2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Design and Experiments with LoCO AUV: A Low Cost Open-Source Autonomous Underwater Vehicle*

Chelsey Edge[1], Sadman Sakib Enan[1], Michael Fulton[1], Jungseok Hong[1], Jiawei Mo[1], Kimberly Barthelemy[2], Hunter Bashaw[3], Berik Kallevig[4], Corey Knutson[5], Kevin Orpen[2], Junaed Sattar[6]

***Abstract*— In this paper we present the LoCO AUV, a Low-Cost, Open Autonomous Underwater Vehicle. LoCO is a general-purpose, single-person-deployable, vision-guided AUV, rated to a depth of 100 meters. We discuss the open and expandable design of this underwater robot, as well as the design of a simulator in Gazebo. Additionally, we explore the platform's preliminary local motion control and state estimation abilities, which enable it to perform maneuvers autonomously. In order to demonstrate its usefulness for a variety of tasks, we implement a variety of our previously presented human-robot interaction capabilities on LoCO, including gestural control, diver following, and robot communication via motion. Finally, we discuss the practical concerns of deployment and our experiences in using this robot in pools, lakes, and the ocean. All design details, instructions on assembly, and code will be released under a permissive, open-source license.**

Fig. 1: LoCO in the Caribbean Sea off the coast of Barbados.

I. INTRODUCTION

Autonomous underwater vehicles (AUVs) are a key tool in scientific and industrial work in marine and aquatic environments. They are used to explore shipwrecks [1], chart biological habitats [2], destroy subsea mines [3], inspect and repair undersea architecture such as pipelines or cables [4], and to do a plethora of other tasks. However, AUVs are often expensive, large, and difficult to deploy, limiting their use to well-funded research groups, such as oceanography institutes, underwater robotics research labs, and university research teams. Additionally, AUVs are often not sold commercially and those which are, tend to be expensive to acquire and maintain. There are a great number of areas where the need for AUVs is outweighed by the challenges outlined above, such as use in marine science at a state and local government level, education at lower-funded universities and secondary schools, and in hobbyist development. If an AUV were available at a lower cost, with less overhead in deployment, and with less constraint on additions and modifications to the platform, those in these under-represented groups would be able to leverage the capabilities of an underwater autonomous agent to achieve their goals in research and education. This would also significantly reduce the barrier of entry into the theories and practices of autonomous underwater robotics.

In their sixty year history, autonomous underwater vehicles (AUVs) have opened an "eye into the deep" for humanity, revealing secrets of undersea life, patterns in currents, shipwrecks, and much more [2], [1]. The work of oceanography institutions has propelled AUVs from their beginnings as military tools into critical scientific instruments. The AUVs developed by these institutions carry complex sensor payloads, and are typically either deployed in concert with a support ship or, in the case of gliders, are sent on long-term missions on their own. While these large and expensive AUVs have played an important role in the development of the field of underwater robotics, a new breed of AUV is on the horizon. Due to improvements in battery technology and rapid growth in consumer-marketed remotely operated vehicles (ROVs), it is now feasible to create an AUV comprised of almost entirely off-the-shelf parts for a fraction of the cost. This will open up the world of autonomous underwater robots to a new group of users who did not have the funds or capability to build or purchase other AUVs. Students at many levels of education will be able to use these new AUVs to learn basic principles of underwater robotics, local and state-level marine research groups will have a new ability to use AUVs in their research, and even hobbyists could start to venture into AUVs at a higher level. We seek to contribute to this new wave of AUVs by presenting an AUV well suited to the small, lower-funded teams common in student groups and local and state researchers.

Authors presented in alphabetical order of surname within academic positions due to collaborative contribution.
1. PhD Student/Candidate, University of Minnesota Computer Science
2. Undergraduate, University of Minnesota Aerospace Engineering
3. Undergraduate, Clarkson University Computer Science
4. Undergraduate, University of Minnesota Mechanical Engineering
5. Undergraduate, University of Minnesota - Duluth Computer Science
6. Assistant Professor of Computer Science at University of Minnesota
*This work was supported by the Minnesota Robotics Institute Seed Grant and the National Science Foundation awards IIS-#1845364 & #00074041.

To this end, we present the *LoCO* AUV, a **Lo**w **C**ost, **O**pen, Autonomous Underwater Vehicle. LoCO is a human-portable, easy to deploy AUV built from approximately $4,000 worth of parts, largely off-the-shelf electronics and additively manufactured. While LoCO is not as specialized as more expensive AUVs, it boasts an impressive array of capabilities at this early stage in development, from autonomous swimming in defined patterns to diver following and control by hand gestures. Its hardware is designed to run vision-based algorithms, often employing deep learning for perception, enabling a great number of applications in the future. Moreover, the LoCO platform's design lends itself well to modularity and modification, making it relatively simple to add new sensor payloads or other hardware. With the openness of the design, this AUV is well-suited as a tool for those without access to the resources required to acquire and deploy more expensive AUVs.

In this paper we make the following contributions:

- We introduce the LoCO AUV: a low-cost, open AUV, designed to be easily built, modified, and deployed by small, low-resource teams.
- We discuss the current status of LoCO's system design, including its electrical, computing, and sensor systems.
- We discuss the current status of LoCO's state estimation and local control algorithms.
- We present a Gazebo-based simulator for LoCO, for prototyping algorithms before deployment.
- We incorporate our research lab's existing human-robot interaction capabilities to LoCO.

We will release all code, plans, 3D printed part models, assembly instructions, and documentation under a permissive, open-source license at `https://loco-auv.github.io/`.

II. RELATED WORK

We present a brief summary of previous work which is related to LoCO. The numerous AUVs and papers on AUVs prohibit an exhaustive review. Therefore, the authors primarily highlight previous work dealing with small and low-cost AUVs. AUV development can be considered to have begun [5] in 1957 with the development of the Special Purpose Underwater Research Vehicle (SPURV), at the University of Washington, funded by the United States Office of Naval Research [6]. SPURV was used until 1979, and followed by SPURV II [7], which improved on the hydrodynamic design of its predecessor, and increased the number of sensors onboard. Soon after, the REMUS AUV [8] was developed by Woods Hole Oceanographic Institute, in an attempt to develop a smaller, lower-cost AUV. The efforts put forward for REMUS are, in many ways, a spiritual predecessor to LoCO, as the team developing REMUS focused on low cost and a small support staff. Later, REMUS 600 [9] improved on the design of REMUS, with a side-looking synthetic aperture sonar and improved endurance and payload flexibility. By this time, gliders such as Seaglider [10] and Deepglider [11] were setting the standard for a new type of AUV: the long endurance glider, with mission times measuring in weeks and months rather than hours. While traditional torpedo-shaped AUVs and gliders still dominated, innovation in the realm of mini-AUVs (mass of 20-100kg) and micro-AUVs (mass of 20kg or less) began to spring up, with the flipper-driven Aqua [12] being an example of the variation now existent in the field.

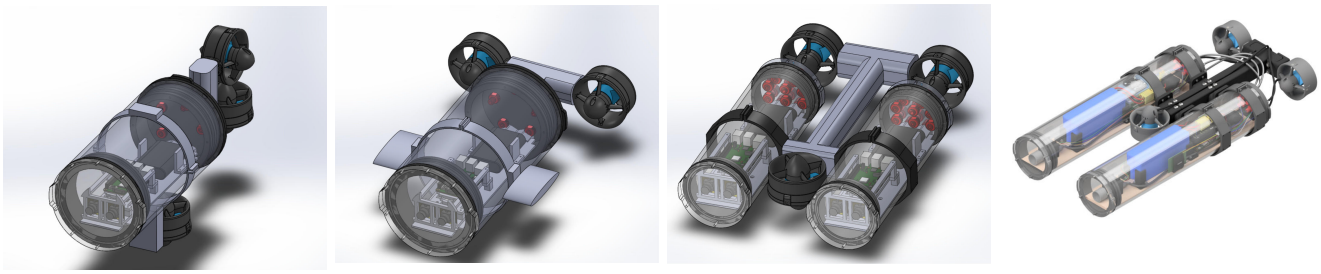
In recent years, development in the field of micro-AUVs has been particularly interesting, with the development of HippoCampus [13] and SEMBIO [14], micro-AUVs for swarm applications, and AUVs with less typical drive designs, such as a momentum-drive single actuated robot, which rotates its inner body to move the exterior passive flaps and produce swimming motion [15]. Other AUVs such as SHAD [15], HOBALIN [16], and Sparus II [17] focused on hovering motion for seabed inspection and observation tasks. Lastly, the development of general-purpose micro-AUVs is still going strong, with Bluefin Sandshark [18], a docking AUV [19], and other similar AUVs appearing in the last few years. The world of AUV development research remains a thriving one, full of new innovation and ideas. LoCO AUV attempts to lower the barrier of entry into that world even further than it has been lowered thus far, with an entirely open and reproducible design that will be available under a permissive, open-source license. LoCO can be purchased in component parts and built, and then modified to ones own standards making it an excellent platform for groups at any level, regardless of level of experience or funding.

III. SYSTEM DESIGN

LoCO is designed to be an all-purpose AUV, adaptable to a variety of missions. In the standard configuration, LoCO is a dual-camera, vision guided AUV with three thrusters. The following subsections describe the robot's electrical, thruster control, computational, and sensor systems.

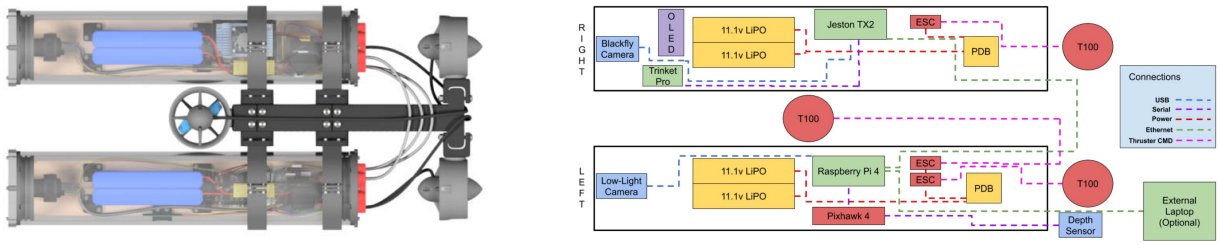
A. Overall System Layout

LoCO is comprised of two water-tight enclosures, each containing various components, with one thruster mounted between the enclosures, and two mounted behind, as seen in Fig. 3. While a number of designs were considered (see Fig. 2), the two-enclosure design was selected for a variety of reasons. Firstly, it allowed for a reasonable placement of a pitch-control thruster, along with providing space in between the enclosures which provides space to mount sensors, thrusters, or manipulators in the future. Additionally, the design called for two enclosures side by side, narrower than the enclosures used for the other designs, which would reduce the robot's forward profile, allowing it to move through the water with less resistance. Finally, the separation into two enclosures enforces a base level of modularity. Most control-related electronics are in the left-hand enclosure (shown in Fig. 3a and Fig. 3b), with the computational hardware for deep learning inference in the right-hand enclosure. While any type of hardware modification requires changes throughout the system, changes or replacements can be made with minimal impact on the layout of components internally.



(a) "Helicopter" design. (b) "Submarine" style. (c) "Binocular" design. (d) Final design.

Fig. 2: CAD renderings showing the development of LoCO AUV.



(a) (b)

Fig. 3: (a) Top-down view of the 3D model and (b) system schematic for LoCO AUV.

B. Watertight Enclosures and 3D printed substructure

The enclosures used are cast acrylic structures with a 10.16 cm internal diameter, produced by Blue Robotics. The enclosures are rated to 100m depth of operation, sufficient for our needs. Their length is variable to any modifications that are made, but the length for our configuration is 50.8cm. The front of each enclosure is a flat piece of cast acrylic to reduce visual distortion and increase surface area for mounting cameras, while the back cap is aluminum with openings for cables, plugged by nut-and-bolt assemblies termed penetrators. They are linked with a custom 3D printed structure connected to aluminum clamps manufactured by Blue Robotics. This structure is also used to mount the thrusters.

Internally, components are mounted on a piece of laser-cut medium-density fiberboard (MDF), with 3D printed mounting substructures. The MDF is held in place within the enclosures via a 3D printed part that attaches to the penetrators poking through the back plate. A strip of velcro beneath the MDF is used to attach 28 gram blocks of ballast cut from stainless steel bar stock. There is also enough space under the MDF to fit bags of desiccant, which help to eliminate condensation in the event that the enclosure is sealed in humid air (common in pool and tropical environments). This internal design is beneficial to the modularity of LoCO, as it allows for additional parts to be mounted by allocating space on the MDF and then simply attaching them with screws. Additionally, the external structure design is flexible, allowing for the movement of the clamps along the enclosures to fix the thrusters wherever is appropriate. The "backbone" 3D-printed part between the clamps can be omitted with no apparent loss of structural integrity. It would, however, provide useful mounting space for external sensors or actuators, such as a sonar altimeter or a gripper.

LoCO AUV Specifications

Dimensions (L x W x H)	73.1cm x 34.4cm x 14.1 cm
Weight	12.47 kg (27 lb)
Maximum Speed	1.5 m/s
Battery Life (Idle)	18 hrs, 30 mins
Battery Life (Average)	2 hrs, 20 mins
Battery Life (Max Thrust)	30 mins

Fig. 4: LoCO AUV Specifications.

C. Electrical Systems

LoCO's electronics are powered by four 11.1 volt lithium-polymer 8000 mAh batteries, two in each enclosure. The batteries in each tube are connected in parallel, providing a supply of 9.6-12.6 volts depending on their charge. Each tube has a low-voltage alarm which sounds when the two batteries fall below the minimum threshold of 9.6 volts. However, a custom circuit which will measure the batteries' level of charge and report the data to LoCO's computing systems is in development. The electrical systems in each enclosure are designed in such a way that they only power components in their respective tube. Each pair of batteries is connected to a power distribution board (PDB) which provides six 12 volt outputs, and one 5 volt output. In the case of the left-hand enclosure, an external 5 volt step down converter was added in order to power the additional computing systems in that enclosure. Despite the electrical systems of each tube being separate, a single power switch circuit controls the On/Off state of the entire robot. The circuit is situated in the left enclosure and utilizes a magnetic reed switch for input. This

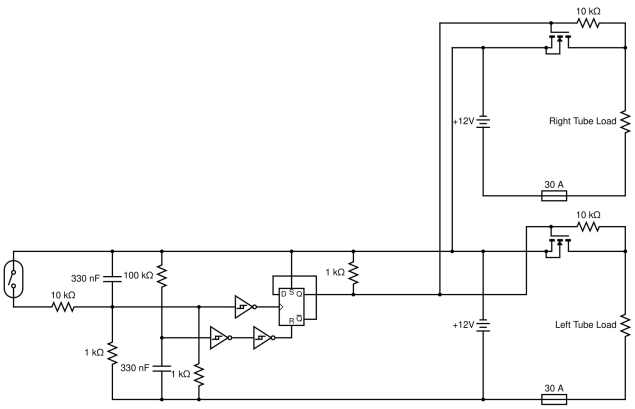


Fig. 5: Power Systems Diagram.

allows LoCO to be turned on and off using a magnetic "key". By using a magnetic key rather than a physical switch, we reduce possible points of failure due to leakage. The left portion of Fig. 5 contains the reed switch and components that create the toggle signal, while the right side diagrams how the toggle signal is used to control the power sources.

D. Thruster Control Systems

The thrusters employed for LoCO are Blue Robotics T100s, which use a brushless DC motor and a plastic propeller. Blue Robotics is discontinuing the T100, but they also produce a more powerful thruster (the T200) in the same dimensions, so future builds of LoCO AUVs will use T200s. These thrusters are controlled via pulse-width-modulation (PWM), which is managed by electronic speed controllers (ESCs). In turn, the ESCs are controlled by a Pixhawk autopilot board, employing the ArduPilot/ArduSub control software. While we do not currently make use of all the features of the Pixhawk and ArduSub, the open software and hardware of the PX4 [20] and ArduPilot [21] projects lend themselves well to our goals of creating an open platform which others can contribute to and adapt to their needs.

E. Computing Systems

LoCO has two primary computer systems: a Jetson TX2 for deep learning inference and a Raspberry Pi 4 (4GB) for control. The Jetson TX2 is mounted on a Connect Tech Orbitty carrier board for interfacing, and is largely responsible for managing processes which involve deep neural network inference. Due to its location in the right-hand enclosure, however, the TX2 is also responsible for managing the robot's OLED display via the connected microcontroller and processing images from the camera mounted in the right enclosure. In the left enclosure, the Raspberry Pi is used to process images from the camera in that enclosure and used as the controller interfacing with the Pixhawk autopilot over a serial connection. The Jetson and the Raspberry Pi are connected via Ethernet with a Cat5e cable for a maximum throughput of 100Mbit/s.

F. Vision Systems And Other Sensors

LoCO was designed with a stereo vision system in mind. The authors are currently investigating a number of cameras in order to select the appropriate camera for use in the final release version of the robot. Currently, an FLIR Blackfly^{TM} S USB3 camera is mounted in the right enclosure: a high powered camera which costs nearly $600. The left enclosure currently contains a significantly less expensive USB camera from Blue Robotics, which is nearly a sixth the price. The cameras are currently being compared for their quality, maximum feasible frame-rate, image quality, and effectiveness as part of the overall system before a decision is made. Stereo cameras are being considered, though finding cameras small enough for the enclosure's diameter has proved challenging. Another improvement to the vision system that is dependent on the camera selection process is installing an artificial light system to improve visibility in deep underwater environments. In addition to its vision system, LoCO employs a pressure sensor to measure its depth under the surface, which is mounted on the back plate of the left enclosure. There is also an inertial measurement unit (IMU) contained within the Pixhawk autopilot unit. Lastly, while it is not a part of the design, the authors are currently working on integrating a sonar altimeter, as it is likely to be a commonly used sensor.

IV. SOFTWARE

LoCO has a variety of computing devices: a Raspberry Pi 4, an Nvidia Jetson TX2, a Pixhawk autopilot unit, and an Adafruit Trinket Pro microcontroller. The Raspberry Pi and Jetson TX2 both run versions of Ubuntu, a popular open source operating system, while the Pixhawk runs a real-time open source OS with ArduSub, and the Trinket is flashed directly with open-source software. The software which allows LoCO to function as an untethered autonomous vehicle rather than an ROV is distributed across the computing devices, and consists of a mixture of ROS packages, Ardusub, and Arduino code. All of this software is under some form of permissive, open source license, which makes LoCO's software stack free for users to explore, expand, and enhance. In this section we describe a portion of LoCO's software, omitting a full description for brevity.

A. Camera Drivers

As mentioned previously in Section III-F, two cameras are currently being tested for use with LoCO. The first, a FLIR Blackfly S model, is controlled through a package from Neufield Robotics [22] which implements a Spinnaker SDK driver in ROS, the Robot Operating System [23]. This package allows for relatively full-featured control of the camera's many parameters, as well as controlling the camera's frame capture either through software triggering, or by waiting for a signal from the camera's GPIO pins. The other camera being tested conforms to the UVC standard, so the `libuvc_camera` ROS package is used to control it.

B. State Estimation

A number of options are under development for state estimation in LoCO. Firstly, the Pixhawk provides an IMU-based state estimation using the Extended Kalman Filter [24]. While this is useful, LoCO currently uses the

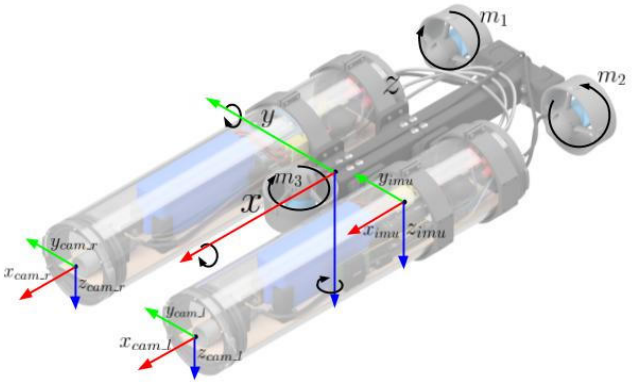


Fig. 6: A free-body diagram of LoCO AUV with IMU, camera, thruster, and robot frames.

`robot_localization` [25] ROS package to estimate its orientation via IMU data directly from the Pixhawk's IMU, as this provides greater control over the tuning of the extended or unscented Kalman filters provided by the package. Additionally, the package allows the fusion of multiple sources of information into one estimate, so if additional IMUs or sources of information became available, they could easily be integrated. Two possible sources of this information are currently in development: a downward-facing camera for monocular visual odometry and an odometry estimate based on a combination of thruster inputs and a hydrodynamic motion model of the robot.

C. LoCO Pilot Controller

In order to facilitate motion control of LoCO from a variety of sources (a teleoperation mode, or autonomous behavior algorithms), a motion control package entitled `loco_pilot` has been developed. The package provides an interface which abstracts the control into a simple message type (`\loco_pilot\Command`), containing thrust, pitch, and yaw values between -1.0 and 1.0. This allows users to avoid the `mav_ros`, MAVLink, and Ardusub systems which are required for controlling the robot, and simply publish these messages to the correct topic. In addition, the `loco_pilot` package implements a set of motion primitives and advertises them as ROS services, allowing one to simply call a service to turn the robot to an angle, move forward or back, or follow a circle or square trajectory. The package is under active development, adding more features and capabilities as the state estimation of the robot improves.

D. Menu Control System

Discussed in more detail in Section VI-A, LoCO has a menu system, implemented in the ROS package `loco_menu` which allows an operator to control the robot in untethered mode, by inputting commands via hand gestures or ARToolkit Tags [26]. This menu system allows a user to select an option, which can assigned to be a variety of subroutines, including ROS service calls, ROS launch files, etc. It uses a yaml format to define menus, making it a simple matter to create new menus for any task or environment.

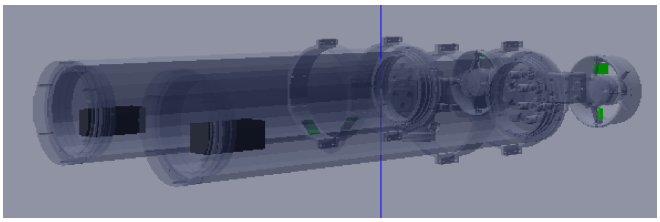


Fig. 7: The LoCO AUV in Gazebo simulation.

V. SIMULATION

The Gazebo-based simulator [27] currently under development for LoCO has the potential to reduce time and money spent testing software and dynamic behavior of the robot. The Gazebo simulation utilizes ROS for robot modeling and control.

A. Modeling and Visualization

The SolidWorks CAD design model shown in Fig. 2d was used as the template for creating the robot's Universal Robotic Description Format (URDF) model file. LoCO's inertial matrix and center of mass location were approximated based on a uniform density for all components using the SolidWorks Mass Analysis tool, taking into account the pre-existing mass measurements of LoCO. The center of mass was approximated between the LoCO tubes laterally and vertically, just behind the vertical thruster (see Fig. 6). To improve simulation efficiency while not affecting simulation physics, the mesh file that provides the visual representation of the robot does not include internal components. The initial components modeled in the simulation are representations of the front cameras to provide future creation of sensor data, and thruster links to allow for modeling of LoCO motion. Mass and inertial properties for the cameras and thruster propellers are assumed to be negligible.

Collision properties for each link of the robot were also specified so the simulation can model physical impacts between the robot and its surroundings. The collision boundaries were modeled to be the same as the visual boundaries for all links except the main body of LoCO. Due to the size and complexity of the mesh, the collision boundary for the main body was modeled as a box in order to decrease simulation computational requirements. The model is shown in Fig. 7, where the removal of internal components from the mesh can be seen.

B. Physics

In Gazebo, real-time fluid mechanics are not directly simulated, but rather the hydrodynamic forces are calculated, then applied to the robot. First, the ODE physics engine automatically applies a gravitational force to LoCO. Since LoCO is ballasted to be neutrally buoyant, the Gazebo BuoyancyPlugin applies the appropriate buoyancy force. The volume of LoCO is set in its model to overwrite the errors that would be produced by the bounding box approximation implemented by Gazebo to calculate model volume.

There are various options available for simulating movement or propeller propulsion in Gazebo, but it was determined the most accurate control method for the simulator would be directly applying thruster forces with the GazeboRosForce plugin. The underwater thrust force data for the Blue Robotics T100 (and the T200) thrusters is readily available on their website, which provides for simple calculation and application of forces in Gazebo. The robot thruster links are modeled as sets of blades in case future integration of propeller dynamics is desired.

Determining the drag forces applied to an underwater model is a complex task. Though methods have been developed to model underwater locomotion with modifications to a typical scalar mass and inertia matrix through a Kirchhoff tensor [28], this would require modification to the default Gazebo physics engine. To balance the complexity of the simulation program, an approximation of underwater forces served as the preferable option. Experimental underwater trials were performed with LoCO, with the objective for the robot to reach horizontal, steady-state forward velocity in a calm environment at various thruster power inputs. IMU data was used to calculate velocity, and, with thruster forces known from the available Blue Robotics data, the balancing water drag force could be calculated, and therefore the cumulative coefficient of drag. This is used in the simulation to apply drag force against LoCO's forward and reverse motion. Since similar underwater data cannot be readily gathered for vertical, lateral, and rotational motion of the robot, those drag parameters currently remain to be evaluated.

C. Control

A simulation control node was written to bridge existing LoCO software with the Gazebo simulation. Receiving typical LoCO motion commands, it applies thrust and drag forces to the robot by using ROS Wrench messages. It also subscribes to the link states published by Gazebo so that frame transformations can be applied to desired thrust within the node before publishing the Wrench commands.

VI. HUMAN-ROBOT INTERFACE

In order to demonstrate LoCO's capabilities to be used for a variety of AUV research topics, as well as to improve the usability of the platform, we ported a number of HRI capabilities previously developed for the Aqua AUV. These capabilities require the use of LoCO's deep learning, vision guidance, and local control capabilities. Additionally, a strong interactive interface will be key to making LoCO usable by scientists with little programming experience.

A. Menu System

LoCO's current interface is based on simple menu selection system. The robot's user selects from a set of 5 options using one of the methods described in the following subsection. While these options are displayed on LoCO's OLED display for ease of use, the menu system operates independently of its display component. These options could be connected to a variety of ROS endpoints, including

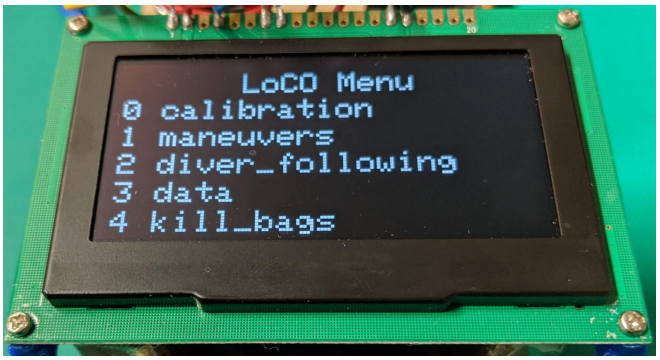


Fig. 8: LoCO's OLED display showing a sample menu.

running a launch file, launching a node, calling a service, terminating a node or setting a parameter. Menu options can also be set to submenus, allowing nesting and categorization of options. Once a menu item has been selected, while response depends on what endpoint the option has been linked to, typically an action is taken relatively quickly, then the menu is available again once the action has been completed. In some cases, the option has a timeout associated with it, or must be manually canceled. The system is designed to be as adaptable as possible, with menu definitions being loaded in the form of yaml configuration files, making the process of writing new menu configurations as painless as possible.

B. ARToolkit and Hand Gestures

LoCO employs two vision-based methods to select menu items. The first uses small AR tags as "flashcards". A ROS package based on AR Toolkit detects tags in the view of the left enclosure's camera. To select a menu item, one simply has to display an AR tag corresponding to the number of the item. Alternatively, LoCO has been outfitted with a gesture recognition system, which was initially presented for the Aqua AUV [29], and is based on a hand pose classification deep neural network. To use the menu, one must display the "Ok" gesture, then the number for the appropriate item.

C. Robot Communication Via Motion

While the menu is displayed on the vehicle's OLED, LoCO also employs a method for robot-to-human communication previously developed for the Aqua AUV: RCVM [30]. RCVM (robot communication via motion) is a method for

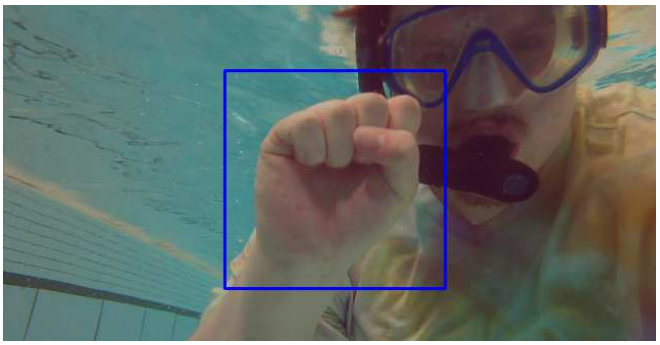


Fig. 9: Gesture detection for a "0" gesture.

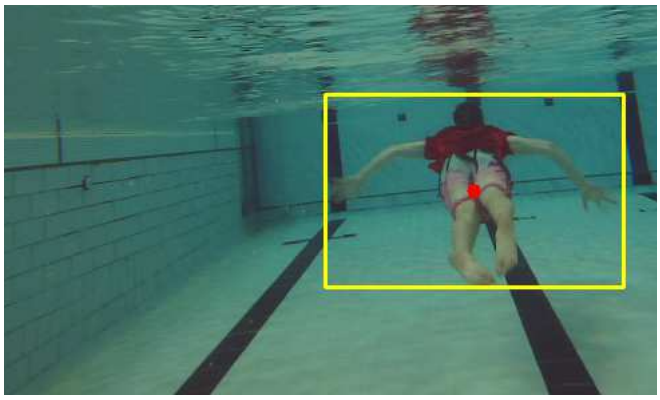


Fig. 10: LoCO-eye view of following a diver in a pool.

robot feedback using motion as the communication vector. For instance, to confirm some information to its human interactant, the robot can pitch up and down, mimicking a head nodding gesture. RCVM enables communication at greater distances and from more accessible angles than the OLED, though it does sacrifice communication "bandwidth". It also serves as a good test of the local motion control systems previously described, as fast and accurate motion control is key to producing proper motion for communication.

D. Diver Following

The final HRI capability reproduced on LoCO was diver following. While diver following has been explored by a variety of authors, the ability of an AUV which works with human partners to follow those partners is foundational. Diver following can be used to convoy with a robot to a chosen location, to guide a robot through a specific route during data collection, or simply as the end goal, to name a few uses. As with the gesture recognition system, diver following on LoCO employs a deep neural network for diver detection previously used for the Aqua AUV [31], making it an excellent use case test for the platform. Once the detection has been made, a PID-based controller is used to generate thruster control inputs which will maneuver the bounding box into the center of the frame. The algorithm uses bounding box size as a rough, stereo-free estimate of distance to the diver. While there is room for improvement in the diver following in practice, the existing algorithm does a serviceable job of following.

VII. EXPERIMENTS

One of LoCO's many strengths is its ability to be deployed by a small team. Particularly when deploying the robot without a tether, it is easy to assemble, seal, and carry by hand to the water's edge and deploy. In this section we will briefly discuss some of the deployment process, as well as our experiences from a variety of deployment scenarios.

A. Deployment Methodologies

When deploying LoCO, a number of things must be done. Firstly, the batteries must be placed into their seats on the MDF and connected with the electrical system. This is a simple enough affair, merely requiring the use of a few XT90 connectors. After this, the watertight enclosures must be closed, a straightforward process of pushing the MDF substructure into the tubes until the gaskets are seated. Following this, it is recommended to pump air out of the enclosures to a vacuum level of -15mm hg to ensure that there are no leaks in any of the enclosures' seal. With a weak vacuum established, the specialized penetrators used for attaching the vacuum pump are sealed with vent plugs. Once that has been done, the only thing remaining is to assemble the superstructure by simply inserting four bolts through 3D printed parts and tightening their respective nuts. With this complete, the robot is ready to be deployed.

B. Deployment Scenarios Thus Far

LoCO AUV has been deployed in pools seven times, at Wayzata Bay, Minnesota once, and off the coast of Barbados five times. LoCO's deployments have, thus far, been without incident. Of those deployments, the majority have been tethered for data collection and live debugging purposes. There have been a handful of tetherless deployments, all in the Caribbean Sea. Most of these deployments were undertaken with a team of 4 or fewer people. In most cases, one person acted as "robot wrangler", interacting with the robot and staying near it in the case of field deployments. The other team members were typically spread between taking external videos of the deployment, observing and debugging systems via tether, or working as the target of some algorithm (i.e. diver following). This shows the ease with which this robot can be deployed. As LoCO's local motion controller and HRI suite improve, the feasibility of single-person deployments will grow exponentially. The primary roadblocks to a single-person deployment at the moment are the lack of diver-relative station keeping and the fact that most deployments of LoCO at the moment focus on developing some aspect of the robot, not achieving some other disconnected goal. The authors are in collaboration with marine biologists from the University of Minnesota to begin trials with LoCO acting as data acquisition support to their typical observation methods. These trials will undoubtedly provide useful insight into ways that the robot's interfaces and capabilities can be used by scientists with little-to-no robotics knowledge.

VIII. CONCLUSION

In this paper, we presented the design of LoCO AUV, a Low-Cost, Open Autonomous Underwater Vehicle. Along with details of the vehicle's hardware and software design, we discussed our development of a Gazebo-based simulator as well as an HRI interface for LoCO, and our experiences in deployment thus far. More detailed documentation of hardware and software, as well as a parts list and assembly instructions will soon be available at `https://loco-auv.github.io/` under a permissive open-source license. While no individual scientific leap forward is presented in this paper, LoCO represents the summation of many smaller pieces of progress. It is not the first AUV of its size or capabilities, nor is it faster or more depth capable than some AUVs. However, LoCO is, to our knowledge,

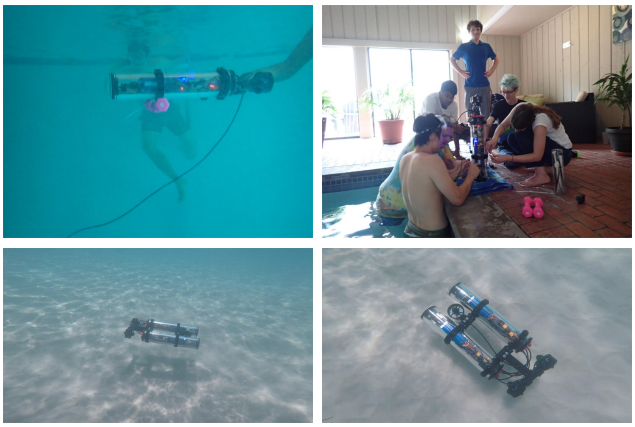


Fig. 11: A sampling of early LoCO deployments.

one of the first open AUVs to be made available to the public, significantly lowering the barrier to entry in the field of robotics. LoCO's designed purpose is and always has been to provide a platform for underwater robotics to those who could not have previously afforded their own. It is in this capacity that LoCO shines, and will perform with distinction. We hope to see LoCO robots and new variants of the platform begin to appear throughout the world, as LoCO enables more and more interested parties to take their first steps into the field of underwater robotics.

References

[1] B. P. Foley, K. Dellaporta, D. Sakellariou, B. S. Bingham, R. Camilli, R. M. Eustice, D. Evagelistis, V. L. Ferrini, K. Katsaros, D. Kourkoumelis, A. Mallios, P. Micha, D. A. Mindell, C. Roman, H. Singh, D. S. Switzer, and T. Theodoulou, "The 2005 Chios Ancient Shipwreck Survey: New Methods for Underwater Archaeology," *Hesperia: The Journal of the American School of Classical Studies at Athens*, vol. 78, no. 2, pp. 269–305, 2009.

[2] S. B. Williams, O. Pizarro, M. Jakuba, and N. Barrett, "AUV Benthic Habitat Mapping in South Eastern Tasmania," in *Field and Service Robotics*, A. Howard, K. Iagnemma, and A. Kelly, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 275–284.

[3] S. Sariel, T. Balch, and J. Stack, "Distributed multi-auv coordination in naval mine countermeasure missions," Georgia Institute of Technology, Tech. Rep., 2006.

[4] Y. R. Petillot, S. R. Reed, and J. M. Bell, "Real time AUV pipeline detection and tracking using side scan sonar and multi-beam echo-sounder," in *OCEANS '02 MTS/IEEE*, Oct 2002, pp. 217–222 vol.1.

[5] S. A. Gafurov and E. V. Klochkov, "Autonomous Unmanned Underwater Vehicles Development Tendencies," *Procedia Engineering*, vol. 106, pp. 141 – 148, 2015.

[6] H. R. Widditsch, "SPURV - The First Decade," Defense Technical Information Center, Fort Belvoir, VA, Tech. Rep., Oct. 1973.

[7] W. Nodland, T. Ewart, W. Bendiner, J. Miller, and E. Aagaard, "SPURV II-An Unmanned, Free-Swimming Submersible Developed for Oceanographic Research," in *OCEANS 81*, Sep. 1981, pp. 92–98.

[8] B. Allen, R. Stokey, T. Austin, N. Forrester, R. Goldsborough, M. Purcell, and C. v. Alt, "REMUS: a small, low cost AUV; system description, field trials and performance results," in *Oceans '97. MTS/IEEE Conference Proceedings*, Oct. 1997, pp. 994–1000.

[9] R. P. Stokey, A. Roup, C. v. Alt, B. Allen, N. Forrester, T. Austin, R. Goldsborough, M. Purcell, F. Jaffre, G. Packard, and A. Kukulya, "Development of the REMUS 600 autonomous underwater vehicle," in *Proceedings of OCEANS 2005 MTS/IEEE*, Sep. 2005, pp. 1301–1304 Vol. 2.

[10] C. C. Eriksen, T. J. Osse, R. D. Light, T. Wen, T. W. Lehman, P. L. Sabin, J. W. Ballard, and A. M. Chiodi, "Seaglider: a long-range autonomous underwater vehicle for oceanographic research," *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 424–436, Oct. 2001.

[11] T. J. Osse and C. C. Eriksen, "The Deepglider: A Full Ocean Depth Glider for Oceanographic Research," in *OCEANS 2007*, Sep. 2007, pp. 1–12.

[12] G. Dudek, P. Giguère, C. Prahacs, S. Saunderson, J. Sattar, L. A. Torres-Méndez, M. Jenkin, A. German, A. Hogue, A. Ripsman, J. Zacher, E. Milios, H. Liu, P. Zhang, M. Buehler, and C. Georgiades, "AQUA: An amphibious autonomous robot," *Computer*, vol. 40, pp. 46–53, Feb. 2007.

[13] A. Hackbarth, E. Kreuzer, and E. Solowjow, "HippoCampus: A micro underwater vehicle for swarm applications," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 2258–2263.

[14] A. Amory and E. Maehle, "SEMBIO - a small energy-efficient swarm AUV," in *OCEANS 2016 MTS/IEEE Monterey*, Sep. 2016, pp. 1–7.

[15] C. S. Gonçalves, B. M. Ferreira, and A. C. Matos, "Design and development of SHAD - a Small Hovering AUV with Differential actuation," in *OCEANS 2016 MTS/IEEE Monterey*, Sep. 2016, pp. 1–4.

[16] A. Okamoto, K. Tamura, M. Sasano, K. Sawada, T. Seta, S. Inaba, T. Ura, Y. Nishida, J. Kojima, and Y. Itoh, "Development of hovering-type AUV "HOBALIN" for exploring seafloor hydrothermal deposits," in *OCEANS 2016 MTS/IEEE Monterey*, Sep. 2016, pp. 1–4.

[17] M. Carreras, J. D. Hernández, E. Vidal, N. Palomeras, D. Ribas, and P. Ridao, "Sparus II AUV—A Hovering Vehicle for Seabed Inspection," *IEEE Journal of Oceanic Engineering*, vol. 43, no. 2, pp. 344–355, Apr. 2018.

[18] A. Underwood and C. Murphy, "Design of a micro-AUV for autonomy development and multi-vehicle systems," in *OCEANS 2017 - Aberdeen*, Jun. 2017, pp. 1–6.

[19] J. Wu, S. Peng, T. Xu, R. Hu, S. Wang, M. Pan, and X. Weng, "Test Bed AUV for Docking Algorithm Research," in *OCEANS 2018 MTS/IEEE Charleston*, Oct. 2018, pp. 1–6.

[20] L. Meier, D. Honegger, and M. Pollefeys, "PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 6235–6240.

[21] "ArduPilot," https://ardupilot.org/, (Accessed on 03/01/2020).

[22] "(FLIR) Spinnaker based camera driver," https://github.com/neufieldrobotics/spinnaker_sdk_camera_driver.

[23] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3. Kobe, Japan, 2009, p. 5.

[24] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *Signal processing, sensor fusion, and target recognition VI*, vol. 3068. International Society for Optics and Photonics, 1997, pp. 182–193.

[25] "robot_localization," https://github.com/cra-ros-pkg/robot_localization.

[26] H. Kato and M. Billinghurst, "Marker Tracking and HMD Calibration for a video-based Augmented Reality Conferencing System," in *Proceedings of the 2nd International Workshop on Augmented Reality (IWAR 99)*, San Francisco, October 1999.

[27] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, Sep. 2004, pp. 2149–2154.

[28] S. Weissmann and U. Pinkall, "Underwater rigid body dynamics," *ACM Transactions on Graphics (TOG)*, vol. 31, July 2012.

[29] M. J. Islam, M. Ho, and J. Sattar, "Dynamic reconfiguration of mission parameters in underwater human-robot collaboration," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 6212–6219.

[30] M. Fulton, C. Edge, and J. Sattar, "Robot communication via motion: Closing the underwater human-robot interaction loop," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 4660–4666.

[31] M. J. Islam, M. Fulton, and J. Sattar, "Toward a generic diver-following algorithm: Balancing robustness and efficiency in deep visual detection," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 113–120, Jan 2019.