10.).

[18] Suzuki, K. and K. Kawabata(2020), Development of a Simulator for Underwater Reconnaissance Tasks by Utilizing Remotely Operated Robots, IEEE/SICE International Symposium on System Integration (SII).

[19] Slocum glider ROS interfaces(2022), https://gitlab.com/sentinel-aug/ro-s/slocum_glider (last accessed, 2022.5.10.).

[20] Smith, R.(2015), Open dynamics engine ode. multibody dynamics simulation software.challenge, in International Conference on Simulation, Modeling, and Programming for Autonomous Robots. Springer, pp. 37-48.

[21] T. W. Research, Slocum Fleet Mission Control User Manual, Teledyne Webb Research.

Received : 2022. 04. 14.
Revised : 2022. 05. 10.
Accepted : 2022. 05. 28.