

Article

A Backseat Control Architecture for a Slocum Glider

Yaomei Wang ^{1,*} , Craig Bulger ² , Worakanok Thanyamanta ¹  and Neil Bose ¹ 

¹ Department of Ocean and Naval Architectural Engineering, Memorial University of Newfoundland, St. John's, NL A1B 3X7, Canada; s02wt@mun.ca (W.T.); nbose@mun.ca (N.B.)

² Centre for Applied Ocean Technology, Fisheries and Marine Institute of Memorial University of Newfoundland, St. John's, NL A1C 5R3, Canada; Craig.Bulger@mi.mun.ca

* Correspondence: yaomeiw@mun.ca

Abstract: Adaptive sampling provides an innovative and favorable method of improving the effectiveness of underwater vehicles in collecting data. Adaptive sampling works by controlling an underwater vehicle by using measurements from sensors and states of the vehicle. A backseat driver system was developed in this work and installed on a Slocum glider to equip it with an ability to perform adaptive sampling tasks underwater. This backseat driver communicated with the main vehicle control system of the glider through a robot operating system (ROS) interface. The external control algorithms were implemented through ROS nodes, which subscribed simulated sensor measurements and states of the glider and published desired states to the glider. The glider was set up in simulation mode to test the performance of the backseat driver as integrated into the control architecture of the glider. Results from the tests revealed that the backseat driver could effectively instruct the depth, heading, and waypoints as well as activate or deactivate behaviors adaptively. The developed backseat driver will be tested in future field experiments with sensors included and safety rules implemented before being applied in adaptive sampling missions such as adaptive oil spill sampling.



Citation: Wang, Y.; Bulger, C.; Thanyamanta, W.; Bose, N. A. Backseat Control Architecture for a Slocum Glider. *J. Mar. Sci. Eng.* **2021**, *9*, 532. <https://doi.org/10.3390/jmse9050532>

Academic Editors: Mario Brito and Worakanok Thanyamanta

Received: 16 April 2021

Accepted: 13 May 2021

Published: 15 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: Slocum glider; backseat driver; adaptive behavior

1. Introduction

Underwater gliders are characterized by long endurance, low energy consumption, and low noise [1]. The sawtooth-like motion endows gliders with inherent advantages in sampling data in the water column. Sensors such as current profilers, fluorometers, and sonar have been equipped on gliders for various missions [2–4]. Underwater gliders have been active components in underwater observation including ecosystem monitoring [5], hurricane prediction [6], climate change investigation [7], hydrography and circulation measurement [8], oil spill survey [9], and gradient tracking [10]. A fleet of underwater gliders was proposed in the late 1980s to monitor the global ocean [11]. Target-based control is critical to getting the utmost out of an underwater glider fleet. In addition, the growing needs for more complex missions such as delineating underwater plumes call for underwater gliders with the ability to respond in real time. Adaptive sampling has emerged as a possible solution.

Adaptive missions or adaptive sampling means that the vehicle changes its states such as heading and depth based on sensor measurements obtained in real time (see Figure 1) [12,13]. The adaptive strategy can help drive gliders to regions of higher interest to, for example, increase the quantity of information acquired during a mission [14]. Theoretically, this intelligent work increases the real working time of gliders by spending more time in information rich areas. In the work of Fiorelli et al. [10], an adaptive sampling strategy was proposed for the coordination of gliders. The glider fleet was controlled by a virtual body and artificial potential multi-vehicle control method to maintain the group motion of gliders. The artificial potentials defined the interaction between vehicles

and reference points on a virtual body. The direction of the virtual body was adaptively controlled by a gradient estimated from the feature measurement of gliders. This gradient information could drive gliders to the maximum or minimum of a field, or to track the boundary of oceanic features.



Figure 1. Structure of an adaptive sampling system.

Adaptive sampling can be realized in practice by adding a backseat driver to an underwater vehicle. Backseat drivers have been successfully implemented on several vehicles. A backseat controller developed by Naglak et al. [15] was assembled in a General Dynamics Bluefin SandShark AUV. This backseat controller was supported by a small Raspberry Pi single board computer, running the Robot Operating System (ROS) and OpenCV to provide control algorithms and computer vision libraries. The backseat driver processed sensor data, ran control algorithms, and sent commands to the frontseat driver of the vehicle. The frontseat driver was housed in the commercial-off-the-shelf vehicle. This frontseat-backseat control architecture was tested in an open-water environment and was planned to be applied to an open-source vehicle named ROUGHIE [16]. A ROS-based system was also used on a Hydroid REMUS 100 AUV as a backseat driver [17]. In the control architecture, an application programming interface RECON on the REMUS vehicle computer provided an interface between the vehicle computer and a sensor/autonomy computer. The RECON interface was handled by a library pyREMUS on the sensor/autonomy computer. This pyREMUS library could not only be used to build an interface with ROS packages, but also connect with the Mission Oriented Operating Suite (MOOS) or sensors.

Eickstedt and Sideleau [18] focused on the implementation of a backseat driver architecture on an Iver2 AUV manufactured by Ocean Server Technology. This backseat driver architecture was built based on the Mission Oriented Operating Suite-Interval Programming (MOOS-IvP), an autonomy software for supporting the operation of autonomous marine vehicles [19]. In their implementation, a MOOS module, iOceanServerCommns, created an interface between a backseat controller and the main vehicle control system. An intelligent control component in the backseat controller received data both through the interface and sensors connected directly to the backseat computer. This autonomy system provided decisions on states of the vehicle such as speed, heading, and depth to the dynamic control of the main vehicle control system. The dynamic control component in the main vehicle control system was responsible for executing the commands from the backseat controller and sending navigation information to the backseat driver. In addition to the Iver2 AUV, there are some other underwater vehicles that have implemented a backseat driver by using MOOS-IvP such as the Bluefin 9 and Bluefin 21 [20], Teledyne Gavia AUV [21], and so on [18]. To facilitate AUVs with adaptive maneuvering capability for homing to a single beacon, a homing application pHomeToBeacon based on the MOOS-IvP was developed and demonstrated on a Teledyne Gavia AUV [21]. The application pHomeToBeacon received range reports and AUV positions from the frontseat driver through an interface called iGavia between the frontseat driver and the backseat driver. The application then processed these inputs with a localization algorithm to estimate the beacon position. The beacon position was used to obtain the dynamic homing waypoints that were sent back to the frontseat driver operating system to achieve adaptive maneuvering ability.

The existing control system of Slocum gliders is not favorable for carrying out an adaptive mission such as delineating underwater oil plumes intelligently. A mission with existing gliders is defined by simple methods such as waypoints before the gliders are deployed. Due to ocean dynamics or patchy characteristics of oil plumes, gliders can miss some areas or spend a lot of time outside the plume. An adaptive control system can solve this problem by enabling the glider to change its states based on the sensor measurements.

This will increase the quality of information acquired during a mission, particularly in time-sensitive missions like an oil spill response [10]. To equip Slocum gliders with the ability to do an adaptive mission, a backseat driver was developed for Slocum gliders in this work. This backseat driver enabled a glider to use the state information from the main vehicle control system and real-time measurements from sensors to reevaluate its mission when the glider was underwater. The main focus of this work was to interface external codes with a Slocum glider. This paper also investigated the ability of the backseat control system to control the states such as depth, heading, and waypoints of the Slocum glider. With the possibility of controlling the states of the glider through the backseat control system, additional sensors can be included in the backseat driver to further control the motion of the glider such as improving the path following performance considering water currents [22–24]. The glider used in this work and the developed backseat driver are introduced in the next section. In Section 3, the tests of the backseat driver through simulations are presented, followed by a discussion. A conclusion is included in the final section.

2. System Architecture

This section introduces the Slocum glider used for the development and the developed backseat driver. This work mainly focuses on the software part of the backseat driver that has been successfully tested in the Slocum glider.

2.1. The Slocum Glider

The glider used in this work was a Slocum Generation 1 (G1) glider (see Figure 2). Specifications of this glider are shown in Table 1. This glider was equipped with a 200-m depth rated ballast pump and had a rated horizontal speed of 0.4 m/s. The glider changes its weight by pumping water in and out of the glider and converts vertical motion into horizontal motion by using body and wing lift. The 200-m depth rated ballast pump assembly used can move 504 cm³ of water into and out of the glider [25]. Given that the buoyancy changes from a density change between freshwater and seawater is in the order of 1.04 L to 1.05 L, the ballast pump cannot compensate for the force induced by such a large change in water density. The structure of the glider system including the main part can be referred to the manual of the glider [25].



Figure 2. Slocum G1 glider.

Table 1. Specifications of Slocum G1 glider [25].

Parameter	Value
Weight in air	~52 Kg
Dimension	0.213 m (Diameter)
	1.003 m (Width)
	1.5 m (Length)
Operation Depth	4–200 m
Speed	0.4 m/s horizontal
Energy	Alkaline batteries (primary)
Sensors	GPS, altimeter, acoustic modem, conductivity, temperature, depth sensor

The Slocum glider can be commanded to conduct various behaviors by defining mission files that are sent to the glider before a mission. Behaviors that can be defined in a mission file include, but are not limited to:

- Yo (a single up and down pattern through water column) behavior: instruct the glider to dive and climb by setting the depth, altitude, and the number of yos;
- Go to a waypoint or waypoint list: instruct the glider to go to a waypoint or waypoint list defined in mission files;
- Set a heading: instruct the glider to follow a predefined heading;
- Surface: instruct the glider to surface for communication or recovery; and
- Abend: define conditions when a mission should be aborted.

2.2. Backseat Driver

A backseat driver system was developed for the Slocum G1 glider. This backseat driver had some features that enabled the glider to conduct adaptive missions and facilitated the application of gliders in various missions:

- Onboard replanning. The backseat driver system could receive state information from the main vehicle control system and measurements from sensors connected with the backseat driver. With this information, the backseat driver could replan the trajectory or mission target and publish the new plan to the main vehicle controller in real-time. Onboard replanning enables the glider to deal with unexpected and unforeseen situations and improve the quality of collected data.
- Modular design. The modular design in both software and hardware of the backseat driver system facilitated the reconfiguration of the sensors and control system. A new sensor could be easily added to the backseat driver without changing other modules of the control system. The structure of the backseat driver system is concise and clear.
- Energy efficient. The backseat driver computer and the sensors used for backseat control were energy efficient, which used minimum energy from the onboard battery. This feature ensured that the Slocum glider would keep its favorable long endurance with low battery energy consumption.

2.2.1. Software for the Backseat Driver

The software for the backseat driver consisted of an interface and external controllers.

(1) Interface for the backseat driver

The communication between the backseat driver and the main vehicle computer was realized through a ROS interface, provided by Teledyne Webb Research. ROS is a collection of libraries, tools, and conventions that simplify the development of robot application [26]. ROS enables a transfer of codes developed in other domains such as aerial robotics to underwater robotics [15]. With the ROS interface, state parameters of the glider such as depth (m_{depth}), heading ($m_{heading}$), and location (m_{gps_lat} , m_{gps_lon} , m_{lat} , m_{lon}) can be sent from the main vehicle control system to the backseat driver. Likewise, commands such as activation or deactivation of behaviors ($u_{mission_mode}$), desired heading and depth can be sent from the backseat driver to the main vehicle control

system. Examples of the parameters that can be exchanged between the main vehicle control system and the backseat driver are shown in Figure 4. Before the introduction of the backseat driver, the condition for the activation and deactivation of behaviors were defined in mission files before deployment. This restricted the implementation of combinations of individual behaviors and generation of new behaviors. The backseat controllers enabled the vehicle to activate or deactivate behaviors adaptively, for example, to activate a surface behavior by publishing parameter $u_{mission_mode}$ when the vehicle is in the water for a period of time and needs to come to the surface. Parameters such as $u_{mission_param_a}$ and $u_{mission_param_b}$ were used to represent mission parameters such as targeted diving depth and targeted heading.

(2) External controllers

The backseat driver adaptively replans a mission by subscribing the state parameters such as location and heading of the vehicle from the main vehicle control system or measurements from sensors. Then, the backseat driver processes the subscribed data and publishes desired state parameter to the glider. This is realized through external controllers in the backseat driver implemented as ROS nodes. These nodes can publish and subscribe information to any nodes in the backseat driver and process the information with control algorithms. For example, a glider might be deployed to adaptively change the target depth of its diving and climbing in the yo behavior based on the fluorescence reading from an onboard fluorometer. In this case, the external controllers consist of two ROS nodes (see Figure 3). One is a node called the fluorometer controller, which collects the readings of fluorescence from the fluorometer and publishes the fluorescence data to other nodes. Another is a depth controller node, which sends the target depth to the science processor according to the subscribed fluorescence data. The introduction of individual, independent nodes in the external controllers is beneficial for module reuse, which helps accelerate the development process of the controllers. Existing modules can be combined or expanded, providing the external controller with more powerful or new functions. For instance, a waypoint controller is added to a backseat driver consisting of a depth controller and a fluorometer controller without changing the configurations of the existing backseat driver (Figure 3). Data from the fluorometer controller can now be supplied to both the depth controller and waypoint controller at the same time. This added waypoint controller subscribes the data related to fluorescence from the fluorometer controller and publishes the processed data to the glider as part of a new adaptive waypoint behavior.

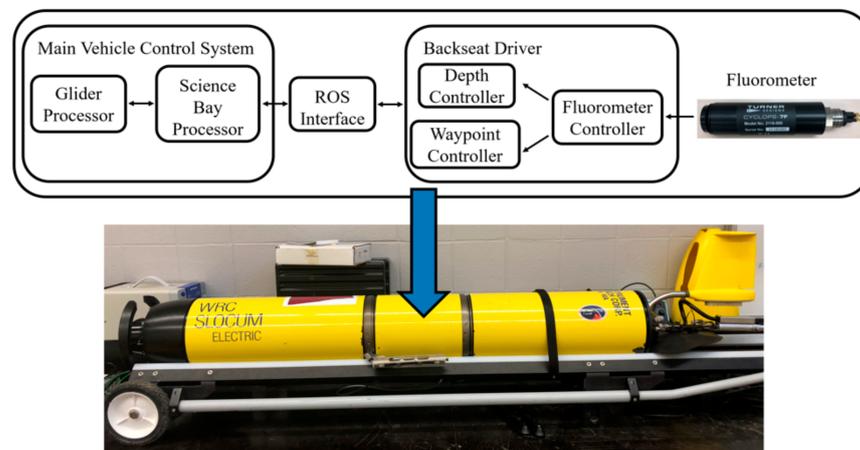


Figure 3. Backseat control architecture of Slocum glider with a fluorometer supporting the control of depth and waypoint.

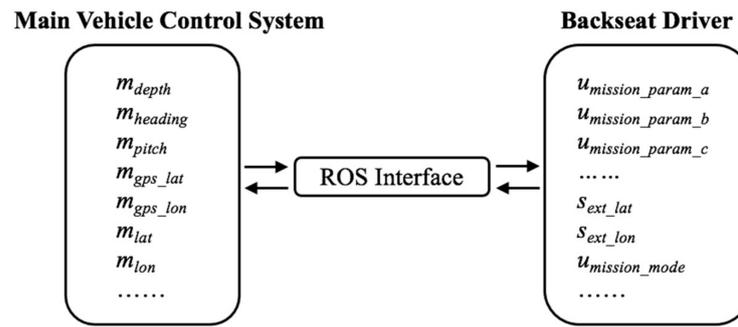


Figure 4. Data exchange between the main vehicle control system and backseat driver.

2.2.2. Hardware of the Backseat Driver

A Beaglebone Black was used as the platform to support the running of the ROS interface, the backseat driver control algorithms, and the sensor controllers. The BeagleBone Black is a small, low-cost, and community-supported single board computer [27]. The computer runs on a 1 GHz processor with a 512 MB RAM, which is sufficient for real-time data processing and backseat control. It requires a maximum of 500 mA if the board is powered from the USB port. This computer supports multiple operating systems such as Debian, Ubuntu, and Android for different users. A BeagleBone Black has two headers, each with 46 pins. These pins provide various functions such as seven analog input pins that can be used for reading sensor data.

3. Simulation

A series of simulations were conducted in a lab to test the performance of the backseat driver before being tested in field experiments. The test was done by connecting the BeagleBone Black, serving as the backseat driver computer to the Slocum glider set up in simulation mode (see Figure 5). The glider’s electronics and vehicle control software were used as a hardware-in-the-loop simulator to test our missions. The mathematical model of the glider in the simulator control software is proprietary to Teledyne Webb, the glider manufacturer. A similar empirical linear model for glider velocity has been published in the work of Zhou [28–30]. The conducted simulations used the actual control loops of the glider instead of those developed by the authors, which may introduce simplification and inaccuracy. The backseat driver has the ability to change the depth, heading, waypoint of the glider, and activate or deactivate behaviors during missions. Therefore, the performance of the glider in adaptive depth changing, adaptive heading changing, adaptive waypoint changing, and adaptive activation/deactivation of behaviors were tested through simulations.

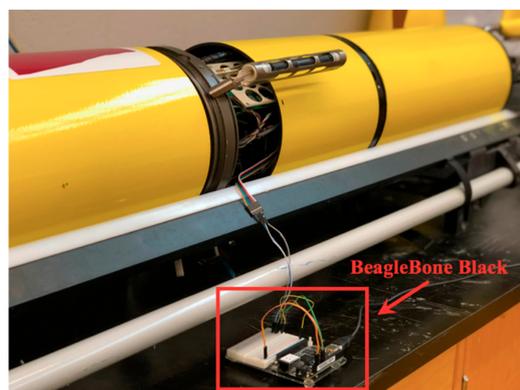


Figure 5. Connecting BeagleBone Black to the Slocum G1 glider to test the performance of the backseat driver.

3.1. Adaptive Depth Changing

The maximum diving depth and the minimum climb depth of a glider are usually defined before a mission. The ability to adaptively change these depths of a glider is beneficial and can reduce mission time. For example, the glider can change its depth based on sensor readings to spend more time in the depth layer where the target is found. In addition, the ability to change depth can help increase safety by reducing risks associated with the change in water density. By using the backseat driver, the glider can analyze the real-time water density collected by its conductivity, temperature, and depth sensor and respond to any risky water density change. For example, the backseat driver can command the glider to move to a depth where the water density change is within that for which the glider can compensate.

3.1.1. Depth Changing When a Waypoint Is Reached

In order to test whether the backseat driver can control the glider to change its maximum depth during a mission, the glider was commanded to go to several predefined waypoints in sequence, as shown in Figure 6 and Table 2. Prior to the mission, the glider’s maximum depth was set to 20 m. The backseat driver was programmed to increase the maximum diving depth of the glider by 10 m each time it reached a waypoint (defined as when the glider was within 10 m from the target waypoint). When the first waypoint (Wpt 1) was reached, the glider would change its maximum diving depth from 20 m to 30 m. When the last waypoint (end point) was reached, the glider surfaced and finished its mission.

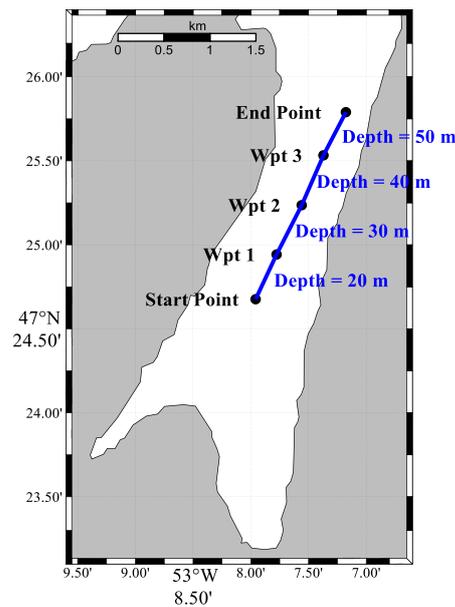


Figure 6. Trajectory of the Slocum glider in the mission of changing the maximum diving depth when a waypoint was reached.

Table 2. Waypoint list for a mission of the Slocum glider with corresponding maximum diving depth after a waypoint was reached.

Waypoints	Coordinate	Depth
Start Point	47°24.6763' N, 53°07.9585' W	20 m
Wpt 1	47°24.9426' N, 53°07.7769' W	30 m
Wpt 2	47°25.2362' N, 53°07.5591' W	40 m
Wpt 3	47°25.5328' N, 53°07.3706' W	50 m
End Point	47°25.7897' N, 53°07.1758' W	50 m

In this simulation, we used a depth controller in the backseat driver (see Figure 7) to control the maximum diving depth of a yo behavior. This depth controller subscribed latitude (m_{lat}) and longitude (m_{lon}) information from the main vehicle control system in real time. Then, the position information was processed by the depth controller to calculate the distance from the predefined waypoints. When a waypoint was reached, the depth controller published a new maximum dive depth ($u_{mission_param_a}$) to the main vehicle controller to change the diving depth of the glider. Otherwise, the glider would keep the maximum diving depth of the yo behavior unchanged. The initial value of $u_{mission_param_a}$ was set to be 5 m.



Figure 7. Exchange of data between the main vehicle control system and a depth controller when the glider changed its depth when a waypoint was reached.

Results from this simulation are presented in Figure 8. The glider went to a depth of around 20 m on its first dive, showing that the $u_{mission_param_a}$ was successfully published to the glider main controller as the glider changed its maximum diving depth to the newly received one. When the first waypoint (Wpt 1) was reached, the glider had already finished its diving behavior and had started to climb. In this case, the backseat driver would command the glider to a new maximum depth of 30 m once it started the next diving behavior. Except for the last diving behavior when the glider finished its mission before reaching the target depth, the glider overshoot the target depth in all its dives, as it took some finite time for the glider to slow down its dive and start climbing again.

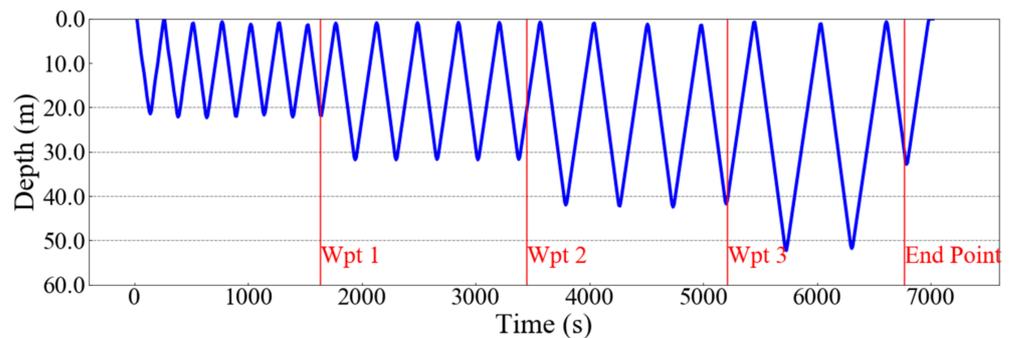


Figure 8. The depth of the glider under the control of a depth controller, which changed the maximum dive depth when a waypoint was reached.

3.1.2. Depth Changing Based on a Simplified Fluorescence Field

The objective of this simulation was to test the ability to command the glider to change its target depth of diving behavior based on fluorescence readings. A simplified artificial fluorescence field in the vertical plane of the water column was defined by a sinusoid:

$$y = 5 \sin\left(\frac{2\pi}{150}x\right) + 30 \tag{1}$$

where x is the distance that the glider has moved in the horizontal direction. It was assumed that fluorescence was present in all of the region above the sine curve in the vertical plane and that there was no fluorescence below the sinusoid. In order to get information of the fluorescence field, it is best for the glider to stay in the information-rich area. Therefore,

the glider was commanded to change the maximum dive depth of the yo behavior when there was no fluorescence detected. In the simulation, the glider moved from start point [47°24.6763' N, 53°07.9585' W] to the end point [47°25.7897' N, 53°07.1758' W] as shown in Figure 9. The mission was ended when the distance to the end point was within 10 m.

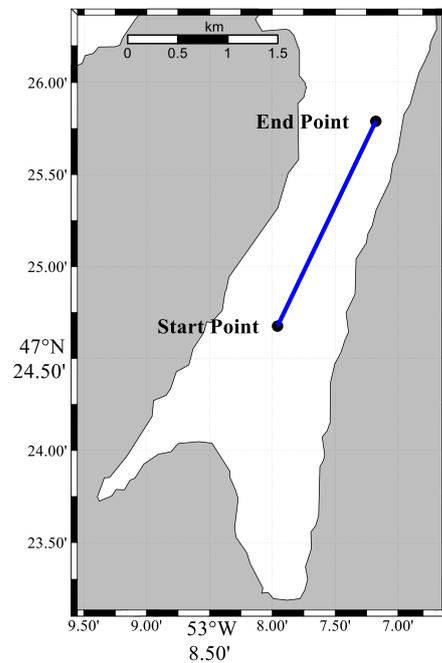


Figure 9. Trajectory of the glider in the mission of changing maximum dive depth based on a simplified fluorescence field.

In the control architecture, the main vehicle control system sent current latitude (m_{lat}), longitude (m_{lon}) and depth (m_{depth}) of the glider to a depth controller in the backseat driver (see Figure 10). In the depth controller, the backseat control algorithm (see Table 3) compared the depth of the glider to the vertical distribution of the fluorescence field. If the glider was within the fluorescence field, a larger maximum diving depth was set $m_{depth} + \delta_{depth}$, where m_{depth} is the measured depth of the glider and δ_{depth} is a parameter with a value larger than 0 m, in order for the glider to dive deeper toward the boundary of the fluorescence field. Otherwise, the desired maximum diving depth was set to m_{depth} , which is the current depth of the glider. The depth controller then sent the desired maximum diving depth ($u_{mission_param_a}$ in Figure 10) to the main vehicle control system to replan the diving behavior.



Figure 10. Exchange of data between the main vehicle control system and a depth controller when the glider changed its depth based on a simplified fluorescence field.

Table 3. Control algorithm of a depth controller for backseat control when the glider changes its depth based on a simplified fluorescence field.

Pseudo code for the Depth Controller

```

# The location of the glider at the start point is [mlat_start, mlon_start]
Node Depth Controller subscribes latitude (mlat), longitude (mlon), and depth (mdepth) of the glider from the
main vehicle control system
# dis is the distance that the glider has moved from the start point
dis = the distance between [mlat, mlon] and [mlat_start, mlon_start]
# y is the lower boundary of the simulated plume in the water column
y = 5*sin(2*pi/150*dis) + 30
# umission_param_a is the desired maximum diving depth
if mdepth < y
    umission_param_a = mdepth + δdepth
else
    umission_param_a = mdepth
end if
publish umission_param_a to the Node Main Vehicle Control System
    
```

Depths of the glider in the simulation when δ_{depth} was set to be 5 m is presented in Figure 11. The glider reacted to the distribution of the fluorescence field on each dive and changed its maximum dive depth according to the boundary of the fluorescence field. The glider did not reach the boundary of the simplified fluorescence field in the last dive behavior as the end point of the mission was reached.

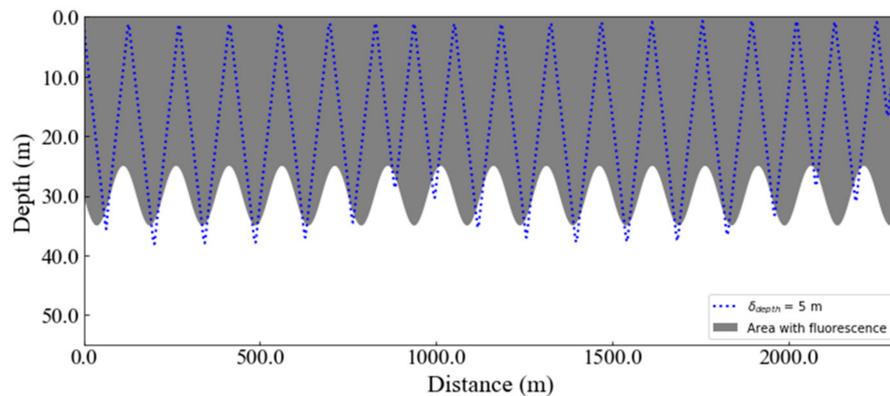


Figure 11. The depth of the glider under the control of a depth controller, which changed the maximum dive depth according to a simplified fluorescence field ($\delta_{depth} = 5$ m).

The performance of the depth controller when δ_{depth} was set to be 1 m, 3 m, and 5 m was compared. Figure 12 shows the depths of the glider when its distance travelled was between 800 m and 1300 m for better resolution. While the values of δ_{depth} were different, the response of the glider to the boundary of the fluorescence field were similar. For example, at label 1 in Figure 12, when the distance travelled of the glider was nearly 900 m, the fluorescence distributed to a deeper depth with the increase in the distance travelled. The overshoot value of the depth of the glider in the diving behavior relative to the boundary of the fluorescence field was almost the same for the three δ_{depth} values at label 1. This also happened at 1000 m (label 2 in Figure 12). The similar overshoot values showed that these different δ_{depth} values had little influence on the performance of the glider in reacting to the boundary of the fluorescence field. Even so, the trajectories of the glider in the three simulations were slightly different, as shown in Figure 12. This was caused by the iteration error in the main vehicle control system when it calculated the position and heading of the glider underwater. When the value of δ_{depth} was small, which meant the next target depth was close such as 1 m, it would not take a long time before reaching the target depth. As there was little difference in the diving of gliders at different

values of δ_{depth} , the depth controller was not sensitive to the value of δ_{depth} . The backseat driver could control well at a depth accuracy of 1 m. This also reviewed that the backseat driver could receive states of the glider from the main vehicle control system and control the depth of the glider in a timely manner.

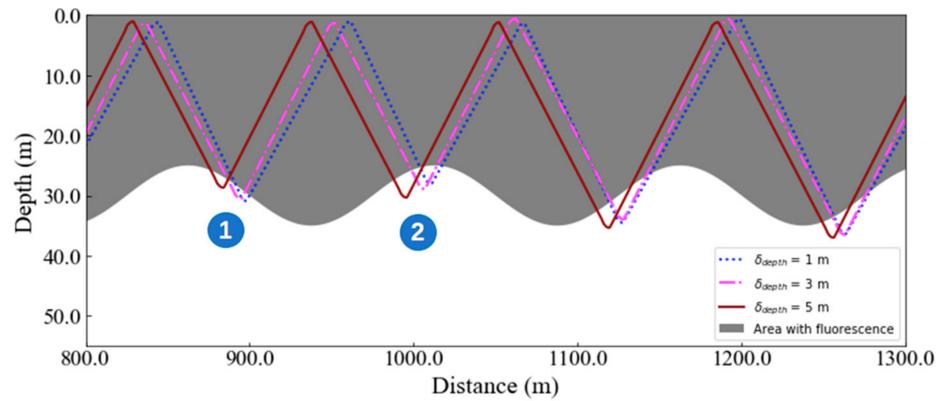


Figure 12. The depth of the glider under the control of a depth controller, which changed the maximum dive depth according to a simplified fluorescence field when δ_{depth} was set as 1 m, 3 m, and 5 m, respectively.

3.2. Adaptive Heading Changing

Adaptive depth changing enables the glider to change its depth adaptively in the water column, similarly, an adaptive heading change equips the glider with the autonomous capability to change its heading in the horizontal plane. With the ability to change heading adaptively, the glider can be set on a mission to delineate the boundary of ocean features (see Figure 13), track the source of plumes, and so on.

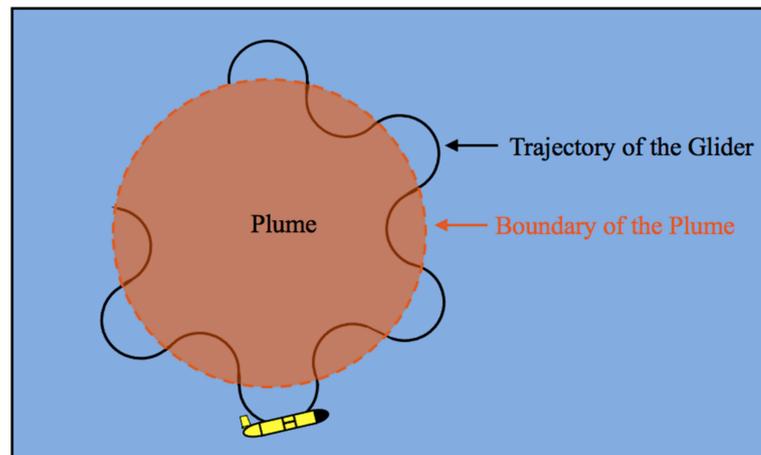


Figure 13. Adaptive changing of the heading of a glider to delineate the boundary of a plume based on the detection from sensors.

3.2.1. Following a Heading

In this simulation, the glider moved from point [47°24.2509' N, 53°08.1370' W] with an initial heading set to 0° (i.e., heading to the north). A command ($u_{mission_param_b}$) was sent from the heading controller in the backseat driver (Figure 14) after initiation to the main vehicle controller to steer the glider to a 30° heading, as shown in Figure 15.



Figure 14. Exchange of data between the main vehicle control system and a heading controller when the glider followed a heading commanded by the backseat driver.

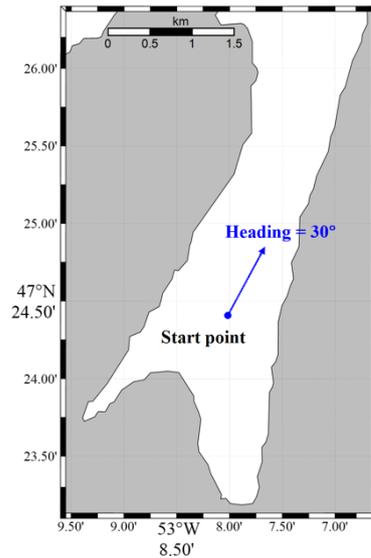


Figure 15. Trajectory of the glider during a mission when following a constant heading of 30°.

Results from the simulation, as shown in Figure 16, showed that the glider could change its heading from 0° to 30° after receiving the command from the heading controller. There was a discrepancy ranging from -6° to 6° between the commanded heading ($C_{heading}$) from the backseat driver and the measured heading ($m_{heading}$) of the glider. However, the glider kept a constant heading in general. This error could be minimized by changing the deadband for heading error in the mission files. Figure 17 shows another simulation when the deadband for heading error was set to 0°. The measured heading coincided with the commanded heading after 200 s when the heading of the glider became stable.

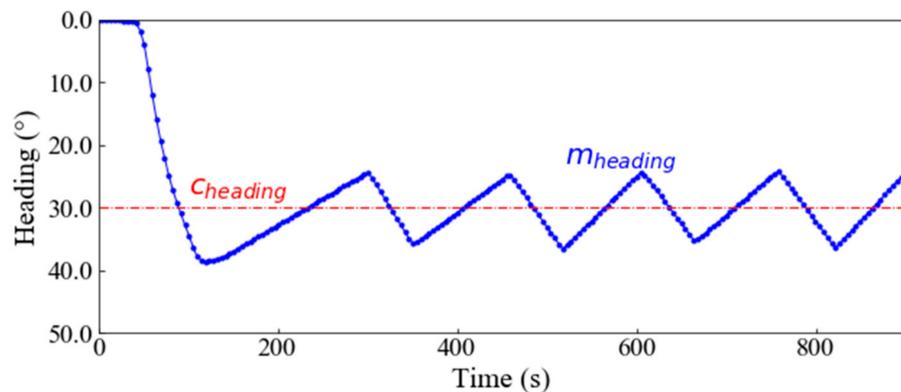


Figure 16. The heading of the glider under the control of a heading controller, which maintained the glider to a desired heading with a deadband setting of 5°.

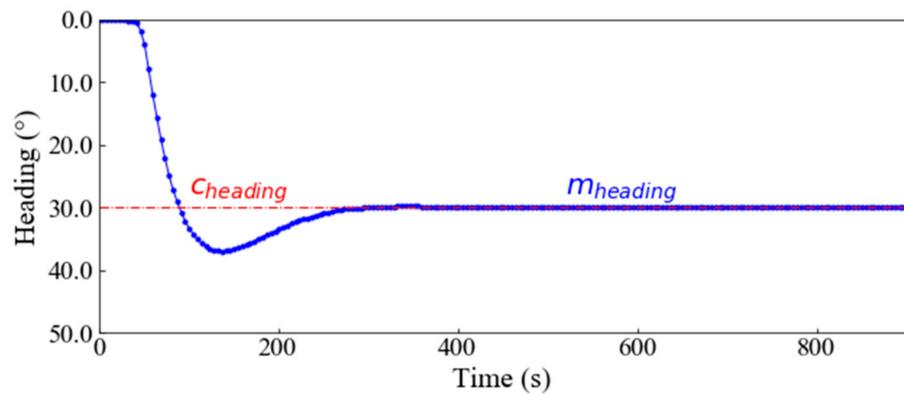


Figure 17. The heading of the glider under the control of a heading controller, which maintained the glider to a desired heading with a deadband setting of 0°.

3.2.2. Changing Heading along the Boundary of a Simulated Plume

The glider was operated to follow the boundary of a simulated plume. The heading of the glider was changed from that specified at the start point at [47°23.9304' N, 53°07.8320' W] in this simulation to test the adaptive heading behavior of the glider in boundary tracking. A simulated plume was created to have a circular boundary, with a center at [47°24.0517' N, 53°07.8320' W] and a radius of 100 m. The frontseat driver sent latitude (m_{lat}), longitude (m_{lon}), and heading ($m_{heading}$) information of the glider to a heading controller (see Figure 18). This heading controller adopted the location information of the glider and determined whether the glider was within the simulated plume or not. When the glider was inside the plume, the glider was requested to turn to starboard by increasing its heading to go toward the boundary of the plume. Otherwise, the glider turned to port to go into the plume. The pseudo code for control of the heading of the glider in this simulation is presented in Table 4.



Figure 18. Exchange of data between the main vehicle control system and the heading controller when the glider changed its heading based on detection of a plume from a fluorometer measurement.

Table 4. Control algorithm of the heading controller for backseat control when the glider changed its heading along the boundary of a simulated plume.

Pseudo code for the Heading Controller

```
# The location of the center of the simulated plume is [mlat_center, mlon_center]
# Δheading is a desired change of heading which is positive
Node Heading Controller subscribes latitude (mlat), longitude (mlon), and heading (mheading) of the glider
from the main vehicle control system
# dis is the distance that between the location of the glider to the center of the simulated plume
dis = the distance between [mlat, mlon] and [mlat_center, mlon_center]
# umission_param_b is the desired heading
if dis < radius
    umission_param_b = mheading + Δheading
else
    umission_param_b = mheading - Δheading
end if
publish umission_param_b to the Node Main Vehicle Control System
```

The trajectory of the glider in the horizontal plane under the control of the backseat driver for tracking the boundary of a simulated plume is shown in Figure 19. The glider could follow the plume boundary, but not precisely. For example, the boundary of the plume within the region labelled by 1 in Figure 19 was not tracked. This was caused by the tracking algorithm in the backseat driver. On the other hand, the performance of the glider in an adaptive heading behavior such as the response time and the ability to change heading according to the measurement from sensors was limited by the turning radius of the glider. When the turning radius is larger than the characteristics of an ocean feature such as the size of the plume, the glider will not be able to follow the boundary precisely. The minimum turning radius of this Slocum G1 glider was measured from simulation by publishing desired headings from 0 to 2π to the glider to make it change its heading continuously. These values of desired heading would drive the glider to turn to starboard. The trajectory of the glider in this simulation is shown in Figure 20, with a minimum turning radius measured, which was approximately 17 m.

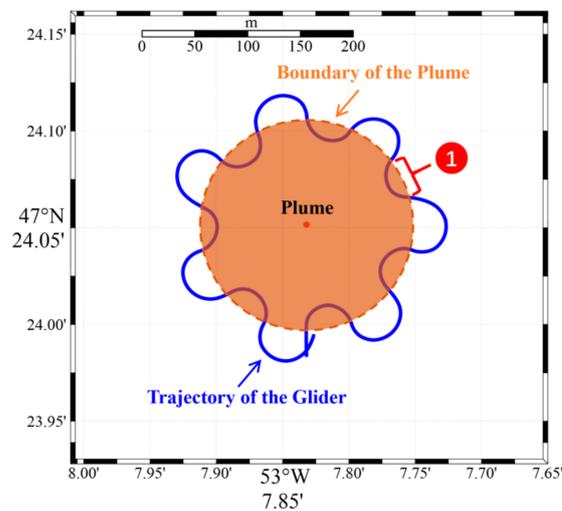


Figure 19. Trajectory of the glider in the horizontal plane under the backseat control of a heading controller, which changed the heading along the boundary of a simulated plume.

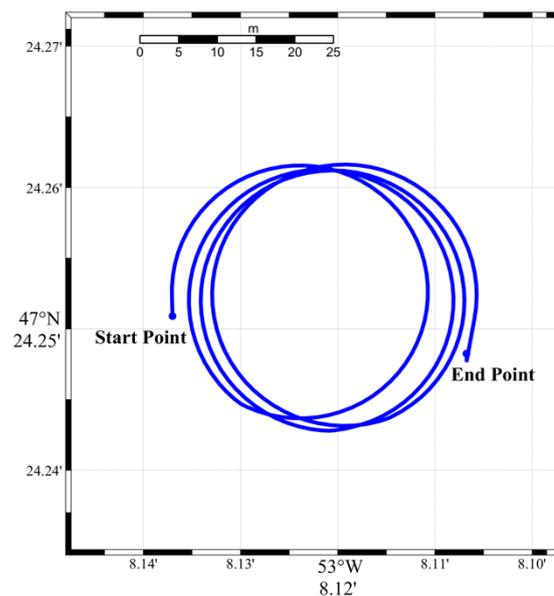


Figure 20. The trajectory of the glider when changing its heading from 0 to 2π continuously for estimating the turning radius of the Slocum G1 glider.

3.3. Adaptively Going to Waypoints

The glider can go to a waypoint or a list of waypoints by providing the location of the waypoints in the mission files in advance. This limits the trajectory of the glider. With the backseat driver, a new target point can be replanned for the glider when it is conducting a mission. The adaptive behavior to move the glider to a new waypoint is to set a heading for the glider toward the new waypoint and stop the waypoint behavior when the glider is within a specified distance from the target point. This adaptive waypoint behavior is not explicitly defined through the backseat driver and is realized by the behavior of the adaptive heading change, as mentioned in Section 3.2.

In the simulation, the glider was commanded to go from the start point to a series of waypoints (Table 5) with a maximum diving depth of 30 m. The locations of these target waypoints were not defined in the mission files, but by the backseat driver. The backseat driver translated the locations of the waypoints into the different headings needed to be followed by the glider and instructed the glider to follow the appropriate trajectory. The first heading was sent to the main vehicle controller to drive the glider in the direction of the first waypoint (Wpt 1 in Table 5). The backseat driver then kept calculating the distance of the glider from the first waypoint. When the distance was less than 10 m, the glider was instructed to go to the second waypoint by providing it with another desired heading. In addition, when the end point was reached, the glider was commanded to activate the surface behavior to terminate its mission by publishing $u_{mission_mode}$ to the main vehicle control system (Figure 21).

Table 5. Waypoint list for the adaptive waypoint behavior of the glider.

Mission Points	Coordinate
Start Point	[47°24.3704' N, 53°07.9213' W]
Wpt 1	[47°24.4511' N, 53°07.7041' W]
Wpt 2	[47°24.5153' N, 53°07.9173' W]
End Point	[47°24.5844' N, 53°07.7081' W]



Figure 21. Exchange of data between the main vehicle control system and the waypoint controller when the glider changed its target points adaptively.

The simulation result of this adaptive waypoint behavior is shown in Figure 22, which presents the trajectory of the glider when it was under the multi-waypoint mission control. The deadband for heading error in this simulation was 5°, and the glider could keep a constant heading in general, which instructed the glider to go to the targeted waypoints. Figure 23 presents the depth of the glider in this mission, which indicates that the glider surfaced after the end point was reached.

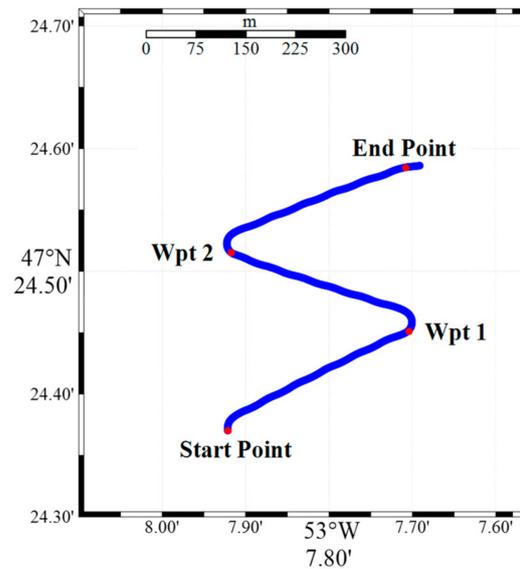


Figure 22. Trajectory of the glider in the mission testing the adaptive waypoint behavior.

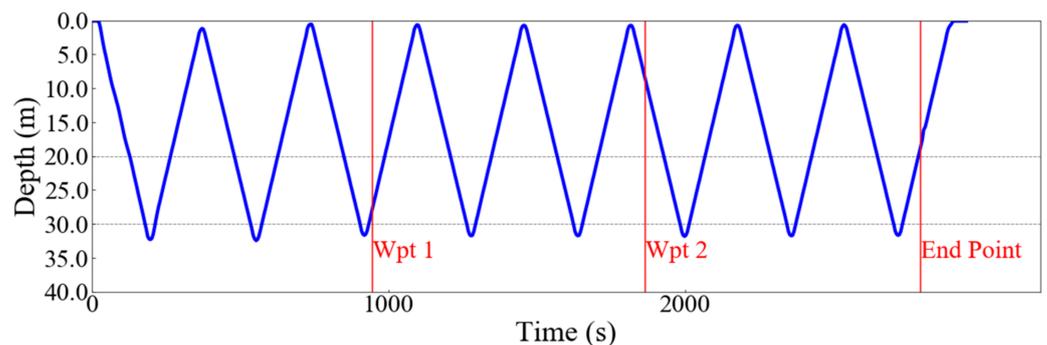


Figure 23. The depth of the glider under the control of the waypoint controller, which changed the target point adaptively and activated the surface behavior when the end point was reached.

4. Discussion

The existing backseat driver for this Slocum G1 glider was running on ROS, with a ROS node as an interface between the main vehicle control system and the backseat driver and other ROS nodes as parts of the backseat driver for controlling sensors and implementing adaptive control algorithms. However, the glider is not limited in using ROS as the platform for the backseat driver for the adaptive control algorithms and sensor controllers. With the ROS interface, there are various software packages that can be incorporated with our existing system for future development such as MOOS-IvP and LCM, or even ROS that was developed for use in other fields.

This work was mainly to present the control structure of the backseat driver for the Slocum G1 glider, with some examples showing adaptive behaviors with the backseat controllers. Simulations in this paper were limited to one ROS node in the backseat driver, which subscribed state information of the glider from the main vehicle control system and published desired states to the glider. Measurements from sensors were simplified and simulated within the ROS node. For field experiments with the backseat driver, the backseat driver can be provided with measurements from real sensors. For instance, for a glider with a fluorometer in field missions, a fluorometer controller can be used by the backseat driver to process measurements from the fluorometer, as shown in Figure 24. The processed fluorescence data will be sent to the depth controller to decide the desired depth of a yo behavior, with the desired depth published to the main vehicle control system to realize adaptive depth control. This control structure can also be applied to the

adaptive heading control of the glider in field missions, for example, including real-time measurements from an on-board fluorometer (see Figure 25).

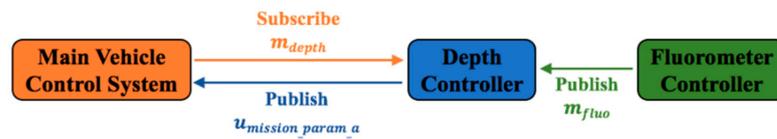


Figure 24. The control structure of the glider with a backseat driver, which takes the measurements from a fluorometer for adaptive depth control.



Figure 25. The control structure of the glider with a backseat driver, which takes the measurements from a fluorometer for adaptive heading control.

With the incorporation of measurements from sensors in field work, the glider autonomously responds to the environment and the path of the glider is unspecified. Besides, as the operating speed of the Slocum glider is 0.4 m/s, the motion of the glider is subject to environmental disturbances such as currents. If the current speed is high, the glider may drift from its designed path and fail to complete its mission. The safety of the glider is guaranteed by safety rules in the dynamic environment. Safety rules can be implemented through both the mission file and the backseat driver control algorithm.

- Implementation through the mission file. For example, a maximum mission time can be defined in the mission file to restrict the duration of a mission. The glider will abort its mission and climb to the surface if it has worked in excess of the maximum mission time. For a yo behavior, a target altitude (from the seabed) can be defined in the mission file. The target altitude will command the glider to abort the diving behavior and climb if the altitude of the glider is smaller than the target altitude, even if the glider has not reached the target diving depth.
- Implementation through the backseat driver. A safety zone can be defined in the backseat driver to specify the region of operation for the glider. The backseat driver subscribes the location and depth information of the glider from the main vehicle controller and determines whether the glider is within the safety zone or not. If the glider is outside of this zone, the backseat driver will instruct the glider to either go back to this safety zone or abort its mission.

The developed backseat driver will be tested in field experiments with measurements from sensors included in the control structure of the backseat driver.

5. Conclusions

The main focus of this paper was to interface external codes with a Slocum glider. This backseat driver communicates with the main vehicle control system through an interface developed by using the Robot Operating System. With this backseat driver, state information of the glider and measurements from sensors can be subscribed by backseat control algorithms, with desired state parameters being sent to the main vehicle control system for adaptive control. A series of simulations were conducted with this Slocum G1 glider to test the performance of this backseat driver in adaptively controlling the depth, heading, waypoints, and behavior state of the glider. Performance of the backseat control from these simulations reveals its ability to control the glider adaptively and in real time.

In the adaptive depth changing mission, the depth controller in the backseat driver was insensitive to the value of δ_{depth} , which defines distance from the next target depth, even at a small value of 1 m. This reviewed that the backseat driver could receive states

of the glider from the main vehicle control system and control the depth of the glider in a timely manner. In the mission of changing the heading of the glider to track the boundary of a simulated plume in the horizontal plane, the performance of boundary following of the glider was affected by its turning radius. The minimum turning radius of this Slocum G1 glider was approximately 17 m, measured from a simulation in this work. When precisely tracking an ocean feature with this glider, the relative size between the ocean feature and the minimum turning radius of the glider should be taken into consideration to plan the path of the glider.

This developed backseat driver will be tested in field experiments in the near future, with sensors and safety rules included in the mission files and control algorithms. The sensors will be used to provide real-time measurements to the backseat driver. With the ability of successful adaptive control, the glider will be able to do intelligent missions and improve its efficiency in sampling interesting targets.

Author Contributions: Conceptualization, Y.W., C.B. and N.B.; Methodology, Y.W. and C.B.; Software, Y.W. and C.B.; Validation, Y.W. and C.B.; Formal analysis, Y.W., C.B., W.T. and N.B.; Investigation, Y.W. and C.B.; Resources, N.B., Y.W., C.B. and W.T.; Data curation, Y.W.; Writing—original draft preparation, Y.W., W.T. and N.B.; Writing—review and editing, Y.W., W.T. and N.B.; Visualization, Y.W.; Supervision, N.B. and W.T.; Project administration, N.B. and W.T.; Funding acquisition, N.B. All authors have read and agreed to the published version of the manuscript.

Funding: Funding was received from Fisheries and Oceans Canada through the Multi-Partner Oil Spill Research Initiative (MPRI) 1.03: Oil Spill Reconnaissance and Delineation through Robotic Autonomous Underwater Vehicle Technology in Open and Iced Waters and the School of Graduate Studies at Memorial University of Newfoundland.

Acknowledgments: The authors would like to thank Teledyne Webb Research for assisting in developing the backseat driver capability interface.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Graver, J.G.; Bachmayer, R.; Leonard, N.E.; Fratantoni, D.M. Underwater glider model parameter identification. In Proceedings of the 13th International Symposium on Unmanned Untethered Submersible Technology (UUST), Durham, NH, USA, 1–5 August 2003; pp. 1–12.
2. Todd, R.E.; Rudnick, D.L.; Sherman, J.T.; Owens, W.B.; George, L. Absolute Velocity Estimates from Autonomous Underwater Gliders Equipped with Doppler Current Profilers. *J. Atmos. Ocean. Technol.* **2017**, *34*, 309–333. [CrossRef]
3. Cetinić, I.; Toro-Farmer, G.; Ragan, M.; Oberg, C.; Jones, B.H. Calibration procedure for Slocum glider deployed optical instruments. *Opt. Express* **2009**, *17*, 15420–15430. [CrossRef] [PubMed]
4. Zhou, M.; Bachmayer, R.; DeYoung, B. Mapping the underside of an iceberg with a modified underwater glider. *J. Field Robot.* **2019**, *36*, 1102–1117. [CrossRef]
5. Mansour, A.; Leblond, I.; Hamad, D.; Artigas, L.F. Sensor Networks for Underwater Ecosystem Monitoring and Port Surveillance Systems. In *Sensor Networks for Sustainable Development*; CRC Press: Boca Raton, FL, USA, 2014; pp. 431–467.
6. Domingues, R.; Kuwano-Yoshida, A.; Chardon-Maldonado, P.; Todd, R.E.; Halliwell, G.R.; Kim, H.S.; Lin, I.I.; Sato, K.; Narazaki, T.; Shay, L.K.; et al. Ocean observations in support of studies and forecasts of tropical and extratropical cyclones. *Front. Mar. Sci.* **2019**, *6*. [CrossRef]
7. Dent, S. Fleet of Underwater Gliders Could Improve Global Weather Forecasts. Available online: https://www.engadget.com/2014/03/17/rutgers-challenger-glider-submarine-mission/?ncid=rss_truncated (accessed on 21 September 2020).
8. Alvarez, A.; Chiggiato, J.; Schroeder, K. Mapping sub-surface geostrophic currents from altimetry and a fleet of gliders. *Deep Sea Res. Part I Oceanogr. Res. Pap.* **2013**, *74*, 115–129. [CrossRef]
9. Dhont, D.; Jatiault, R.; Lattes, P. Combined Use of SAR and Underwater Gliders for Oil Seeps Detection. In Proceedings of the International Geoscience and Remote Sensing Symposium (IGARSS), Yokohama, Japan, 28 July–2 August 2019; pp. 7865–7868.
10. Fiorelli, E.; Bhatta, P.; Leonard, N.E.; Shulman, I. Adaptive sampling using feedback control of an autonomous underwater glider fleet. In Proceedings of the 13th International Symposium on Unmanned Untethered Submersible Technology (UUST), Durham, NH, USA, 1–5 August 2003; pp. 1–16.
11. Stommel, H. The Slocum Mission. *Oceanography* **1989**, *2*, 22–25. [CrossRef]

12. Mavrommati, A.; Tzorakoleftherakis, E.; Abraham, I.; Murphey, T.D. Real-Time Area Coverage and Target Localization using Receding-Horizon Ergodic Exploration. *IEEE Trans. Robot.* **2017**, *34*, 62–80. [[CrossRef](#)]
13. Ivic, S.; Crnkovic, B.; Mezić, I. Ergodicity-Based Cooperative Multiagent Area Coverage via a Potential Field. *IEEE Trans. Cybern.* **2017**, *47*, 1983–1993. [[CrossRef](#)] [[PubMed](#)]
14. Ferri, G.; Cococcioni, M.; Alvarez, A. Mission planning and decision support for underwater glider networks: A sampling on-demand approach. *Sensors* **2016**, *16*, 28. [[CrossRef](#)] [[PubMed](#)]
15. Naglak, J.E.; Page, B.R.; Mahmoudian, N. Backseat Control of SandShark AUV using ROS on RaspberryPi*. In Proceedings of the OCEANS 2018 MTS/IEEE Charleston, Charleston, SC, USA, 22–25 October 2018; pp. 1–5.
16. Page, B.R.; Ziaefard, S.; Pinar, A.J.; Mahmoudian, N. Highly Maneuverable Low-Cost Underwater Glider: Design and Development. *IEEE Robot. Autom. Lett.* **2017**, *2*, 344–349. [[CrossRef](#)]
17. Gallimore, E.; Stokey, R.; Terrill, E. Robot Operating System (ROS) on the REMUS AUV using RECON. In Proceedings of the 2018 IEEE/OES Autonomous Underwater Vehicle Workshop, Porto, Portugal, 6–9 November 2018; pp. 1–6.
18. Eickstedt, D.P.; Sideleau, S.R. The backseat control architecture for autonomous robotic vehicles: A case study with the Iver2 AUV. *Mar. Technol. Soc. J.* **2010**, *44*, 42–54. [[CrossRef](#)]
19. Benjamin, M.R.; Schmidt, H.; Newman, P.M.; Leonard, J.J. Nested autonomy for unmanned marine vehicles with MOOS-IvP. *J. Field Robot.* **2010**, *27*, 834–875. [[CrossRef](#)]
20. Bluefin Robotics Bluefin and MIT Demonstrate AUV Plug-n-Play Payload Autonomy. Available online: <https://defense-unmanned.com/article/1056/bluefin-mit-demo-auv-payload-autonomy.html> (accessed on 9 February 2021).
21. Keane, J.R.; Forrest, A.L.; Johannsson, H.; Battle, D. Autonomous Underwater Vehicle Homing with a Single Range-Only Beacon. *IEEE J. Ocean. Eng.* **2020**, *45*, 395–403. [[CrossRef](#)]
22. Ivić, S.; Crnković, B.; Arbabi, H.; Loire, S.; Clary, P.; Mezić, I. Search strategy in a complex and dynamic environment: The MH370 case. *Sci. Rep.* **2020**, *10*. [[CrossRef](#)] [[PubMed](#)]
23. Meurer, C.; Fuentes-Perez, J.F.; Schwarzwald, K.; Ludvigsen, M.; Sorensen, A.J.; Kruusmaa, M. 2D estimation of velocity relative to water and tidal currents based on differential pressure for autonomous underwater vehicles. *IEEE Robot. Autom. Lett.* **2020**, *5*, 3444–3451. [[CrossRef](#)]
24. Kim, E.; Fan, S.; Bose, N.; Nguyen, H. Current Estimation and Path Following for an Autonomous Underwater Vehicle (AUV) by Using a High-gain Observer Based on an AUV Dynamic Model. *Int. J. Control Autom. Syst.* **2021**, *19*, 478–490. [[CrossRef](#)]
25. Teledyne Webb Research Slocum G1 Glider Manual. Available online: <https://manualzz.com/doc/6922727/slocum-g1-glider-manual> (accessed on 18 April 2020).
26. Ros.org ROS: Powering the World’s Robots. Available online: <https://www.ros.org/> (accessed on 9 February 2021).
27. BeagleBoard.org BeagleBone Black. Available online: <https://beagleboard.org/black> (accessed on 9 February 2021).
28. Zhou, M. The Approach of Improving the Roll Control of a Slocum Autonomous Underwater Glider. Master’s Thesis, Memorial University of Newfoundland, St. John’s, NL, Canada, 2012.
29. Zhou, M.; Bachmayer, R.; Deyoung, B. Underwater acoustic-based navigation towards multi-vehicle operation and adaptive oceanographic sampling. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Vancouver, BC, Canada, 24–28 September 2017; pp. 6091–6097.
30. Zhou, M.; Bachmayer, R. Working towards single beam acoustic iceberg profiling using active roll control on a Slocum glider. In Proceedings of the 2011 IEEE Symposium on Underwater Technology, UT’11 and Workshop on Scientific Use of Submarine Cables and Related Technologies, Tokyo, Japan, 5–8 April 2011; pp. 1–7.