Implementation of Near Real-Time Onboard Processing Software for a Slocum Glider Acoustic Doppler Current Profiler

Julia Engdahl
Center for Ocean Observing Leadership (COOL)
Rutgers University
New Brunswick, USA
engdahl@marine.rutgers.edu

Joseph Gradone
Center for Ocean Observing Leadership (COOL)
Rutgers University
New Brunswick, USA
jgradone@marine.rutgers.edu

Zachary Silva
Teledyne Webb Research
Falmouth, USA
zachary.silva@teledyne.com

David Aragon
Center for Ocean Observing Leadership (COOL)
Rutgers University
New Brunswick, USA
dkaragon@marine.rutgers.edu

Nikitha Shivakumar
College of Computing
Georgia Institute of Technology
Atlanta, USA
nshivakumar6@gatech.edu

Travis Miles
Center for Ocean Observing Leadership (COOL)
Rutgers University
New Brunswick, USA
tnmiles@marine.rutgers.edu

***Abstract*—Advancements in sensor and battery technology over the past two decades have facilitated the integration of complex instruments into autonomous underwater vehicles. The integration of such instruments onto gliders has enabled unprecedented data collection and expanded understanding of ocean processes. Recently, this understanding has been enhanced through the development of a software architecture called "backseat driver" on Slocum gliders, which can increase the efficiency of data collection by increasing the time the glider spends in regions of interest through adaptive mission control. A caveat to this advancement is that sensors that collect large volumes of data cannot efficiently transmit data to shore in real-time, which limits real-time applications (e.g. operational hurricane forecasting, marine mammal detection, water quality monitoring, etc.) and confines shoreside processing to post recovery. This can potentially lead to loss of data in the event of vehicle loss or missions where retrieval is not permissible, resulting in the need to develop a sensor agnostic real-time processing software and hardware architecture. Here, we detail the methodology that will be applied to test a real-time onboard processing algorithm through the use of the backseat driver software on Slocum glider MARACOOS04. An adjunct processor (Raspberry Pi 4) has been integrated into this glider with a data processing algorithm that will parse, process, and compress raw data from an acoustic Doppler current profiler.**

Keywords—Acoustic Doppler Current Profiler, Slocum Glider, Backseat Driver, Real-Time Onboard Processing

I. Introduction

Over the past two decades, advancements in sensor and battery technology have allowed for the integration of a variety of complex sensors into autonomous underwater vehicles (AUV) including but not limited to acoustic doppler current profilers (ADCPs), multi-frequency echo sounders, optical laser diffraction sensors, wave accelerometers, passive acoustics, and turbulence [1]. While integrating these sensors onto platforms such as gliders has led to unprecedented data collection and further understanding of ocean processes, advancement in ocean exploration can be further expanded through the use of a software capability referred to as "backseat driver". This software can increase the glider's working time in regions of high interest through adaptive mission control, which allows the vehicle to change its state (e.g., depth, heading, waypoint, sampling, etc.) based on real-time sensor measurements throughout the duration of the mission [2-3]. Software capabilities like backseat driver are increasingly becoming a necessity as missions become more complex in extreme ocean environments [2,4], and have been implemented on numerous underwater vehicles such as Bluefin SandShark AUV [5], Hydroid REMUS 100 AUV [6], Iver2 AUV [7], Teledyne Gavia AUV [8], Teledyne Webb Research Slocum gliders and many more [2].

The large data volume collected by these sensors during complex missions cannot be transferred to shore efficiently in real-time. As a result, shoreside processing has largely been confined to post recovery, significantly limiting real-time applications. In the event of vehicle loss or missions where retrieval is not immediately permissible, this can lead to an unfortunate loss of data and time. Real-time data processing software is increasingly becoming more important because it can aid in management and scientific applications by increasing data accuracy, preventing data loss, and allowing for adaptive surveys [9]. Recently, some sensors such as the digital acoustic monitoring (DMON) instrument [9], Sequoia Scientific Laser In Situ Scattering and Transmissometry (LISST) sensor [10], and the Fluorescence Induction and Relaxation System (FIRE) [11] have developed self-contained processing algorithms, but they

Funding provided by the Office of Naval Research Project Grant N000142012568.

are proprietary to their systems, limited to their internal processing capabilities, and are not easily modifiable [1]. Thus, the development of a sensor agnostic real-time data processing software and hardware architecture is needed for autonomous underwater vehicles.

Although the backseat driver is currently used to autonomously and adaptively modify glider behavior while in mission to efficiently collect data in regions of interest [2], the data processing algorithms are presently only used for adaptive mission control. The resulting data that is telemetered to shore still has to undergo shoreside processing post recovery. We are presently utilizing this software to expand glider real-time data processing capabilities. Here, we detail initial real-time processing experiments focused on ADCP processing via a widely used linear inversion algorithm, first tailored for ship-based surveys [12], then modified for glider platforms to generate depth-binned, absolute ocean velocities [13].

II. Methods

A. Glider

Underwater gliders have been used to monitor and collect data to gain useful information about oceanographic processes [14]. They are considered to be long endurance, low energy, and relatively low cost vehicles [2,14]. Gliders are mobile sensor platforms that are able to collect spatial and temporal data throughout the water column. These vehicles use a change in buoyancy and a set pitch angle to propel itself vertically and horizontally in a sawtooth-like pattern [14-15]. Gliders can be operated in water depths from 5 m to 1000 m and will have a vertical speed of approximately 20 cm/s [14-15]. During glider dives and climbs, the data is sampled every 2 seconds, which results in a robust dataset that can be telemetered to shore via an Iridium satellite phone after decimation [15].

The glider used in this study is a Teledyne Webb Research Generation 3 Slocum glider, MARACOOS04 (MARA04), operated by two STM32 120 MHz processors. This glider is equipped with a standard oceanographic sensor loadout of SeaBird CTD, a 100 m depth-rated buoyancy pump,Wetlabs 3 channel optical sensor, and a 600 kHz Teledyne RD Instruments Pathfinder ADCP. The Pathfinder utilizes a phased array transducer mounted flush to the glider hull with 4-beams in a 30° Janus configuration. Beams 1 and 3 are forward facing and beams 2 and 4 are aft-facing at 22.5° off the glider's center to both starboard and port sides, respectively. The output format of the raw data is configured to be PD0, which is Teledyne RD Instrument's standard binary format. PD0's from an approximately 3-hour long glider segment (time spent underwater between surfacings) result in a file size on the order of thousands of kilobytes which are too large to transfer over an Iridium connection in a timely and cost-effective manner and can only be downloaded post-deployment.

B. Backseat Driver

Prior to the development and implementation of the backseat driver, gliders did not have the capability of adaptive mission control onboard. The pilot sets the initial parameters including but not limited to heading, pitch, target dive depth, and waypoints. These engineering and flight specifications follow the blue (left) pathway shown in Fig. 1A where these parameters are input into the flight computer so that internal calculations can be computed in order to adjust commanded behaviors. The flight computer creates a decimated file (SBD) that is a subset of raw data to be transmitted to shore. Simultaneously, science sensors follow the pink (right) pathway in Fig. 1B where the data collected is then communicated to the science computer to create its decimated file (TBD) that is used to be transmitted onshore. Although the flight and science computers are communicating throughout the duration of the mission, the engineering and flight inputs cannot be dynamically updated. These inputs have to be manually updated by pilots when the glider reaches the surface and establishes communications over iridium.

The development of the backseat driver system by Teledyne Webb Research enhances the glider operation cycle immensely. Gliders equipped with backseat driver follow the same flight and science pathways as Fig. 1A with an additional external processor integrated into the science computer (Fig. 1B). This external processor can receive engineering and science sensor data that enables the system to adaptively update mission parameters. To accomplish the sensor agnostic real-time processing scheme, the external controller is able to receive science sensor data that is both logged to the TBD and data that is logged to a sensor's native data file format. This data is then passed to the external controller, which runs an open-source python processing algorithm (available here: https://github.com/JGradone/Glider_ADCP_Real_Time_Processing) and saves the output to a netCDF file that is able to be efficiently telemetered to shore.

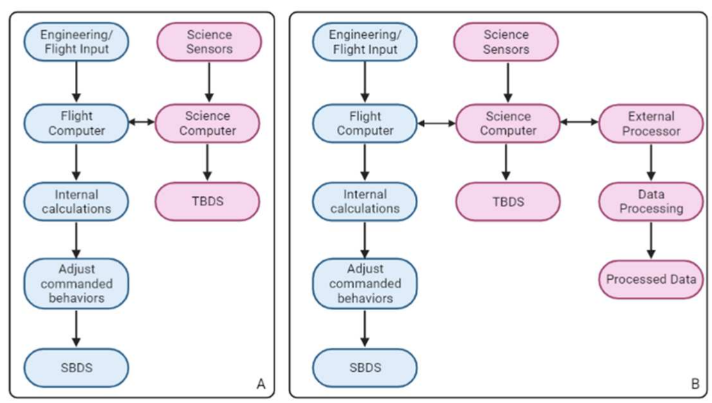

Fig 1. Glider operation cycle (a) without backseat driver and (b) with backseat driver.

a) External Processor Hardware: We are utilizing a Raspberry Pi 4 Model B because it has both sufficient computational power to execute real-time data processing and takes up minimal space within the payload [5]. A Raspberry Pi is a single board computer that is equipped with a 32-bit quad core processor running at 1.8 GHz, standard 40 pin GPIO header, Micro-SD card slot, 2 GB of RAM, 2 USB 3.0 ports, 2 USB 2.0 ports. The Raspberry Pi is being powered through its

5 V USB-C connector by the glider science computer through a USB to serial connector that is equipped with a 12 V to 5 V regulator. This connector utilizes the FTDI chipset and allows communication between the science computer and the Raspberry Pi using RS232 protocol. The Raspberry Pi is running Raspberry Pi OS, a Debian based operating system that has a Linux kernel. Additionally, Python version 3.9.2 is installed which is used to run the external controller script and the linear inversion algorithm. The Raspberry Pi is mounted in the forward section of the glider science payload on the starboard support bar (Fig. 2). The mount pictured, is a UCTRONICS DIN rail mount and is installed with the mounting bracket facing the inside of the payload and the base is fastened to the starboard support bar. Because the mount is made out of 1.2 mm carbon steel, we placed a piece of plastic between the base and the Raspberry Pi so that the pins of the Raspberry Pi do not touch the base of the mount. Additionally, we drilled a large hole into the mounting bracket of the mount so that we can access the power port. The Raspberry Pi is integrated into the science computer using the USB to Serial connector mentioned above. The backseat driver recognizes it through the proglet, but the current version does not include the use of a sensor sample behavior argument to control the power of the Raspberry Pi. This can prove problematic because that means there is a constant power draw throughout the duration of the mission. To address this issue, a new version of backseat driver is under development that will enable sensor sample behavior control over the Raspberry Pi and increase power efficiency for the duration of the mission.

Fig. 2. Raspberry Pi 4 Model B mounted inside glider science payload.

b) External Controller Software: The external controller software is able to communicate with the glider's flight computer through the backseat driver, a software developed by Teledyne Webb Research. This software is able to subscribe state parameters to the backseat driver which enables the subscribed parameter to be processed by an external controller. The external controller can both modify and update the result back to the glider to update desired parameters. In order to facilitate near real-time data transfer, we have developed an enhanced external controller that is a modified version of [1] onboard processing application. When the .PD0 file is read into the script, standard quality control methods are performed to optimize data quality for the final velocity profile derivation [1,16-17]. Data below a correlation threshold of 50%, an echo intensity of 70 dB, and percent good of 80% are discarded. Glider pitch and roll as well as individual beam angles and directions are used to map beam velocity data to depth cells and then transformed to level true depth. Following this transformation, additional QC is performed, where first measurement cells found to be below the seafloor are removed. Velocities relative to the glider exceeding 75 cm/s are also removed [13]. Then, data from the velocity bin closest to the glider are removed to avoid ringing and other contamination and data shallower than 5 m are removed to ensure the glider has achieved a state more typical of glider flight behavior for valid measurements [18]. Lastly, the data goes through a final check to ensure that the inversion can be completed. If the data fails any of the tests included in the final check, the inversion cannot be performed, otherwise the algorithm will continue to the inversion portion. This check includes three steps consisting of a depth range check, missing data check, and if there are two or more non-nan data points. The depth range check looks to see if there is a change in depth. If all depth values are the same, for example 0 m, the error flag is set to 1. The missing data check looks to see if all data is missing, and if it is, the error flag is set to 2. If there are less than two existing data points, then the error flag is set to 3, or if there are two or more existing data points the error flag is set to 4. Two is chosen for the purpose of receiving all possible processed data even if it produces a poor inversion, and may change as testing continues. Once the data has gone through the quality control steps, the data is fed through the linear inversion algorithm that was first developed for ship-based observations [12], modified by [13] to calculate depth-binned absolute ocean velocities on the Spray glider platform, and further modified by [1] for the Slocum glider platform.

III. Testing

To test and optimize the ability of the backseat driver software to initiate real-time processing of the raw data generated from the integrated ADCP. The backseat driver will be implemented via an external controller script, written in Python on the onboard Raspberry Pi computer, which will act as the interface between the backseat driver software and the science computer on the glider. Because it is important to limit the time the glider spends at the surface processing and transmitting data, we are developing precise software triggers to enable the glider to initiate the real-time processing workflow as quickly as possible. The plan is to have the ADCP programmed to sample during dives, climbs, and while hovering (state_to_sample = 7, Fig. 3A). When the glider enters the surface state (m_depth < u_reqd_depth_at_surface), the ADCP closes the .PD0 file and the external controller script is then initiated (Fig. 3B). This enables the processing to begin and, after optimization, finish before the glider is in a state to send data to shore via Iridium. Due to idiosyncrasies of the glider software, we have designed the external controller to start based on the following triggers:

1) ADCP is told to stop samping (c_dvl_on = -1*)*.

2) Glider is physically at the surface (m_depth < u_reqd_depth_at_surface, typically 3 m for shallow gliders and 7 m for deep gliders)

3) Glider's state changes to be at the surface, indicating state_to_sample =7 is no longer fullfilled and the ADCP is no longer sampling (cc_final_behavior_state = 5).

By design, there is overlap in the manner in which these sensors change their state, leading to redundancy in the trigger checks. The external controller script sends the .PD0 from the science computer to the Raspberry Pi and executes the onboard processing linear inversion algorithm, developed by Gradone et al. (2021) (Fig. 3C). This algorithm solves a system of equations for absolute horizontal water velocities using least squares techniques. The resulting velocity profile is exported into a network Common Data Form (netCDF) file that the external controller then sends from the Raspberry Pi to the glider science computer, so that the data can be efficiently transferred onshore in near real-time (Fig. 3C).

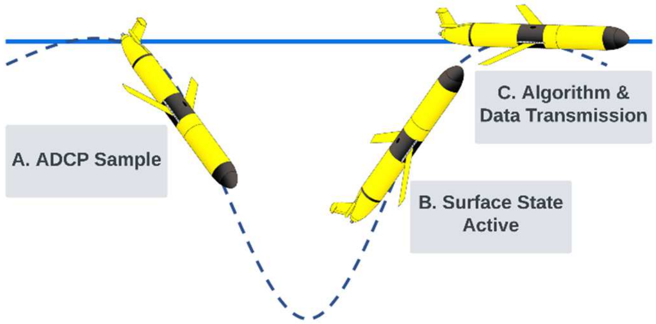

Fig 3. Step by step schematic of near real-time onboard processing.

Additional tests will be conducted to calculate computation time, optimal file size for iridium satellite transmission, and power draw in both simulation mode and the field. Considerations include algorithm constraints, smoothing values, quality control thresholds, and solution vertical resolution among other parameters. We tested the time it took for the Raspberry Pi to execute the linear inversion algorithm on 125 ADCP data files that were a mixture of 30m and 1000m max depth. These files were approximately 4 megabytes each but will be reduced to around 10 kilobytes, making them small enough to transfer over iridium. On average a glider spends 10 to 15 minutes at the surface, processing time would increase surface time by an average of 17.116 seconds, with a minimum of 0.642 seconds and maximum of 28.17 seconds. There will be a myriad of future real-time onboard processing applications once the Raspberry Pi-external controller data processing has been further integrated in the sensor agnostic manner as described here. We achieved our original goals with the resultant short processing time and small processed file sizes in the bench top environment. In the coming weeks we expect to demonstrate this capability in a field environment, including transfer of data to shore via Iridium communications.

ACKNOWLEDGMENT

We thank Rutgers University Center for Ocean Observing Leadership (RUCOOL) glider team for glider preparations. We also thank Eli Hunter and John Kerfoot (RUCOOL) for additional code development. We also acknowledge Teledyne Webb Research staff for technical assistance.

REFERENCES

[1] Gradone, J. C., Hunter, E. J., Mullison, J., & Miles, T. N.. (2021). Development of Onboard Processing Capabilities for a Slocum Glider Acoustic Doppler Current Profiler. https://doi.org/10.23919/oceans44145.2021.9705895

[2] Wang, Y., Bulger, C., Thanyamanta, W., & Bose, N.. (2021). A Backseat Control Architecture for a Slocum Glider. *Journal of Marine Science and Engineering*, *9*(5), 532. https://doi.org/10.3390/jmse9050532

[3] Ferri, G., Cococcioni, M., & Alvarez, A.. (2015). Mission Planning and Decision Support for Underwater Glider Networks: A Sampling on-Demand Approach. *Sensors*, *16*(1), 28. https://doi.org/10.3390/s16010028

[4] Hwang, J., Bose, N., & Fan, S.. (2019). AUV Adaptive Sampling Methods: A Review. *Applied Sciences*, *9*(15), 3145. https://doi.org/10.3390/app9153145

[5] Naglak, J. E., Page, B. R., & Mahmoudian, N.. (2018). Backseat Control of SandShark AUV using ROS on RaspberryPi. https://doi.org/10.1109/oceans.2018.8604630

[6] Gallimore, E., Stokey, R., & Terrill, E.. (2018). Robot Operating System (ROS) on the REMUS AUV using RECON. https://doi.org/10.1109/auv.2018.8729755

[7] Eickstedt, D. P., & Sideleau, S. R.. (2010). The Backseat Control Architecture for Autonomous Robotic Vehicles: A Case Study with the Iver2 AUV. *Marine Technology Society Journal*, *44*(4), 42–54. https://doi.org/10.4031/mtsj.44.4.1

[8] Keane, J. R., Forrest, A. L., Johannsson, H., & Battle, D.. (2020). Autonomous Underwater Vehicle Homing With a Single Range-Only Beacon. *IEEE Journal of Oceanic Engineering*, *45*(2), 395–403. https://doi.org/10.1109/joe.2018.2877535

[9] Baumgartner, M. F., Fratantoni, D. M., Hurst, T. P., Brown, M. W., Cole, T. V. N., Van Parijs, S. M., & Johnson, M.. (2013). Real-time reporting of baleen whale passive acoustic detections from ocean gliders. *The Journal of the Acoustical Society of America*, *134*(3), 1814–1823. https://doi.org/10.1121/1.4816406

[10] Miles, T. N., Kohut, J., Slade, W., & Gong, D.. (2018). Suspended particle characteristics from a glider integrated LISST sensor. https://doi.org/10.1109/oceans.2018.8604627

[11] Haskins, C., & Schofield, O.. (2015). Glider measurements of phytoplankton physiology in penguin foraging zones along the Western Antarctic Peninsula. https://doi.org/10.23919/oceans.2015.7404376

[12] Visbeck, M.. (2002). Deep Velocity Profiling Using Lowered Acoustic Doppler Current Profilers: Bottom Track and Inverse Solutions*. *Journal of Atmospheric and Oceanic Technology*, *19*(5), 794–807. https://doi.org/10.1175/1520-0426(2002)019<0794:dvpula>2.0.co;2

[13] Todd, R. E., Rudnick, D. L., Sherman, J. T., Owens, W. B., & George, L.. (2017). Absolute Velocity Estimates from Autonomous Underwater Gliders Equipped with Doppler Current Profilers. *Journal of Atmospheric and Oceanic Technology*, *34*(2), 309–333. https://doi.org/10.1175/jtech-d-16-0156.1

[14] Schofield, Kohut, J., Aragon, D., Creed, L., Graver, J., Haldeman, C., Kerfoot, J., Roarty, H., Jones, C., Webb, D., & Glenn, S.. (2007). Slocum Gliders: Robust and ready. *Journal of Field Robotics*, *24*(6), 473–485. https://doi.org/10.1002/rob.20200

[15] Miles, T., Seroka, G., Kohut, J., Schofield, O., & Glenn, S.. (2015). Glider observations and modeling of sediment transport in Hurricane Sandy. *Journal of Geophysical Research: Oceans*, *120*(3), 1771–1791. https://doi.org/10.1002/2014jc010474

[16] Taylor, J. A., & Jonas, A. M. (2008). Maximising data return: Towards a quality control strategy for managing and processing TRDI ADCP data sets from moored instrumentation. IEEE/OES 9th Working Conference

on Current Measurement Technology, 2008, 80– 88. https://doi.org/10.1109/CCM.2008.4480848

[17] Haines, S., Crout, R., Bosch, J., Burnett, W., Fredericks, J., Symonds, D., & Thomas, J. (2011). A summary of quality control tests for waves and in situ currents and their effectiveness. In IEEE/OES 10th Current, Waves and Turbulence Measurements (CWTM) (pp. 100– 106). https://doi.org/10.1109/CWTM.2011.5759534

[18] Gradone, J. C., Wilson, W. D., Glenn, S. M., & Miles, T. N.. (2023). Upper Ocean Transport in the Anegada Passage From Multi-Year Glider Surveys. Journal of Geophysical Research: Oceans, 128(7). https://doi.org/10.1029/2022jc019608