2020 IEEE/RSJ

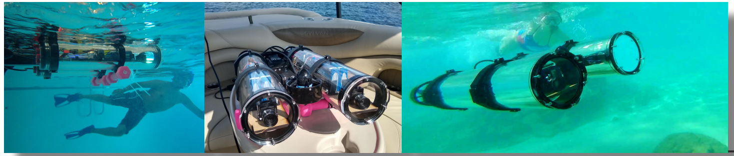
International Conference on
Intelligent Robots and Systems(IROS)

October 25-29, 2020 Las Vegas, NV, USA

Sponsors

RSJ IEEE Robotics & Automation Society ies SICE NTF New Technology Foundation IEEE

Theme: Consumer Robotics and Our Future



LoCO-AUV

Interactive Robotics and Vision Lab
Department of Computer Science and Engineering
Minnesota Robotics Institute

University of Minnesota
Driven to Discover℠

AUVs: Complex and Expensive

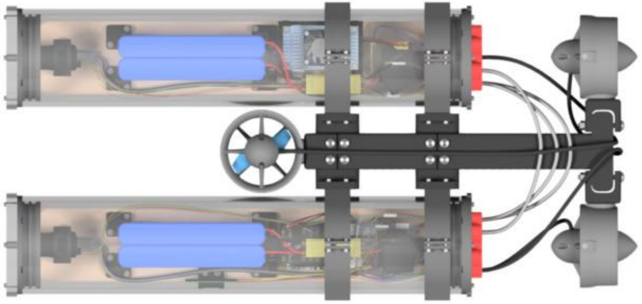
- AUVs (Autonomous Underwater Vehicles) have a long history.
  - Beginning in 1957 with SPURV, moving to REMUS and many, many more.

- Most AUVs are:
  - Expensive.
  - Too large to be deployed by hand.
  - Too complex for non-expert use.

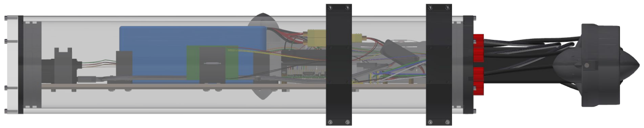
- However:
  - New inexpensive hardware for ROVs and AUVs.
  - Improved battery technology allows for smaller platforms.
  - Development of natural HRI techniques can remove barriers of use.

What Is LoCO?

- Low-Cost:
  - LoCO is an AUV, built with off-the-shelf and 3D-printed parts.
  - The total build cost is around 4,000 USD.



- Open:
  - All software currently used for LoCO is open source.
  - The design, schematics, and assembly instructions will all be released once development has reached 1.0.



- Designed for usability:
  - Made to be deployable by a small team (1-3 people)
  - Variety of HRI capabilities (diver following, gesture recognition, etc.)

System Design

Three early designs for LoCO and the final design.

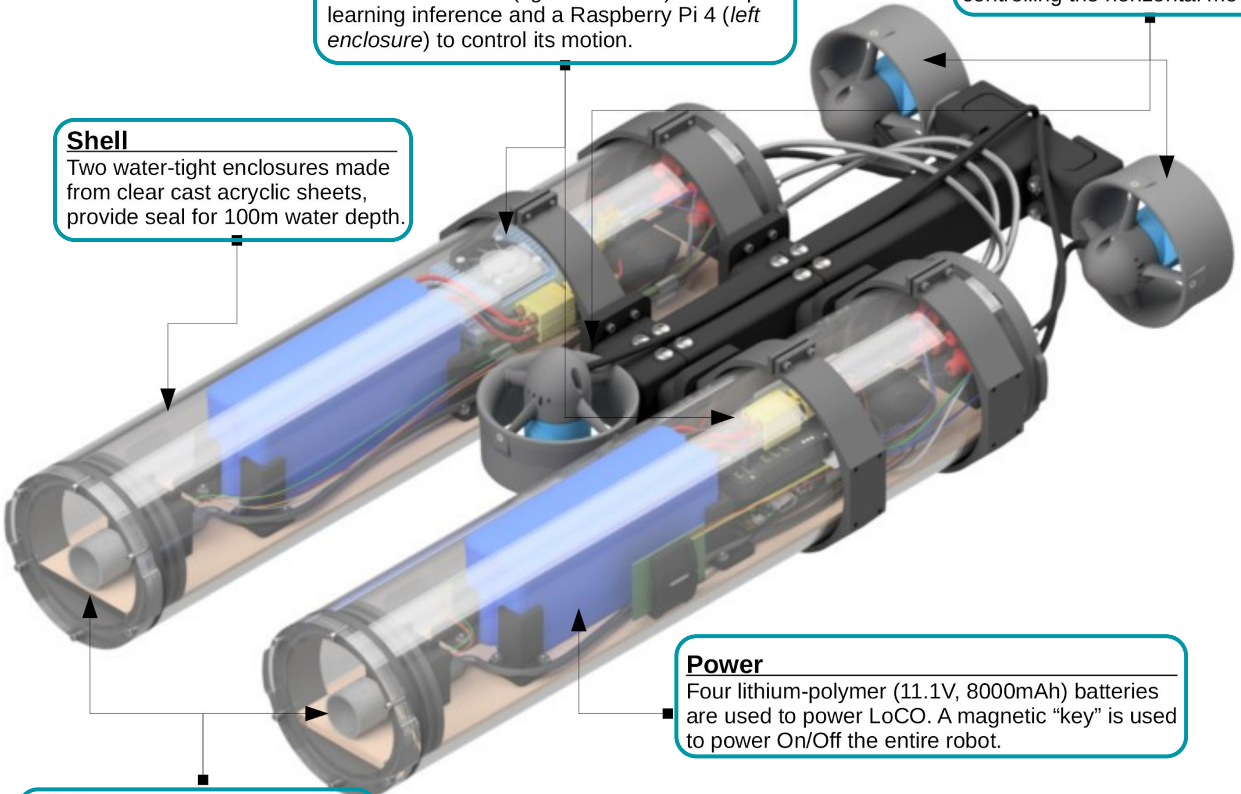
- ❖ System Layout
- ❖ Mounting Structures
- ❖ Electrical Systems
- ❖ Magnetic Power Switch
- ❖ Computing Systems
- ❖ Vision and Other Sensors

LoCO

Computation
LoCO operates with two computer systems: an Nvidia Jetson TX2 (*right enclosure*) for deep learning inference and a Raspberry Pi 4 (*left enclosure*) to control its motion.

Thrusters
LoCO has three thrusters: one for controlling the pitch and the others for controlling the horizontal movement.

Shell
Two water-tight enclosures made from clear cast acryclic sheets, provide seal for 100m water depth.



Power
Four lithium-polymer (11.1V, 8000mAh) batteries are used to power LoCO. A magnetic “key” is used to power On/Off the entire robot.

Vision
Two forward facing USB cameras allow human-robot interaction, as well as vision-based perception.

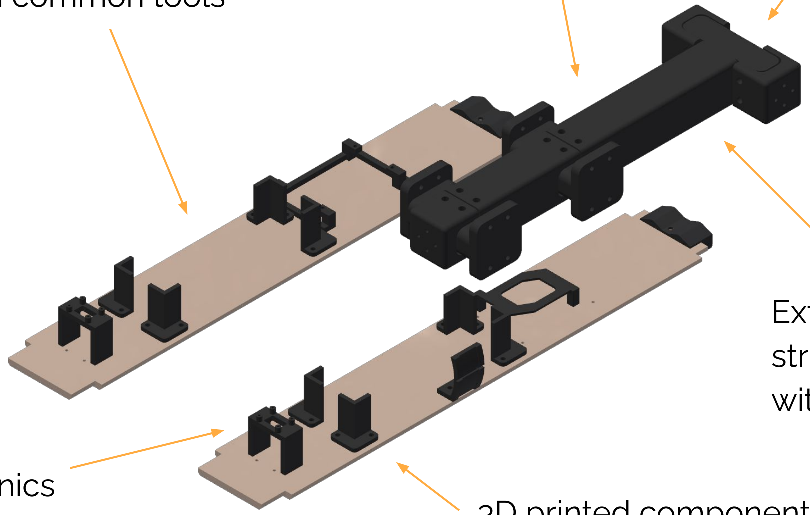
Dimensions: 34.4cm (w) x 73.1 cm (l) x 14.1cm (h)
Weight: 12.47 Kg

Manufactured Mounting Structures

¼” MDF wood base laser cut then post-processed with common tools

PETG plastic printed on desktop 3D printers then post-processed with common tools

External structure connects thrusters and external sensors to watertight enclosures

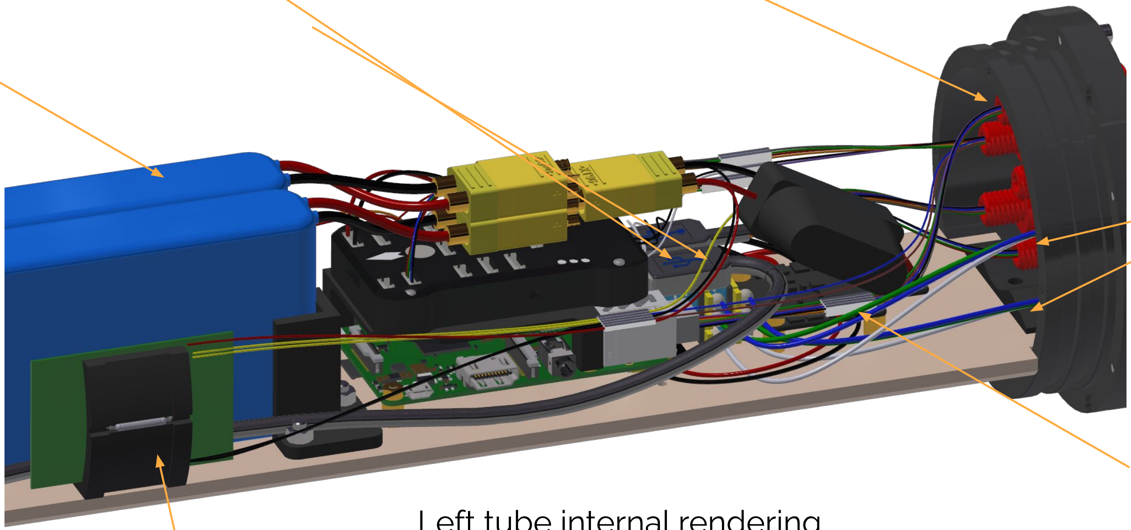


External 3D printed structure connected with larger bolts

Internal structure connects electronics to MDF base, which connects to shell

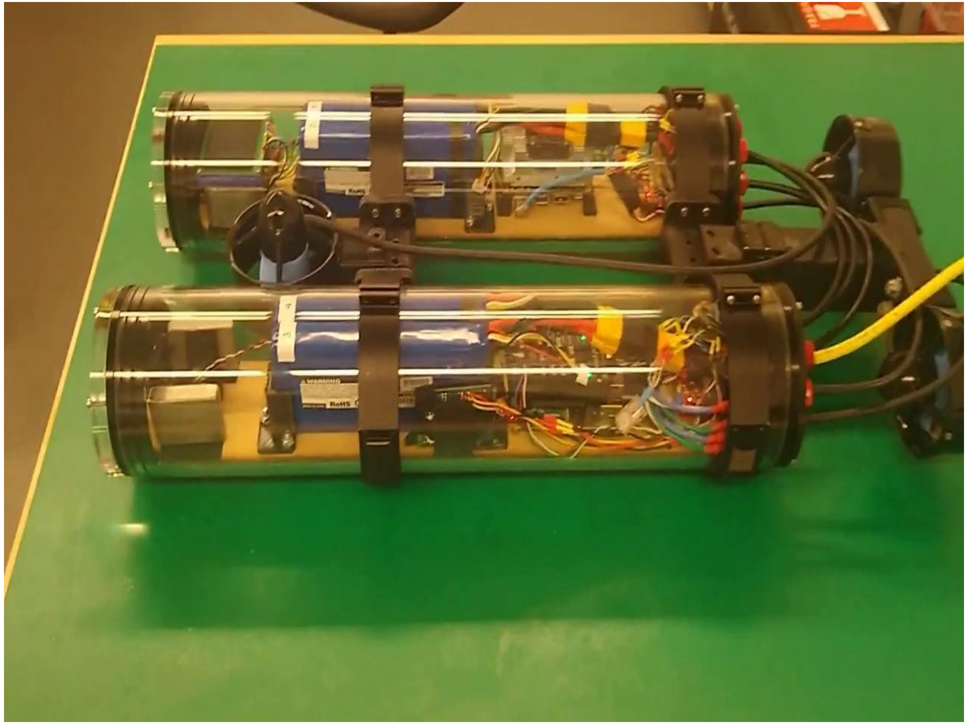
3D printed components attached to MDF base with small wood screws and bolts

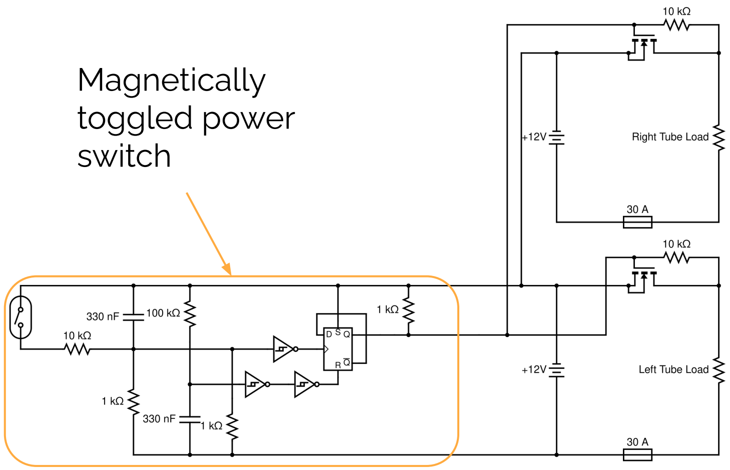
Power and Control Systems



Left tube internal rendering

Magnetic Switch





Power System Diagram

Computing Systems

- LoCO has two primary computer systems:
  - **Nvidia Jetson TX2**: responsible for deep learning inference.
  - **Raspberry Pi 4**: used for system control, e.g. interfacing with the Pixhawk.

- Computers are connected via Ethernet with a Cat5e cable, and to on-shore computing via a Blue Robotics Fathom Tether.

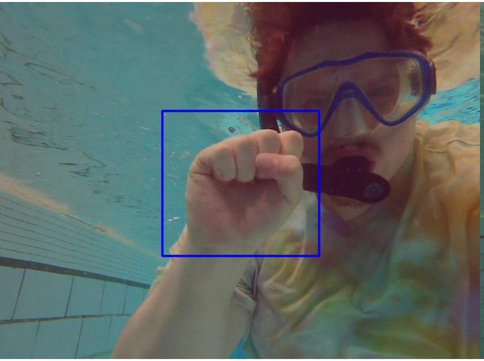
- Removable storage in each computer for easy data transfer.

Vision Systems And Other Sensors

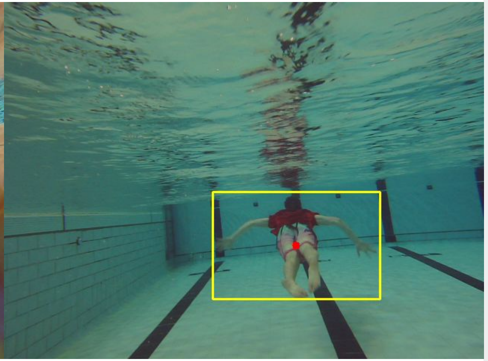
- LoCO employs a dual vision system.
  - Both enclosures contain an efficient USB camera from Blue Robotics.
  - Mini stereo cameras are being considered for the right enclosure.
  - Multiple camera configurations to fit needs/budget are possible.

- Other sensors:
  - Depth (pressure) sensor.
  - IMU (integrated into Pixhawk).
  - Internal Temperature
  - *Sonar Echo Sounder (in progress)*

Software



Detection of hand gesture in pool environment.



Detection of diver in pool environment.

- General Software Info
- State Estimation & LoCO Pilot
- Deep Learning Models
- Simulation and Gazebo

General Software And Licensing

- General Software
  - Distributed across computing.
  - Variety of projects and platforms
  - All under permissive and open source licenses.

- Most of our custom code is:
  - System configuration files.
  - ROS nodes.
  - Tensorflow networks.

State Estimation & LoCO Pilot

- State Estimation
  - The *robot_localization* ROS package estimates an orientation via IMU data.
  - Additional sensors can be integrated to provide improved estimation.

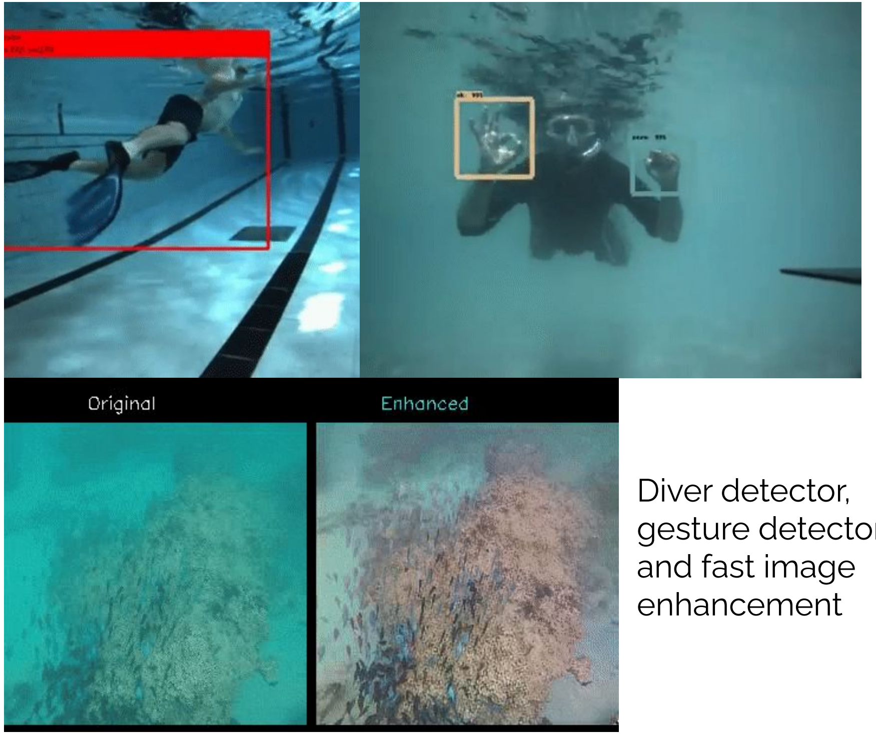
- LoCO Pilot
  - *loco_pilot* is a motion control package developed to offer motion control of LoCO from a variety of sources (a teleoperation node, or autonomous control)

  - The package abstracts the control into a simple message type (\loco_pilot\Command), containing thrust, pitch, and yaw values.

Deep Learning Models

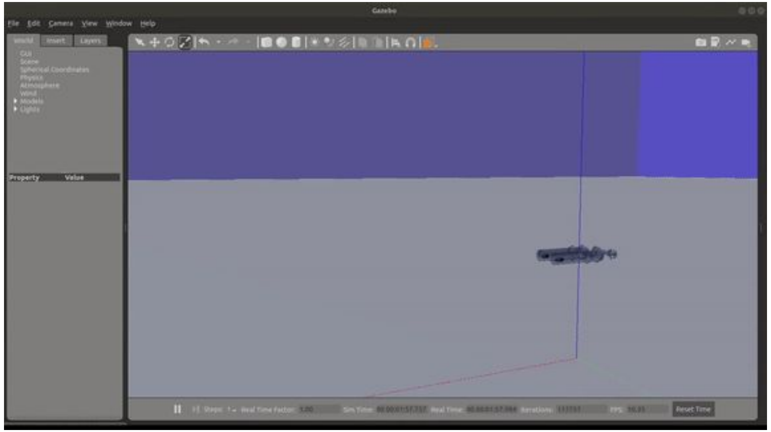
- LoCO currently uses a number of deep learning models created by members of the IRVLab, including:
  - Deep Diver Detection
  - Gesture Detection
  - Fast Underwater Image Enhancement
  - Simultaneous Enhancement and Super-Resolution

- While the image enhancement software is optional, it improves the accuracy of other algorithms, including the diver detector.



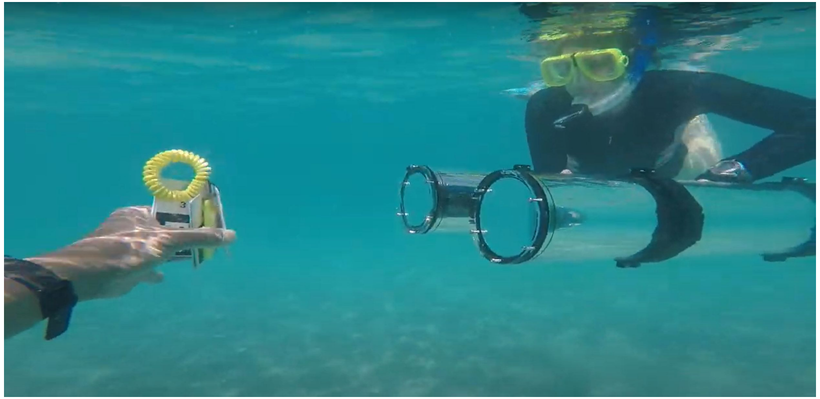
Diver detector, gesture detector, and fast image enhancement

Gazebo Simulation



- ❖ CAD Based Model
  - ➢ Front cameras for future sensor data
  - ➢ Thruster blades for future propeller dynamics
  - ➢ Internal components omitted
  - ➢ Bounding boxes for collision properties
- ❖ Application of Forces
  - ➢ Gravity force from Gazebo physics engine
  - ➢ Buoyancy force applied at center of mass.
  - ➢ Thruster forces applied at propellers based on manufacturer data
  - ➢ Drag forces and torques applied at center of mass
- ❖ Experimental Data
  - ➢ Underwater drag force approximated from underwater trials

Human-Robot Interface

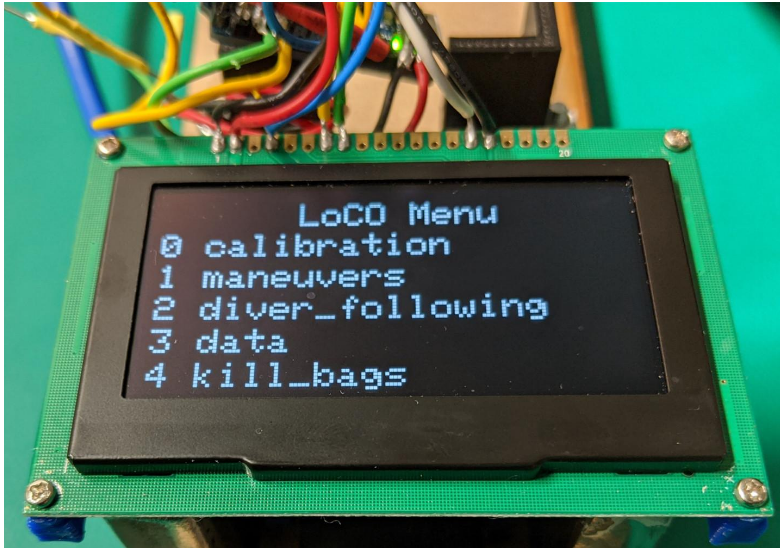


ARToolkit tag being shown to LoCO in Barbados

- Menu System
- ARToolkit and Hand Gestures
- Diver Following
- Deployment Procedures

Menu System

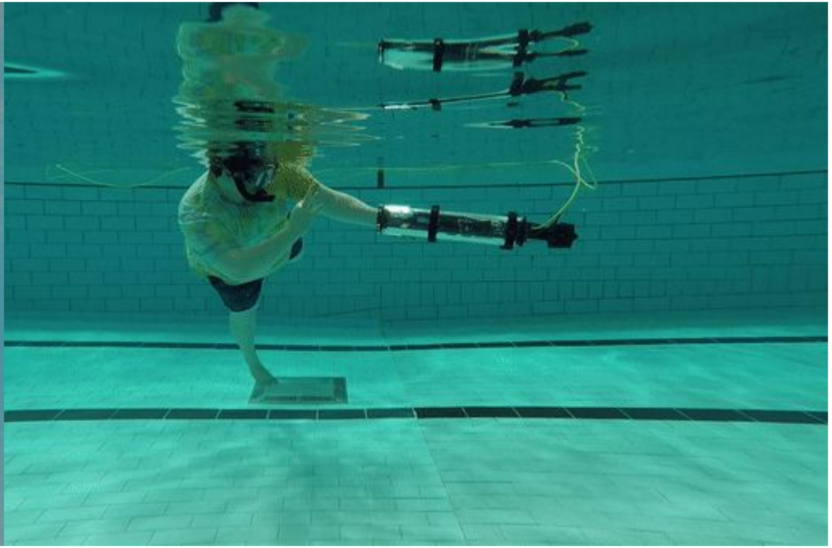
- LoCO employs a menu, displayed on a grayscale OLED to allow interaction.
- Menu items can map to submenus, scripts, or ROS services.
- Items can be selected by a number of methods, and can run indefinitely, for a given duration, or until canceled.



OLED displaying sample menu (right enclosure)

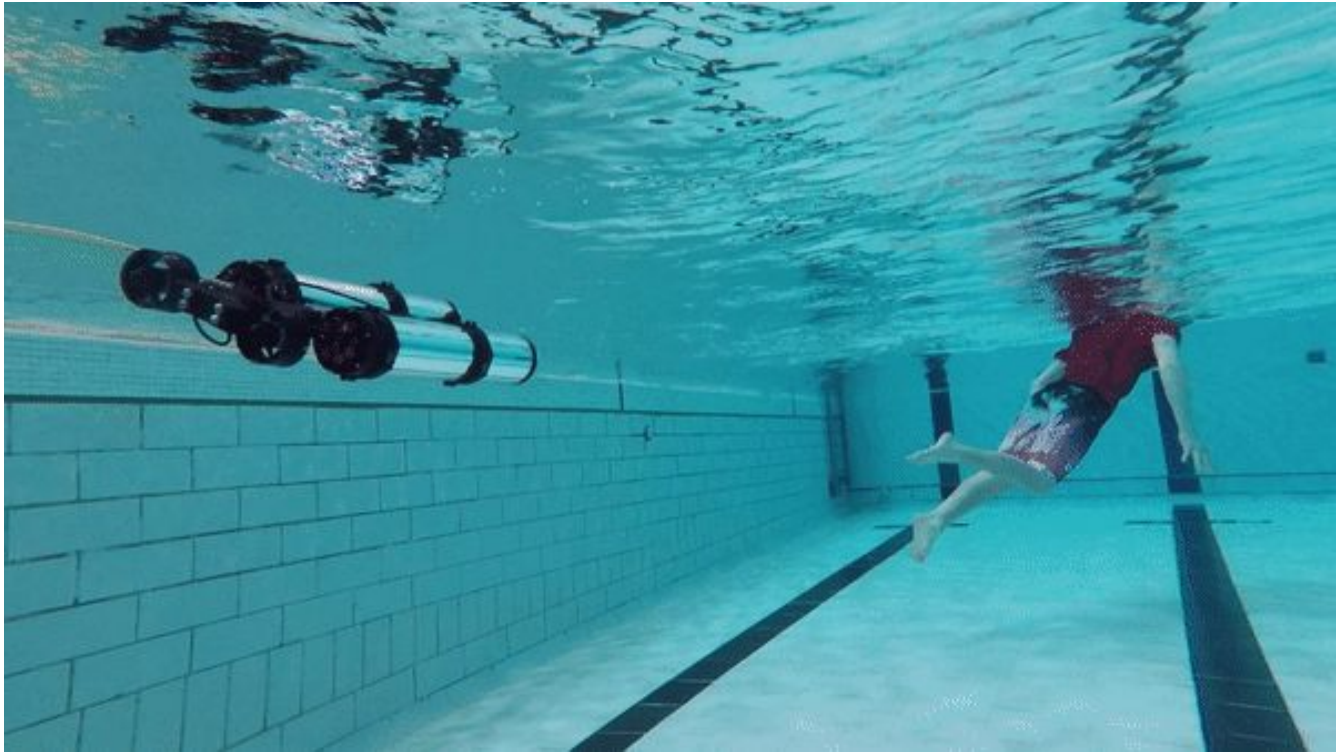
ARToolkit and RoboChatGest

Autonomous swimming triggered by
ARToolkit tag in Barbados



Autonomous swimming triggered by
RoboChatGest gesture in pool

Diver Following



Autonomous diver following in pool environment

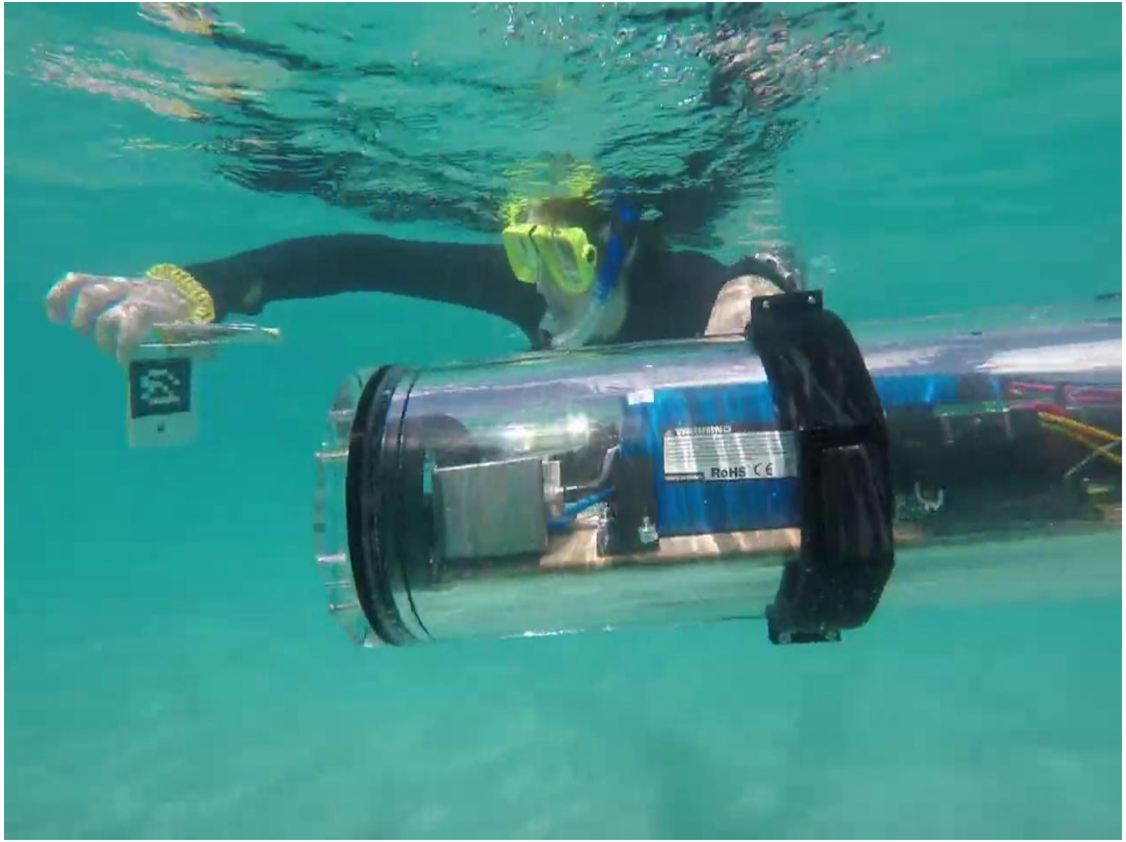
Deployment Methodology

- Pre-deployment process:
  - Connect electrical system, seal enclosures, create vacuum seal, connect thruster backbone to structure.
- Post-deployment process:
  - Rinse in freshwater, dry, disassemble substructure, remove vacuum seal, open enclosures, disconnect batteries.
- Successful Deployments
  - 7 Pool Deployments and MN Lake Deployment
  - 5 Caribbean Sea Deployments

Pumping air out of enclosures to create vacuum seal.

Pool Deployment

Ocean Deployment



Conclusion

- LoCO AUV is:
  - Low Cost
  - Open Source
  - Designed for HRI
  - Operable by small teams

- Successful deployments have been achieved in pool, lake, and ocean environments.

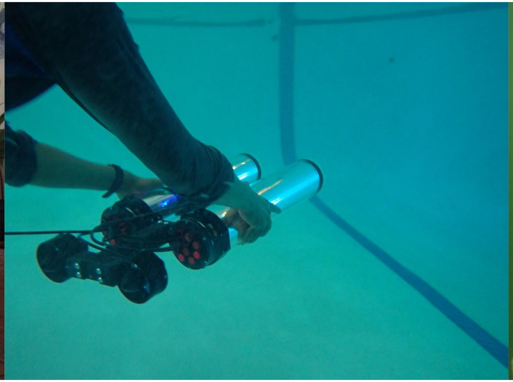
- Release of 1.0 version of hardware specs coming soon.

Acknowledgement

LoCO Project contributors:

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

Numerous others have provided suggestions throughout the design process.


[1]: Graduate students
[2]: Undergraduate students
[3]: Project and Lab director

Sponsors

NSF

MnDRIVE
Robotics, sensors and advanced manufacturing

MINNESOTA ROBOTICS INSTITUTE
UNIVERSITY OF MINNESOTA

irvlab.cs.umn.edu/other-projects/loco-auv

loco-auv.github.io