An Open-Source Autonomous Vessel for Maritime Research

Dr. Robert Kidd, State University of New York, Maritime College

Dr. Kidd completed his B.S., M.S. and Ph.D. at the University of Florida in 2011, 2013, and 2015 respectively. He worked at the Center for Intelligent Machines and Robotics at UF from 2009 to 2015 researching the use autonomous ground vehicles including ATVs, a Toyota Highlander, and a tracked loader. He has taught at SUNY Maritime College since 2015 running the capstone design sequence for mechanical engineers. His research interests include additive manufacturing, fault-tolerant control, artificial immune systems, and autonomous ground vehicles and surface vessels.

An Open-Source Autonomous Vessel for Maritime Research

Abstract

As autonomy becomes increasingly prevalent in the maritime industry, students entering the field will need to receive advanced training in this area. To address this challenge, this work details efforts to develop an independently deployable autonomous vessel (the AG-0) which is generated through combining existing open source resources. The vessel can be used as a low-cost solution for researching and teaching autonomy in the maritime environment and can also be used as an autonomous vessel to perform scientific research in inaccessible areas without requiring experience with coding or autonomy. Without the sensor package for scientific research, the vessel costs below $500 and can be built by students with limited technical knowledge. A first-iteration vessel is generated by engineering students and faculty before being given to undergraduate environmental science researchers. The performance of the design is evaluated by these non-engineers for both function and operability. Their recommendations are detailed along with design and construction information.

Introduction

While there is a consensus that autonomy is coming to the maritime industry and will be revolutionary, there is not a consensus on exactly how it will come. In general, the autonomous vessel community believe that autonomy could begin making certain vessels unmanned in the near future [1]. Many of these projects are currently in the demonstration phase, such as the Falco, an unmanned ferry from Finferries [2]. However, many in the maritime community feel that there are certain applications, such as non-electric vessels, that will require a trained crew onboard [3].

This uncertainty means that academic institutions are hesitant to fund new advanced training programs until they are certain what they should look like. For example, as autonomy increases on vessels, mariners could either focus on traditional computer engineering skills such as coding and data management of the digital twin – a virtual simulation of a vessel used for increasing efficiency and early fault detection – or focus on traditional trade school skills such as engine maintenance and repair. Few institutions are well positioned to develop new programs in computer science or engineering for mariners. Of the 7 collegiate maritime academies, none include a computer science or computer engineering program. Several of the academies offer degrees in electrical engineering, but these focus primarily on power generation and distribution instead of control and automation.

This means a significant skills gap is projected for engineers who are comfortable in both the maritime environment and with automation if changes are not made [4]. To deal with this, academic institutions need to be proactively building the infrastructure to prepare the next generation of maritime engineers even as the industry evolves. Low-cost programs will be critical to enable institutions to explore these high-tech programs and develop experience before the industry needs become clear.

Education with autonomy is difficult to get started in. Firstly, most autonomous vehicle programs are directed toward those with a coding or computer science background. For example, the majority of these programs are written for Linux. This means that both students and, sometimes, instructors must learn not only new coding skills and new programs, but also a new operating system. Once these software challenges are overcome, moving to real-world hardware can also be challenging. Purchasing a platform to test these new autonomy algorithms can be expensive. The well-equipped Clearpath Robotics Heron USV costs $35,000-$40,000. While significantly less capable, the BlueRobotics BlueROV2 can still cost $3,000 to $5,000. These costs can increase greatly if multiple vessels are needed for a class on autonomy.

Beyond these engineering considerations, environmental scientists studying the maritime environment have a need for remote data collection platforms. Without access to robotic platforms, scientists are relegated to either getting data from other scientists, collecting samples from the shoreline, or placing their sensors on a manned vessel. Again, these robots can be prohibitively expensive for small institutions or citizen scientists. Combining these two needs provides an opportunity to develop a low-cost autonomous vessel that can serve multiple roles on a campus and can be customized based on any local needs.

Design Specifications

The base AG-0 vessel, shown in figure 1, uses a 3D-printed hull with customizable inserts for different research applications and hardware configurations. Unlike most autonomous surface vessels (ASVs), this vessel used a monohulled design instead of a catamaran or trimaran design. The multihulled designs are common due to their high maneuverability and stability. These benefits come at a cost to both storage size and expense which were not feasible for this project. Additionally, since the goal of the AG-0 was to serve as a platform that could be used to teach real-world autonomy for ocean-going vessels in the future, a monohulled design was chosen to mirror the majority of these vessels. The hull is adapted from the n3m0 vessel created by Mike Holden at California State University Maritime Academy. The n3m0 hull is modeled after a US Navy PT boat.

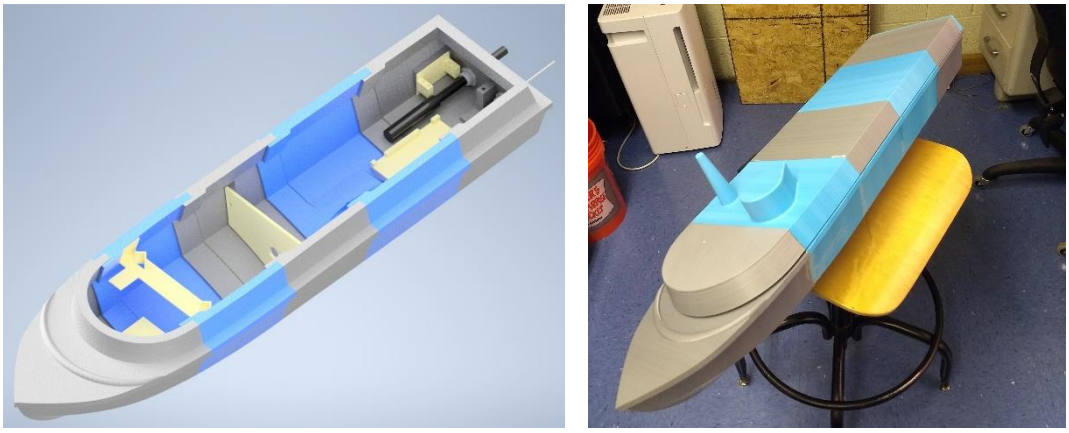

Figure 1: CAD model of the hull (left) and the 3D printed hull (right)

The hull has a length overall (LOA) of 35.5” (90.2 cm) and a beam of approximately 8.5” (21.6 cm). Internally, the vessel has a bulkhead at its approximate center that separates the vessel into a fore and aft sections. The fore section is designated for the electronic control package while the aft section is designated for power storage and sensor packages. The hull is designed to be printed in sections which fit snugly within the print bed of an Ultimaker 3D printer or any printer with a print bed at least 8.5” (21.6 cm) square. The top of the vessel is also 3D printed in sections and is held down through elastic straps. All antenna masts are included in a single section of the top so that only one part would need to be edited for different antenna configurations or if external atmospheric sensors were desired. The foremost sections of the hull and the top are left unused so future autonomy packages can utilize the space for additional sensing capabilities such as sonar or LIDAR. The aftmost section of the hull, shown in figure 2, includes all sensor mounts. If a different sensor package is desired, this section can be edited and replaced.

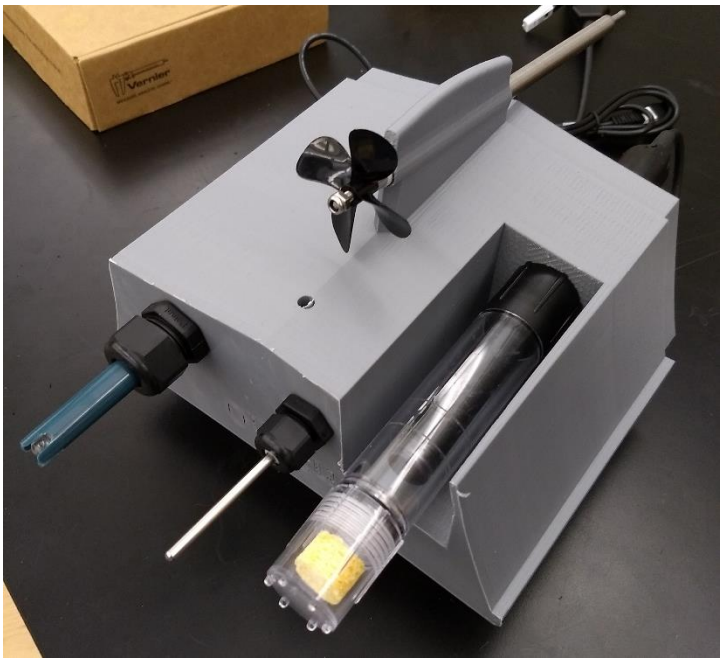

Figure 2: The aftmost section of the AG-0 hull showing the three sensors along with the propeller and shaft. Rudder not shown for clarity

The hull sections can then be epoxied together to form the uniform hull, and the top sections can be epoxied together to form a uniform cover. It is recommended to seal both the hull and the top to prevent any potential leaks and limit environmental deterioration. While PLA is a generally acceptable material to be used if protected, printing with ABS is recommended due to its superior performance in a saltwater environment. The files for the hull can be found at https://github.com/bokidd/AG-0_CAD.

Internally, the vessel has a series of customizable mounts for different configurations. The current implementation utilizes the center bulkhead as a motor mount for the propulsion system. This bulkhead is 3D printed and can be customized based on the motors available. If desired, this bulkhead could be expanded to separate the vessel into two watertight compartments. Within this bulkhead, routing points are included for cables connecting the sensor package to the electronics

package. The electronics package sits on a shelf that conforms to the curvature of the hull. This shelf provides a level mounting platform for the electronics while elevating them above the inner surface of the hull in case there are any leaks. For this application, the electronics package is protected in a separate IP67 waterproof case that was on-hand.

The electronics package consists of a Pixhawk flight controller, a Raspberry Pi 3 B+, and an Arduino Uno. The Pixhawk performs the basic navigation functions and interfaces with the propulsion hardware. It is an open-source hardware project that runs a suite of open-source software. This software includes the PX4 autopilot and the MAVLink protocol from the Dronecode project. The Arduino manages the sensor payload used for environmental testing and logs the generated data. The Pi provides high-level intelligence, control, and management of the vessel through MOOS-IvP, an open-source software ecosystem from MIT. Separating the electronics into these modules – navigation, payload, and intelligence – slightly increases the overall cost and complexity of the design. However, this modularity means the modules can be swapped as users desire. The navigation package can be upgraded to any other package that is compliant with the MAVLink protocol, such as the BeagleBone Blue. The modular design also allows the sensor payload to be removed from the vessel and operated manually on shore.

The propulsion system utilizes a brushless DC motor attached to a stock shaft and propeller. The motor, controller, and propeller can be varied if greater or lesser speeds are desired. Minimum design speeds and motor selections will be determined by the user's particular application space. While this is beyond the scope of this phase of the project, future work will define these relationships.

In consultation with local environmental scientists, the sensor package must be able to geotag and record the temperature, dissolved oxygen, and pH of the water. The data should be recorded approximately every second and should have accuracy of ± 0.9° F (0.5º C), $\pm 1.7*10^{-6}$ lb/gal (0.2 mg/L), and ±0.2 pH units, respectively. These values correspond to the specifications of the equipment currently used by the scientists.

To control the vessel, the Raspberry Pi is designed to use two modes, control via MOOS-IvP and control via a standard ground control station (GCS) such as Qgroundcontrol from Dronecode on a companion computer or smartphone. The GCS option allows unskilled users to plan and execute missions while the MOOS-IvP option allows complex intelligence tasks to be performed.

Implementation

For propulsion, the AG-0 utilizes a 1.375" (3.5 mm) propeller, a 4300 KV motor, and a 60 A electric speed controller powered by a 3S LiPo battery pack. Costs for the propulsion system were approximately $60 for the motor, controller, transmission shaft, and propeller with an additional $30 for the battery pack. Parts were connected through machined couplers and shafts. These were made by students in the campus machine shop to increase student exposure to the equipment and reduce costs. These are expected to provide a speed of 4 MPH (1.8 m/s) and an

endurance of 30 min. Future work will determine the real-world speed and endurance of the vessel.

The environmental scientists currently use a package from Vernier Software and Technology including a LabQuest 2 DAQ and three Vernier probes. The Vernier package is extremely expensive relative to the rest of the device, costing $762 before any software is used. The sensor package for the AG-0 replaces the DAQ entirely with the Arduino Uno and utilizes a DS18B20 waterproof temperature sensor in the place of the Vernier temperature probe. Interfacing the Vernier probes with the Arduino utilizes open libraries from Vernier. This reduces the cost to approximately $400-$450 depending on the equipment on-hand. The Arduino reads the data from the sensors continuously and logs their values twice a second. The Arduino also listens to a serial connection from the Raspberry Pi for GPS data and logs this information at the same time. The data is stored in a microSD card utilizing an OpenLog from SparkFun Electronics on an I2C interface. The code used for the Arduino can be found at https://github.com/bokidd/AG-0_Arduino.

Separating the Arduino from the Raspberry Pi allows several benefits. Firstly, the Arduino sensor package can be removed for use outside of the vessel. For user convenience during this process, a small I2C OLED screen – also an open project from SparkFun Electronics – is included to display the readings from the three environmental probes. Secondly, if users wish to use a different sensor package, such as the water sampling apparatus discussed later in the future work section, no changes need to be made to the Raspberry Pi. Finally, as the vessel's autonomous capabilities increase, the Raspberry Pi will likely need to be replaced with another single-board computer such as an ODROID or a full-size computer. In either case, no changes would need to be made to the sensor package. The Arduino board could likely be bypassed with the Vernier sensors communicating directly with the Raspberry Pi, however, adding this capability was determined to not be necessary for this project.

To implement the two different control modes in the Raspberry Pi, the system was configured to run one of two routines. The code for both routines can be found at https://github.com/bokidd/AG-0_PiCode. The first routine, called MAVLink_connection, is the GCS option. It acted as an information relay to query the autopilot for information such as current location and target waypoint. This information was then transmitted serially to the Arduino for logging. The GCS platform controls the vessel entirely. In the AG-0, a radio transmitter is connected to an Android phone running the Qgroundcontrol app. Missions are created in Qgroundcontrol according to the standard process in the program.

The second routine uses a MOOS-IvP wrapper for the MAVLink command node. The autopilot software in the Pixhawk is configured to run in a manual mode, accepting commands from a joystick. The MOOS-IvP wrapper converts the desired speed and desired heading commands into MAVLink compliant virtual joystick command messages and converts the MAVLink position and orientation information messages into MOOS-IvP messages. The Raspberry Pi can then run all remaining MOOS-IvP programs to enable the AG-0 to operate autonomously.

Results

Creating the AG-0 provided several important insights. Student participation was limited to 2 maritime engineering students working on hardware, 2 maritime engineering students working on software, and 2 non-engineering students working as testers for the environmental science components. Students were either juniors or seniors and spent between 2 and 10 hours per week on the project depending on the week. The engineering students had basic familiarity with coding in MATLAB or C++ and Autodesk Inventor or AutoCAD. The environmental science students had no familiarity with any autonomous vessel topics.

Firstly, the students working on the software implementation repeatedly discovered one of the difficulties in combining open source resources: incompatibility. An example can be seen in the MAVLink communication to the Arduino. The MAVLink implementation on the Arduino utilizes version 1.0 of the protocol while the majority of other equipment suppliers utilize the 2.0 version of the protocol. While systems are designed to be backward compatible, this version discrepancy was unable to be overcome easily. Upgrading the Arduino to the 2.0 version was not feasible due to the library memory requirements of the 2.0 version. Additionally, this meant that the Arduino libraries used in most tutorials called functions that were deprecated in the latest versions of MAVLink. Based on this experience, the students recommended including copies of all libraries used in the code repositories for the project. This will allow fallback options if future incompatibilities arise for other implementations.

Student comments regarding this underscored another major teaching point regarding the utilization of existing resources. Within this institution, as in others, there is a constant struggle to balance the need to ensure students learn material with the desire for them to leverage available resources. When incompatibility issues arose, the students learned that they will be unable to resolve these issues if they do not understand the fundamentals of the different systems. At the beginning of the project, students would copy-paste code snippets found online without hesitation. Toward the end of the project, students were much more interested in determining the limitations of the snippets before they committed to using them.

Students working on the hardware for the system demonstrated an increased ability to create new designs and design modifications that are both practical and easily manufactured. This ability to design for manufacturing, additive manufacturing in this case, is a critical ability for undergraduate students. This can naturally be done in coursework, but providing students with practical applications often increases their motivation and participation.

Comments from the non-engineering students emphasized that the equipment is an acceptable substitution when it operates cleanly. Specifically, occasional human errors on startup prevented certain components from initializing in the correct order, leading to communication errors and preventing the system from coming online. Additionally, occasional sensor readings were erroneous due to hiccups in the communication between the components. These did not cause issues during startup or with the performance of the autonomy but did cause concern for the students when the readouts did not show reasonable readings.

The students felt the smartphone interface worked well, with one exception. The students were required to run a few short terminal commands to start the applications on the Pi, which the students felt uncomfortable with. This also led to students occasionally starting components in the wrong order, creating the communication errors. Future revisions will remove this requirement.

One other major issue was the accuracy of the collected data. For most of the time, the data was sufficiently accurate as no error was discernable between the on-board sensors and the sensors currently used. However, the data intermittently included extreme outliers. This revision of the vessel used direct sensor readings at one instant for logging and did not use a checksum on received data. These oversights led to data entries that would occasionally spike to inaccurate values. These errors, such as a GPS reading that is on the opposite side of the globe or temperature readings below absolute zero, are easily identifiable in the logged data. Increasing the data rate, incorporating filters to smooth the data, outlier checking to remove erroneous info points, and incorporating a checksum where appropriate will reduce these issues.

Finally, they recommended several changes to the user interface, such as increased feedback to the user regarding the state of the vessel and the quality of the data being recorded. For example, including a visual readout showing whether the Arduino and Pi are communicating accurately would be beneficial for troubleshooting.

Conclusion

This project provided a vessel that can be operated by untrained non-engineers as well as providing a platform for future code development. In total, the vessel provided a low-cost option, costing approximately $75 for propulsion and steering hardware, $30 for batteries, $30-60 in 3D printing filament depending on infill densities and insert complexity, $200 for the Pixhawk, and $50 for the Raspberry Pi and Arduino. In total, the AG-0 without the sensor package costs approximately $400-$450. Including the sensor package, the cost increases to $800-$900. If cheaper alternatives for the sensor package are used, these costs could be reduced as well.

Future work for this project will include increasing the intelligence of the system, such as adding obstacle avoidance to the navigation package. Increasing the appeal of the user interfaces will be critical to increase adoption more broadly. Exchanging the sensor package for a water sampling apparatus is being explored as a potential option to expand the utility of the vessel. This apparatus would remove the cost of the sensor package from each vessel and allow additional tests to be performed on land including testing for microplastics. Finally, a new hull design is being explored as an opportunity for undergraduate naval architects to build a design utilizing what they have learned in their coursework.

Acknowledgements

Funding for this project was provided by the State University of New York Innovative Instructional Technology Grant.

The author would like to acknowledge Mike Holden of California State University Maritime Academy, SparkFun Electronics, Vernier Software and Technology, the Dronecode Project, and MIT for their contributions to the open source community, which have greatly benefited this project.

References

[1] O. Levander, "Path to Remote and Autonomous Shipping," in *Achieving Critical MASS: Spotlight on the U.S. Vessel Automation Industry*, Baltimore, 2019. Conference Presentation.

[2] AUVSI News, "Rolls-Royce and FinFerries Successfully Demonstrate World's First Fully Autonomous Ferry," 3 December 2018. [Online]. Available: https://www.auvsi.org/industry-news/rolls-royce-and-finferries-successfully-demonstrate-worlds-first-fully-autonomous.

[3] J. Strandberg, "Global Context of Maritime Automation and Autonomy," in *Achieving Critical MASS: Spotlight on the U.S. Vessel Automation Industry*, Baltimore, 2019. Conference Presentation. https://www.maritime.dot.gov/sites/marad.dot.gov/files/docs/about-us/foia/11721/w%C3%A4rtsil%C3%A4-global-context-maritime-automation-and-autonomy.pdf.

[4] M. H. Buzby, "Remarks Prepared for Mark H. Buzby Maritime Administrator MASS Conference," in *Achieving Critical MASS: Spotlight on the U.S. Vessel Automation Industry*, Baltimore, 2019. Conference Address. https://www.maritime.dot.gov/newsroom/speeches/mass-conference-2019.