Subsea Resident Autonomous Underwater Vehicle

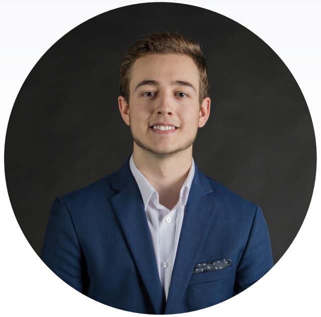
ECE Capstone Group 13

The Team

Mark Belbin

Electrical

Mechanical, Camera, and System Design



Mitchell Stride

Electrical

Electrical Systems Design

Francis Walsh

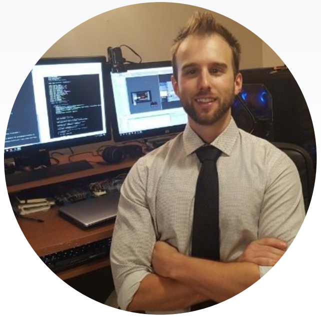
Electrical

Battery Systems Design

Andrew Nash

Computer

Software & Auto Pilot, Video Editor



Mark Duffett

Computer

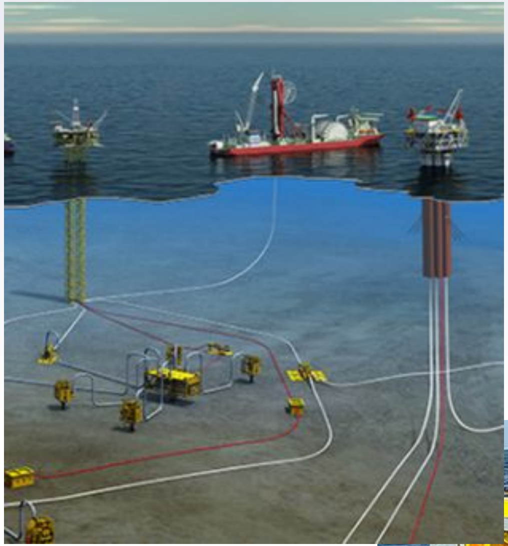
Software

Sim, GUI, SRAUV OS

Note: Mohamed Birama was an original Computer member of the team

Problem

- Subsea asset inspection for O&G
- Divers, ROVs AUVs
- Surface condition limitations
- Costly vessel / personnel overhead
- On demand only



https://dwcengineering.com/service/subsea-asset-management/

Client

- Autonomous Ocean Systems Laboratory of Memorial University
  - Dr. David Molyneux
  - Federico Luchino M.A.Sc

Supervisor

- Dr. Andrew Vardy of Memorial University

http://www.dofsubsea.com/rov/schilling-robotics-uhd/

Solution: Subsea Resident AUV

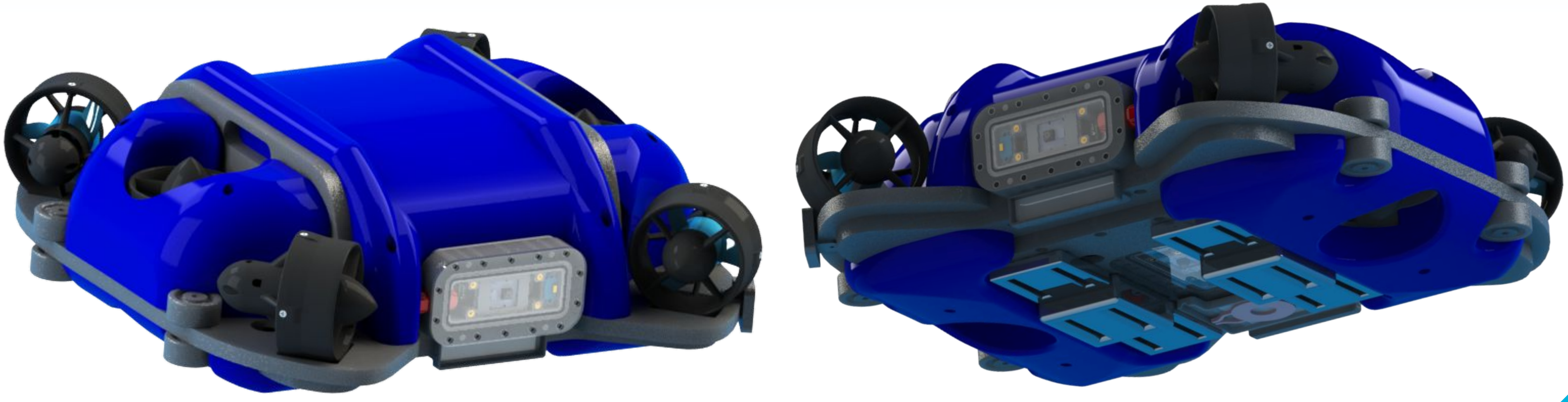
Solution: Goals & Scope

Primary Goals

- Perform an autonomous run with a custom designed AUV.
- Launch and recover the AUV autonomously.

Stretch Goal

- Wirelessly charge the AUV in the docking station underwater.

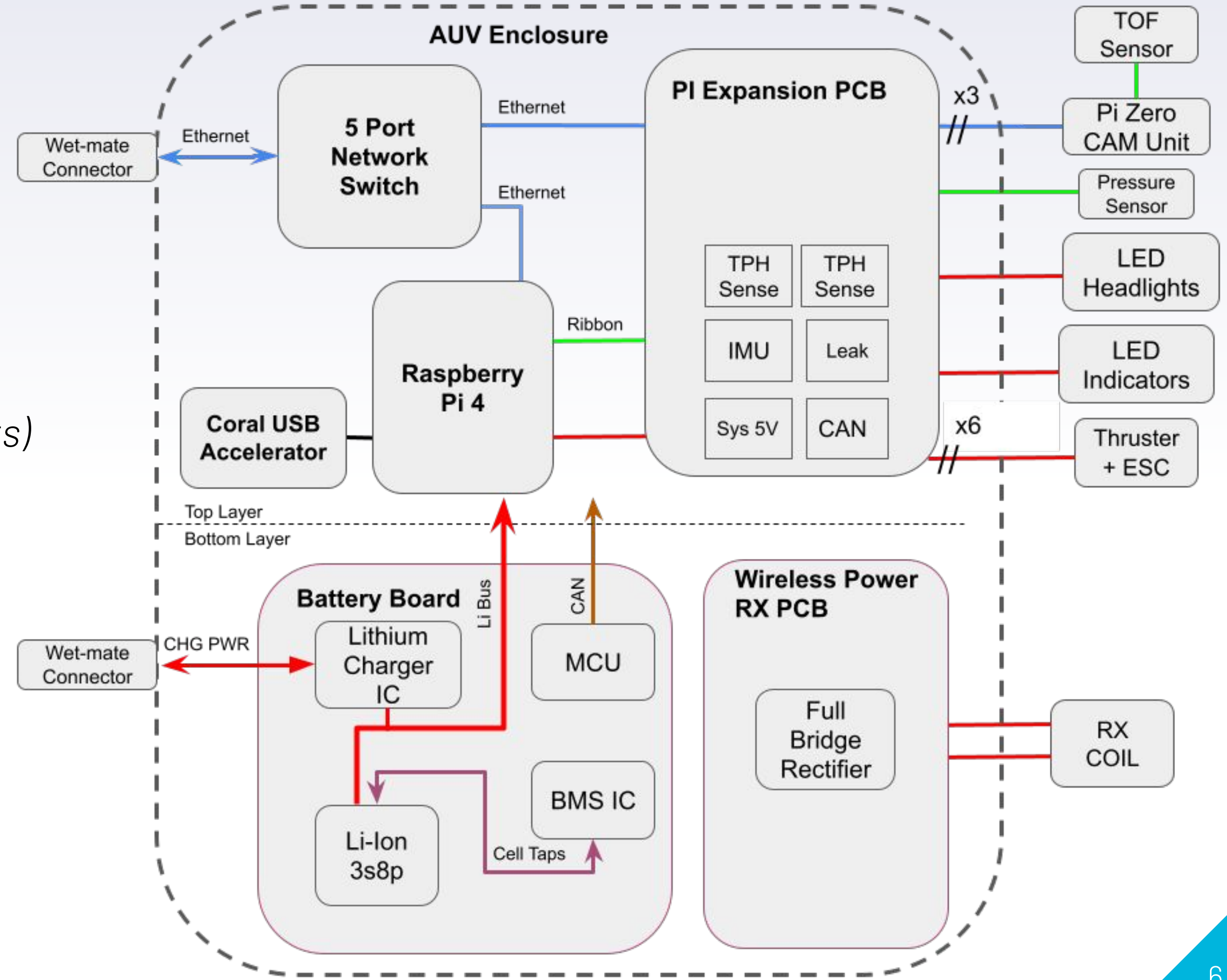


CAD Renders of the Final SRAUV Design

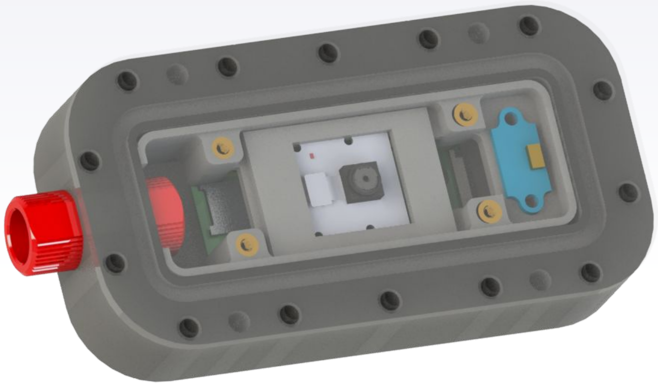
Electrical System Overview

System Features

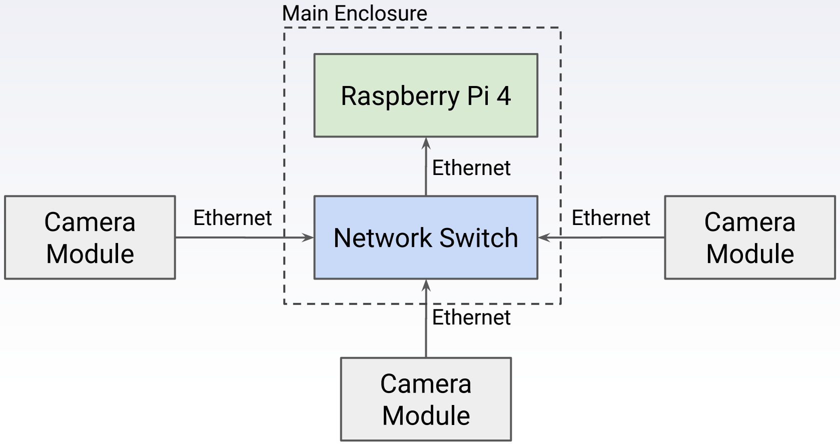
- *8 custom PCBs (3 Breakouts)*
- *6 Thrusters, 2 Lights, RPi4, Network Switch, Edge TPU Coprocessor (Coral)*
- *3 RPi Zero Camera Modules*
- *24 18650s Li-Ion Cells*
- *Wireless Power Transfer*



Camera System



Camera Module CAD Assembly



Custom Camera Module

- Raspberry Pi Zero W -> $10
- Raspberry Pi Cam V1 -> $13
- USB-to-Ethernet Adapter -> $13
- Plastic Stock / Bolts -> $14
  - TOTAL: **$50**

Already cheaper than many alternatives, such as USB cameras

Pi Expansion PCB

Design Features

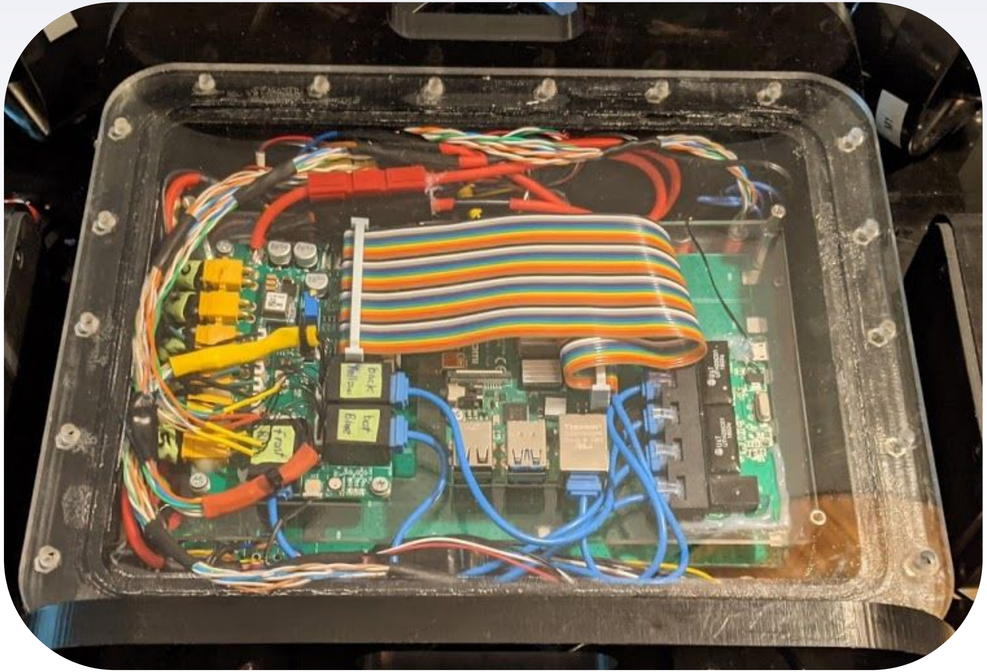
- 30W 5V Buck Regulator
- Temperature, Pressure, Humidity Sensing, IMU
- Thruster Power & CAN Network
- Pi Zero Camera Modules POE & Connectors
- Light Control

Justification

- Required to implement IMU, sensors, and other ICs
- Power Distribution
- Wiring Harness needed breakout and connectors

Pi AUV Expansion Board

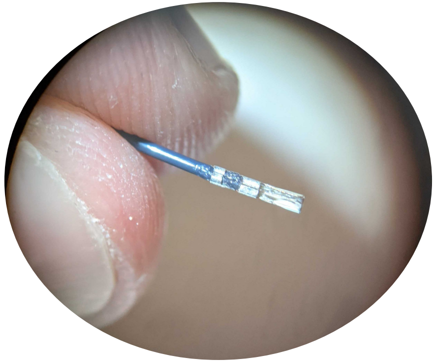
Wiring Harness & System Assembly



```
pi@raspberrypi:~ $ python wire_harness_test.py
Running SRAUV network and CAN tests...

Front Pi Zero Camera .50.............. FAIL
Back Pi Zero Camera .51.............. FAIL
Bottom Pi Zero Camera .52.............. FAIL

can0 inf status.............. PASS

Motor TestsSIOCGIFINDEX: No such device
Motor M1-208 .............. FAIL
Motor M2-248 .............. FAIL
Motor M3-288 .............. FAIL
Motor M4-2C8 .............. FAIL
Motor M5-308 .............. FAIL
Motor M6-348 .............. FAIL
Finished.
```

Issues

- 1mm pitch connectors w/ bad wire



Battery System Design

Design Constraints: Tether & Size

Cell Chemistry Selection:
Power Density and Energy Density tradeoff.

- Lead Acid
- Nickel Metal Hydride
- Lithium Ion

Battery System Components

- Battery Management System
    - Current Transducers
    - Multiple Relays
    - Balancing Resistors
- Charging Circuit
- Protection Circuitry

OTS Battery Management System

OTS Lithium-Ion Battery Charger

Output Protection

Battery System

Custom Battery Features:

- Over/Under Temperature/Voltage Protection
- CANBus Telemetry
- Autonomous Cell Balancing

Battery System Specifications

Battery Config.	3s8p (24 cells)
Max Voltage	12.6V
Nom. Cont. Power	~250W
Burst Power (30s)	500W
Energy Capacity	195Wh
Flight Time	~3 Hours
Standby Time	+10 Hours

Dock Charging

How to charge battery system underwater?

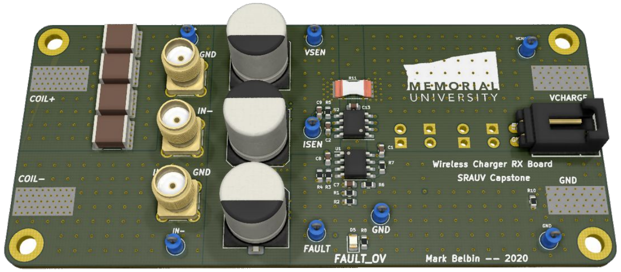
1. Mechanical Wet-mate Connector
2. OTS Wireless Power
3. Custom Wireless Power

Wireless Charging Specifications

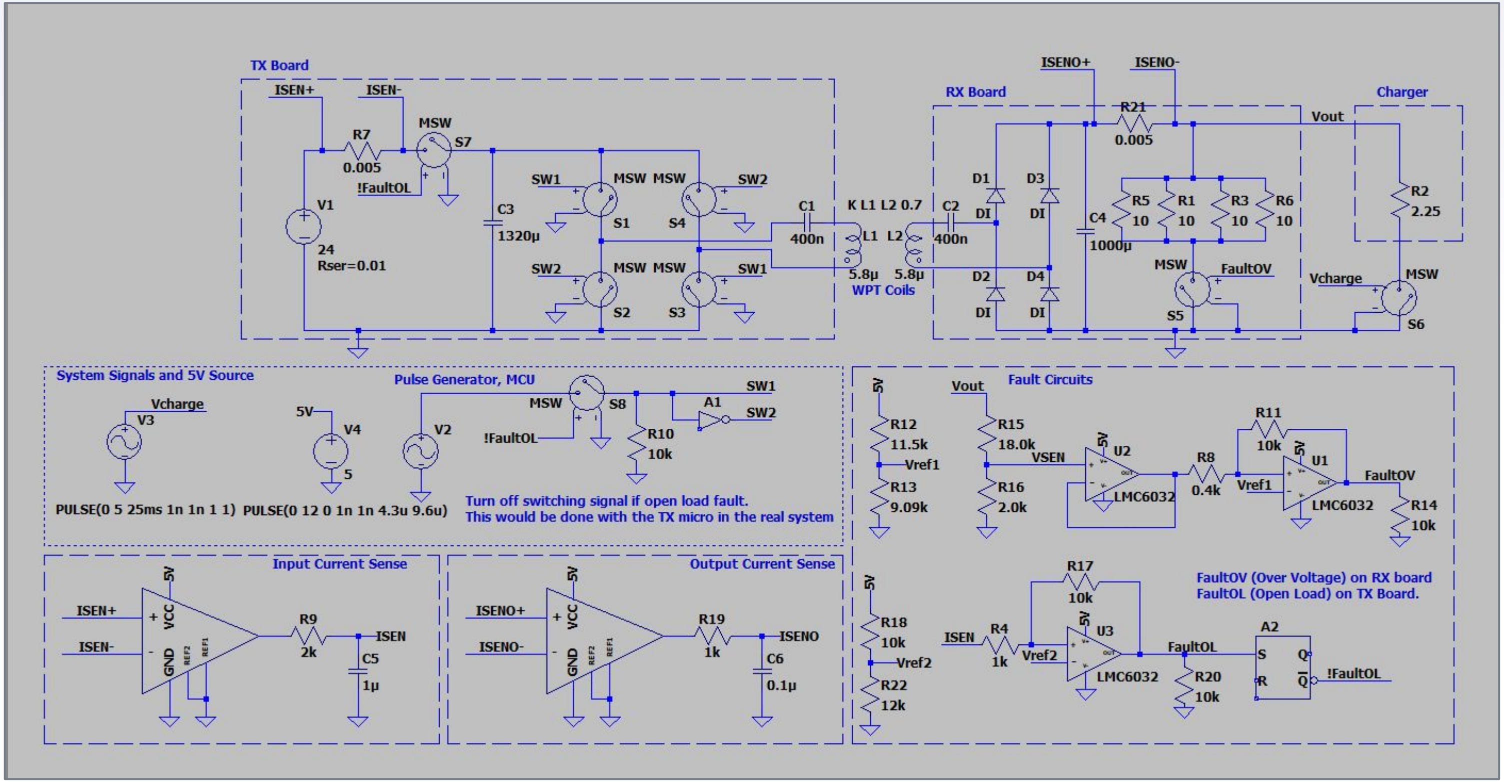
Input Voltage	24V
Max Current	3A
Output Voltage	18V
Max Power	~50W
Operating Freq.	300-100 kHz
Charge Time	~ 2.5 Hours

TX Board

RX Board



Wireless Power Simulation



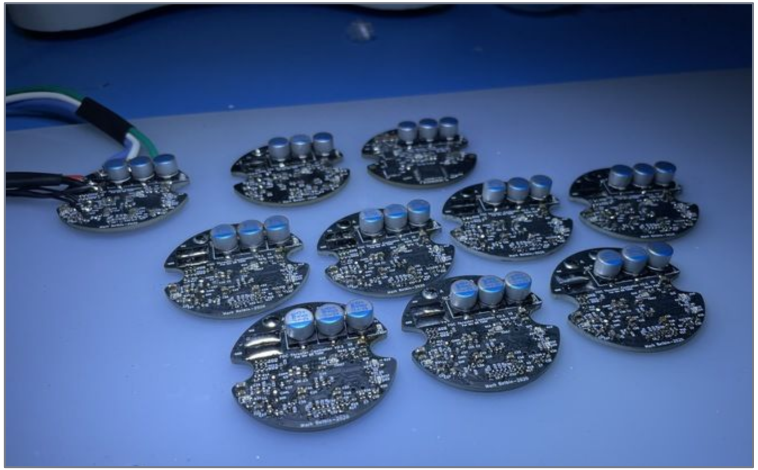
LTspice Wireless Power Schematic

Thrusters

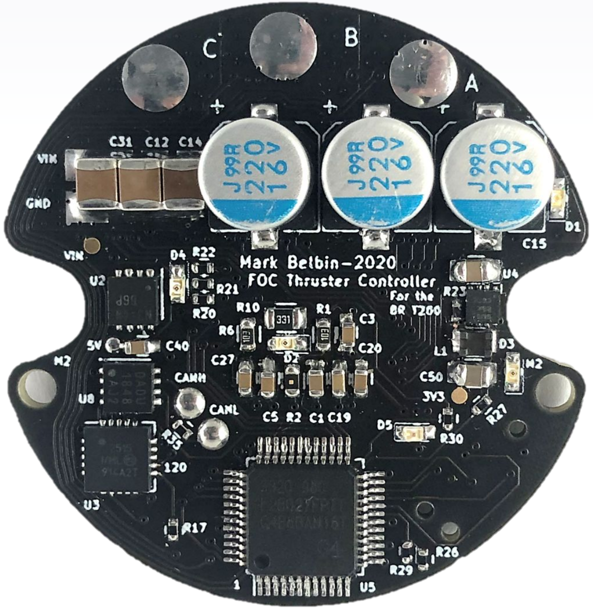
Assembled motor controller on T200 thruster.

While not explicitly created for this capstone, these motor controllers were finished and integrated into our AUV.

They are based on FOC, providing 5-15% better efficiency, lower audible noise, and smoother low speed operation, *perfect for AUVs!*



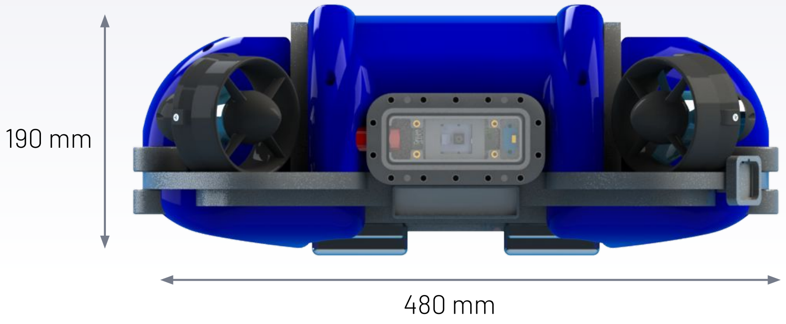
Motor controller PCBs during assembly



Motor controller PCB, size of a poker chip

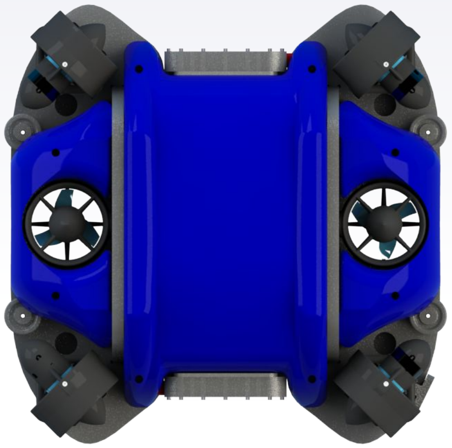
Mechanical Design

Front View



Side View

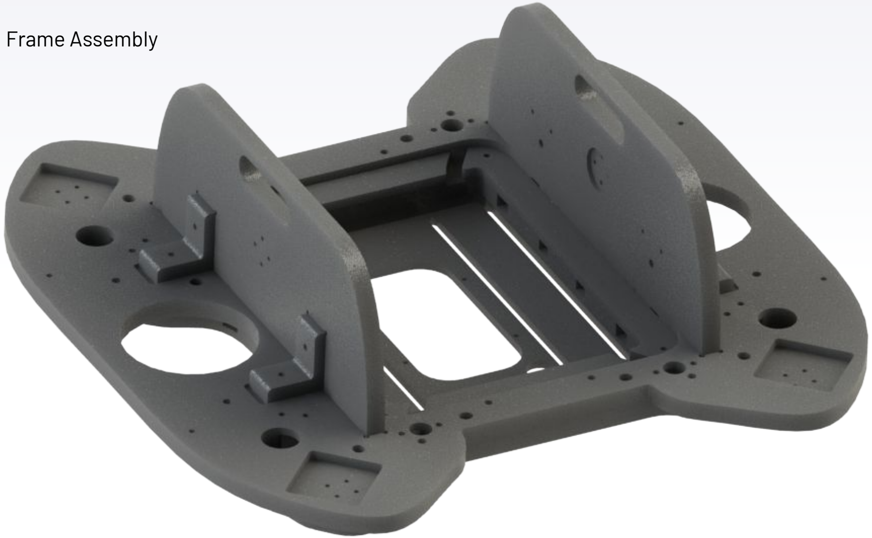
Top View

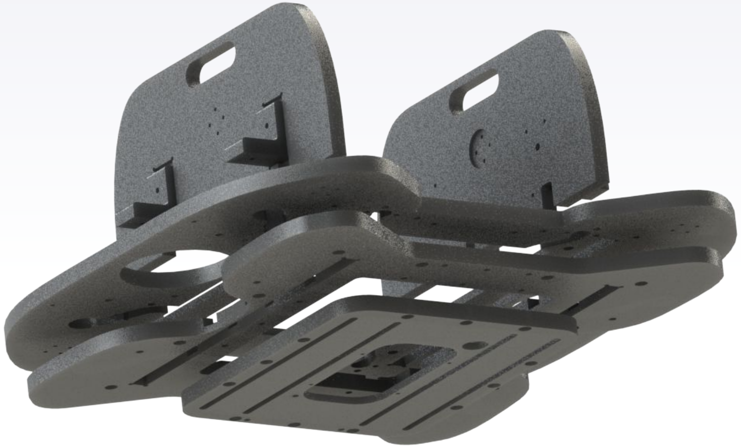


Dimensions	190x480x480 mm
Mass	~15 Kg

SRAUV Frame

Frame Assembly





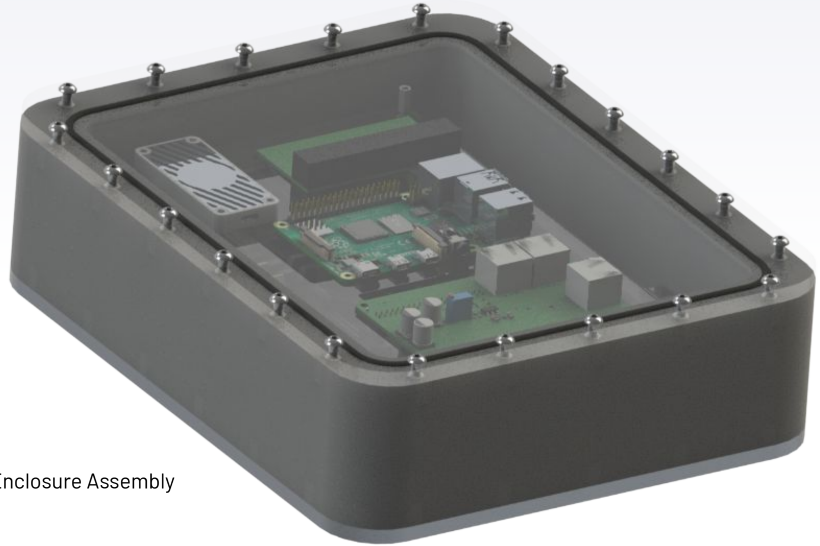
Frame Assembly Exploded

The AUV frame is constructed from 0.5″ HDPE stock. HDPE is cheap, sturdy, chemically inert, and neutrally buoyant.

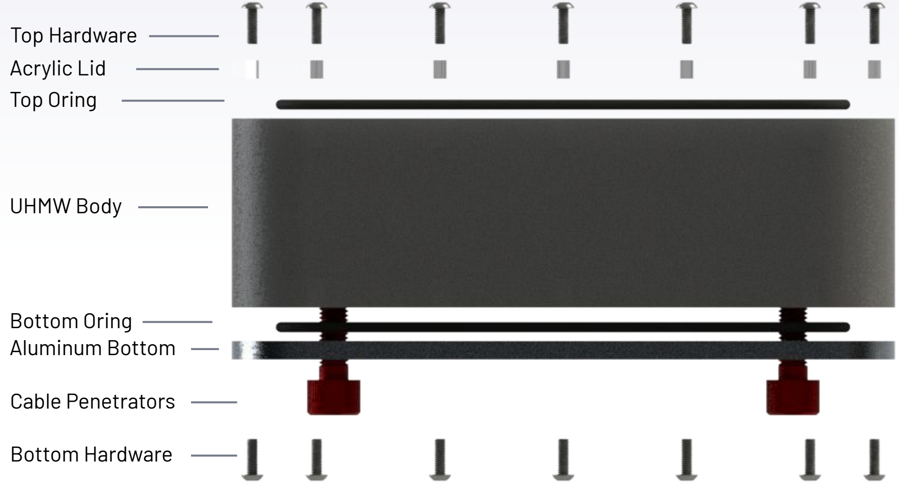
Perfect for underwater robotics.

Total cost: $160
Total Weight: 3.3kg

Electronics Enclosure



Enclosure Assembly



Enclosure Assembly Exploded

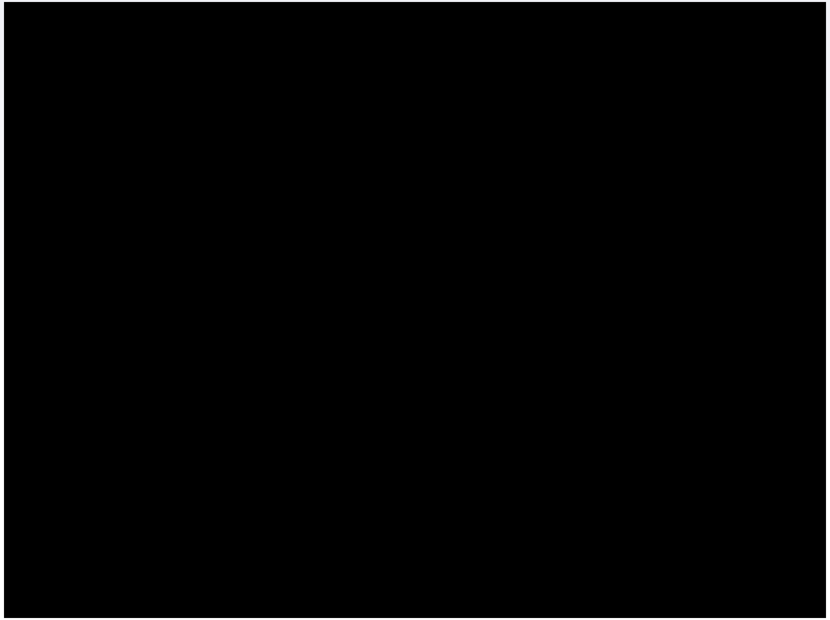
The enclosure consists of a tri layer of UHMW, acrylic, and aluminum. It's rated for depths up to 20 ft, and includes a double oring seal.

Total cost: $220
Total Weight: 2.5kg with no electronics

CNC Machining

Buoyancy & Stability

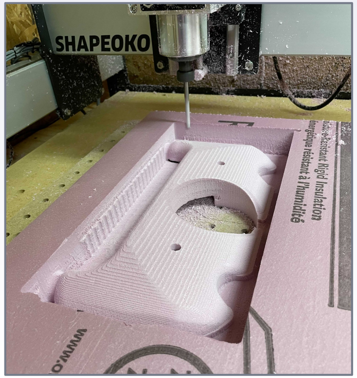
Buoyancy Blocks



Buoyancy blocks are essential in order to create 'neutral-buoyancy' in water, making the AUV essentially weightless.

Weight was attached to the bottom of the vehicle to increase stability, while buoyancy attached to the top.

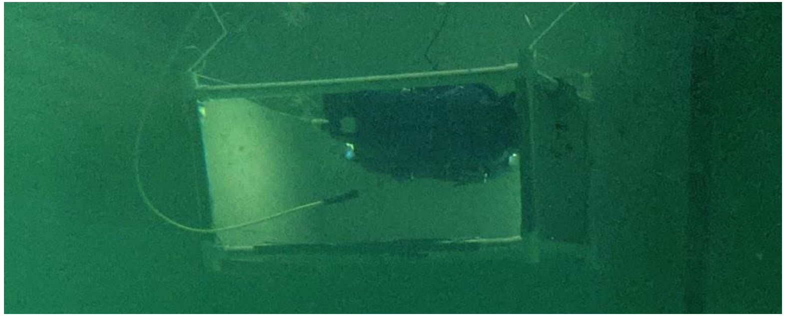
Foam Construction



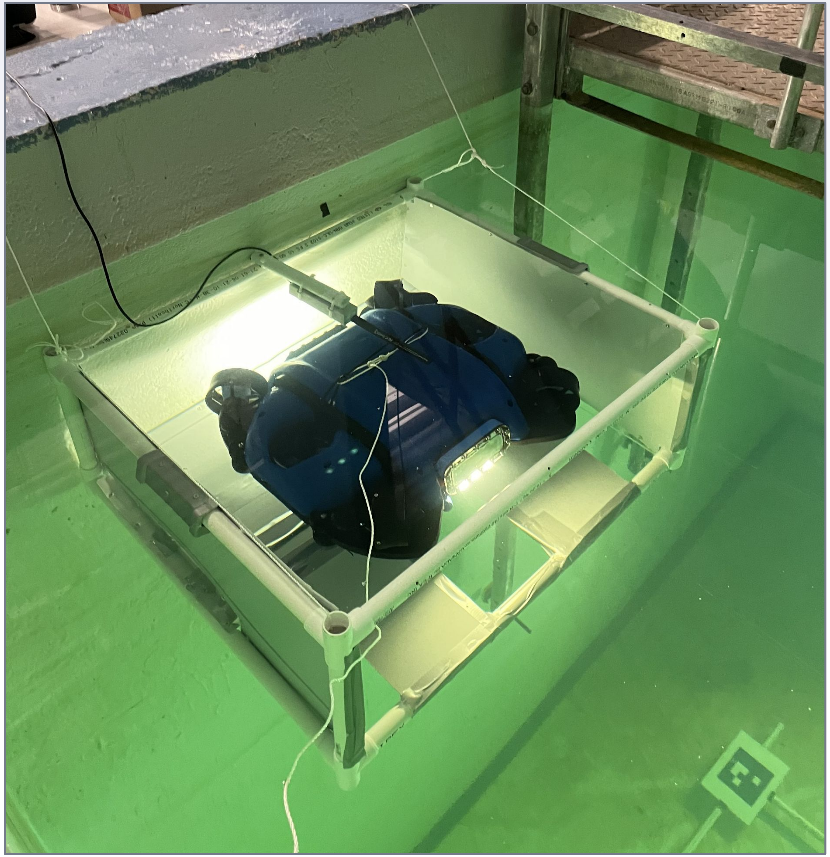
Dock Design

Our dock design used cheap and local materials.

The AUV can communicate with the surface station over WiFi while in the dock.

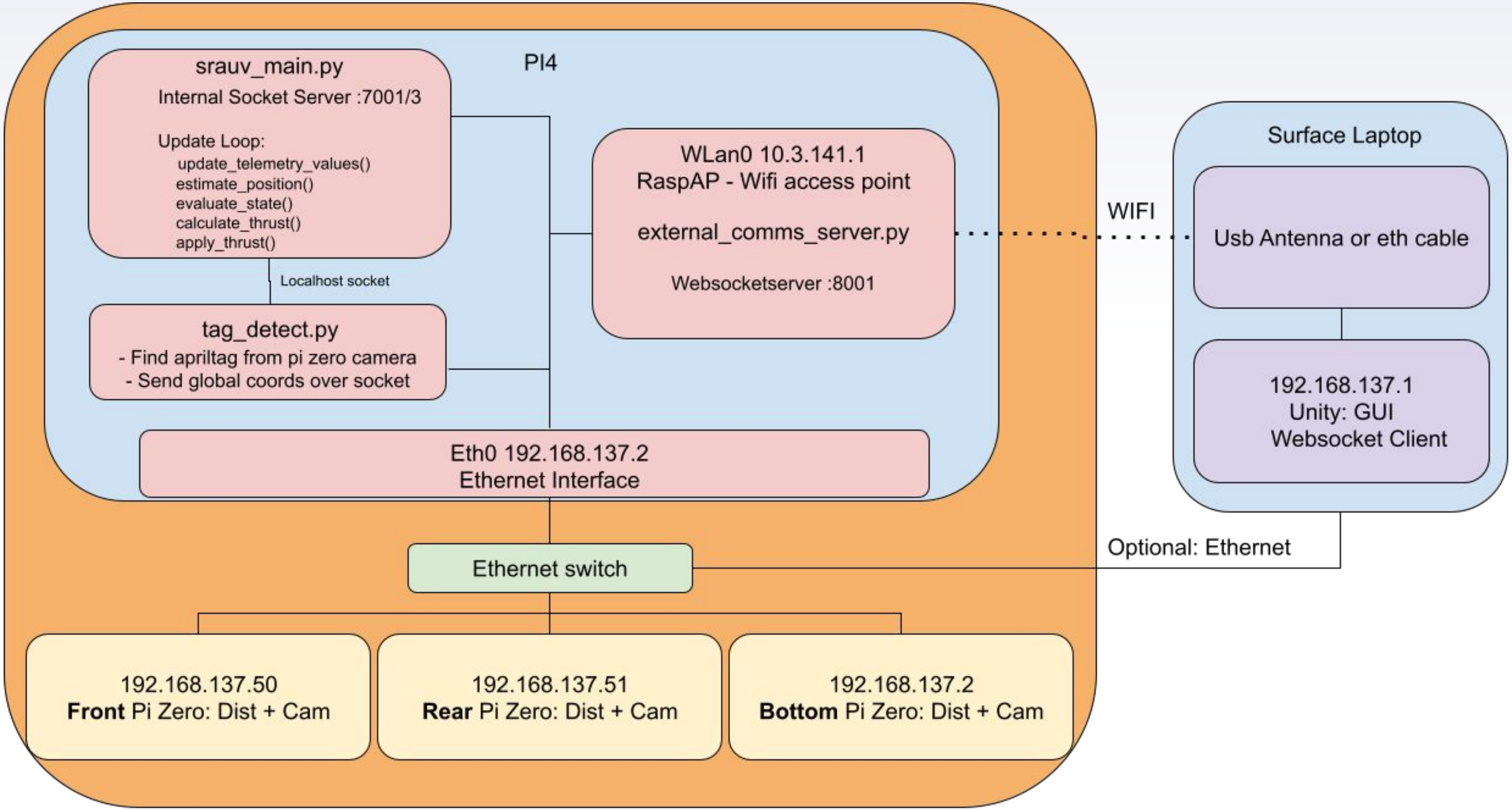


Dock Front View in Tank



Dock Angled View in Tank

Software Design

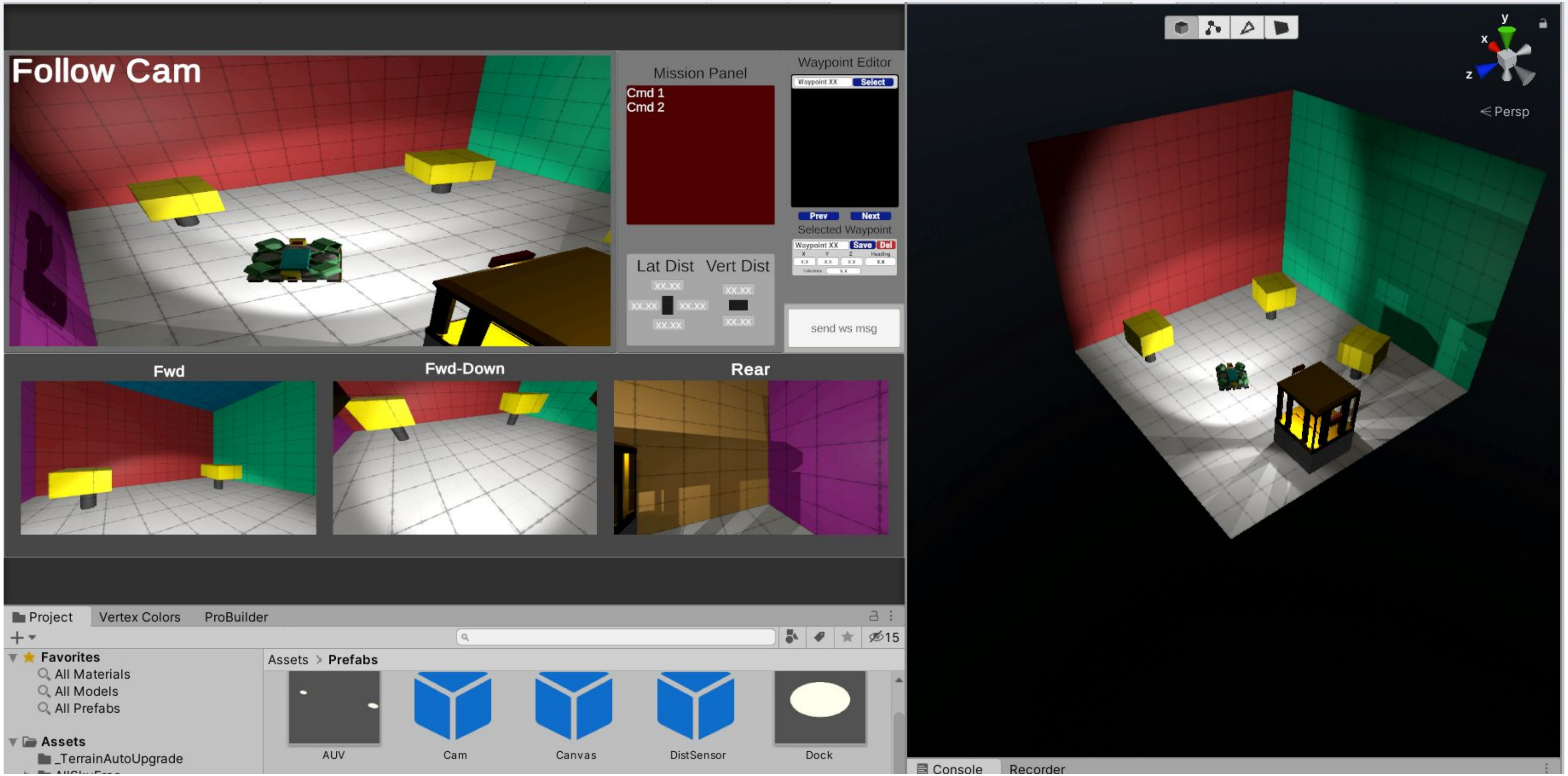


Software Examples

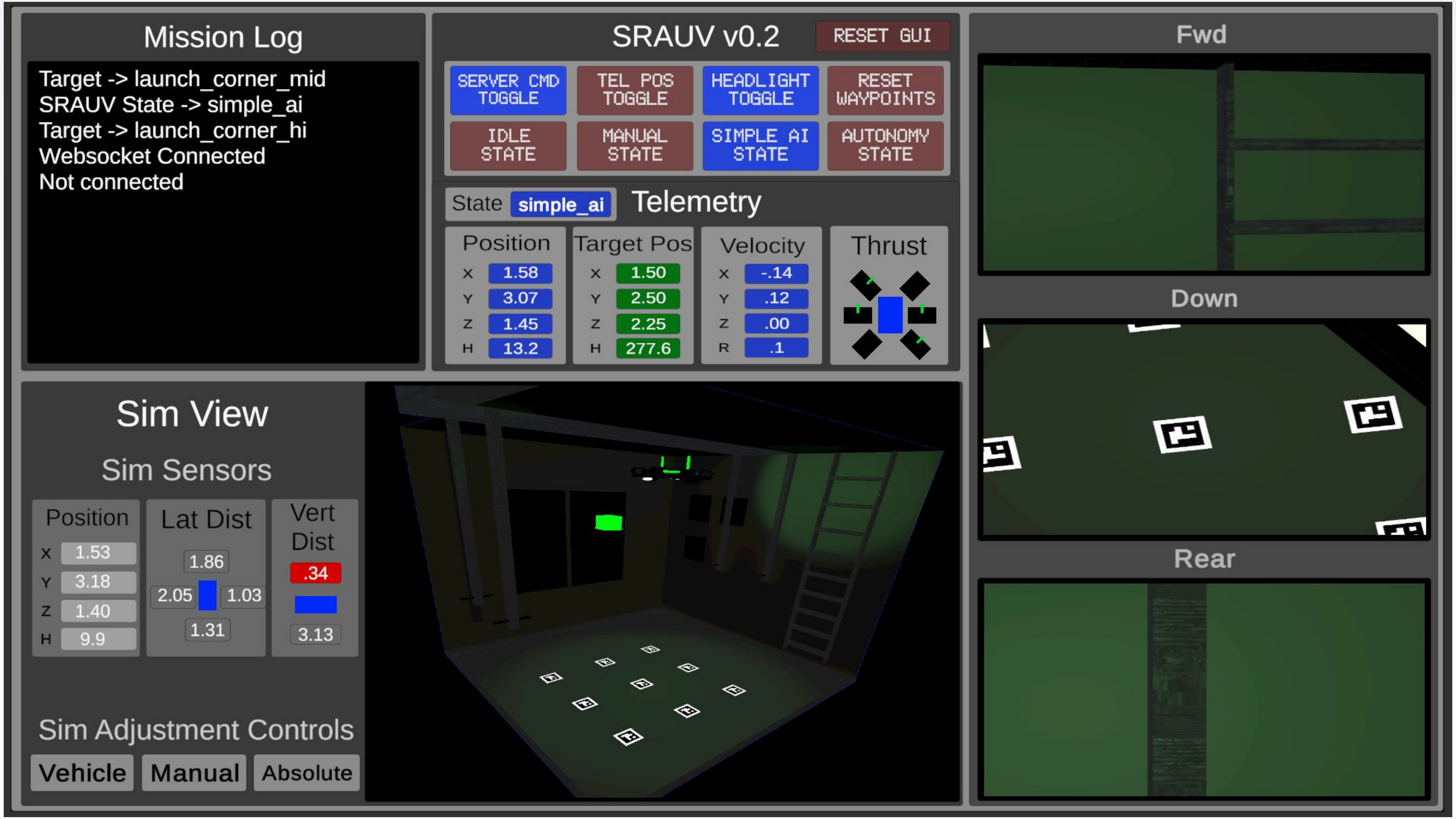
```python
g_logger.info(f'state:{g_tel_msg["state"]} MSG:Starting update loop')
while True:
    try:
        time_now = timestamp.now_int_ms()
        if time_now - last_update_ms >= SETTINGS["update_interval_ms"]:
            ul_perf_timer_start = perf_counter()

            parse_received_command()
            update_telemetry()
            srauv_waypoints.estimate_position(g_tel_msg)
            evaluate_state()
            calculate_thrust()

            # update loop performance timer
            ul_perf_timer_end = perf_counter()
            g_logger.info(f'state:{g_tel_msg["state"]} update loop ms:{(
            last_update_ms = time_now

            # log in blocks for clarity
            g_logger.info(f"thrust_vals    : {g_tel_msg['thrust_values']}
            g_logger.info(f"tel msg        : {g_tel_msg}")
            g_logger.info(f"update loop ms: {(ul_perf_timer_end-ul_perf_
```

```json
"hardware":{
    "can": false,
    "i2c": false,
    "headlights": false,
    "leds": false,
    "coral": false
},
"fly_sim": true,
"sim_sensor_noise": false,
"log_to_stdout": false,
"external_ip": "",
"external_port": 8001,
"internal_ip": "localhost",
"main_msg_port": 7001,
"main_cam_port": 7002,
"main_tag_port": 7003,
"update_interval_ms": 50,
"default_comms_response_str": "dflt_resp",
"manual_deadman_timeout_ms": 5000,
"tag_stale_timeout_s": 0.500,
"tag_kill_timeout_s": 10.0,
"dist_sensor_config":{
    "total_sensors": 5,
```

Simulator Design v0.1



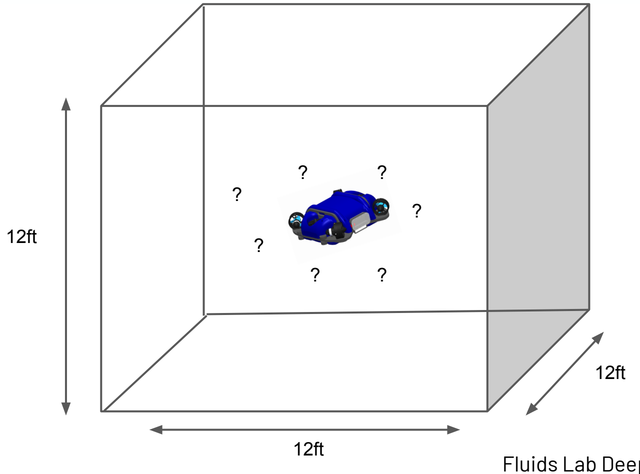
Simulator Design v0.2



AUV Localization

A central problem of autonomous movement is knowing where you are and where you need to go.
It's critical we have a good estimate of where the AUV is in relation to the tank.

Our initial solution was:
- Pressure sensor for depth, Z.
- ToF sensors for distance, X/Y.
- IMU for heading, yaw.



Fluids Lab Deep Tank Dimensions

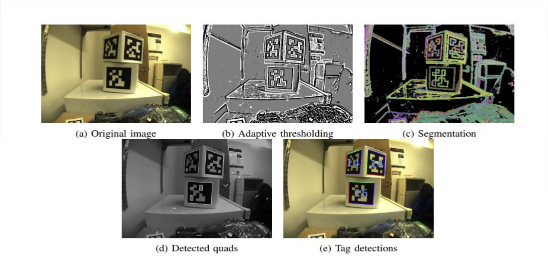
However:

- Our ToF sensors were mostly DOA, bad batch of breakout boards.
  - Still, after finally testing in water, they did not reach farther than 10 cm
- Our IMU heading drifted heavily in water due to its magnetometer.
- And our depth sensor broke

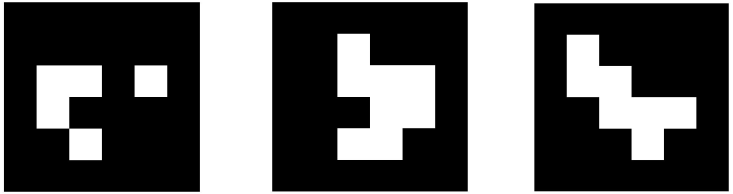
So what do we do now?

AUV Localization

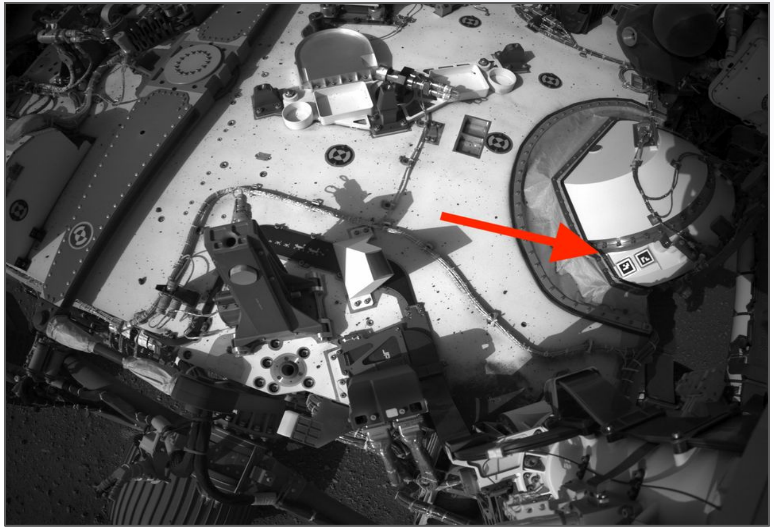
To solve our localization problem late into the project, we decided to use **AprilTags.** [1] With these tags, we can get our estimated position using **computer vision.**



April Tag algorithm steps, from [2]



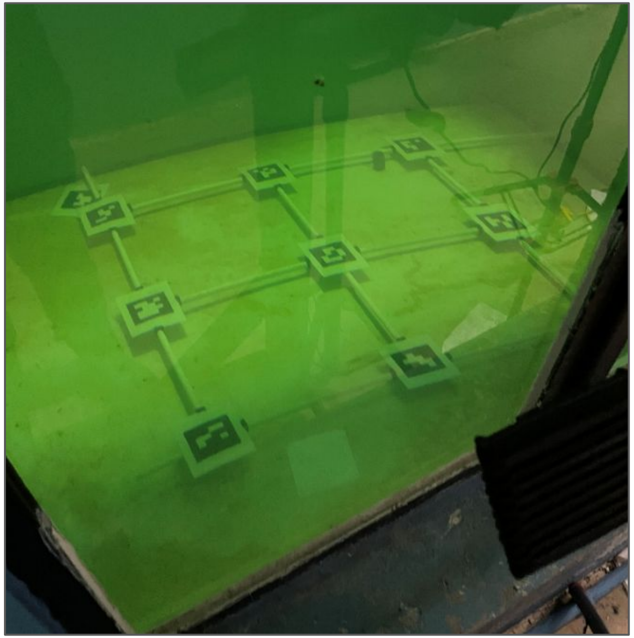
Sample AprilTags from the 16h5 tag family.



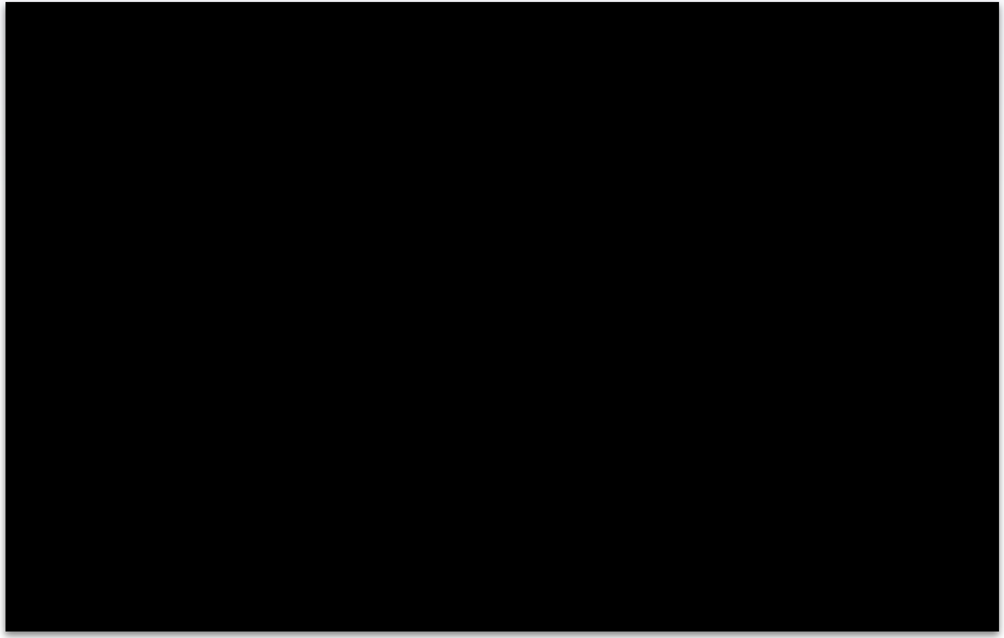
April Tags used on the latest Mars Perseverance Rover, using the same 16h5 tag family. [4]

April Tag Grid

- Since the AUV always has a bottom view of its surroundings, a grid of april tags on the bottom of the tank would enable localization.
- A grid of tags spaced 0.75m apart was determined to be the best size based on our camera's FOV.



Constructed AprilTag grid at the bottom of the Deep Tank.

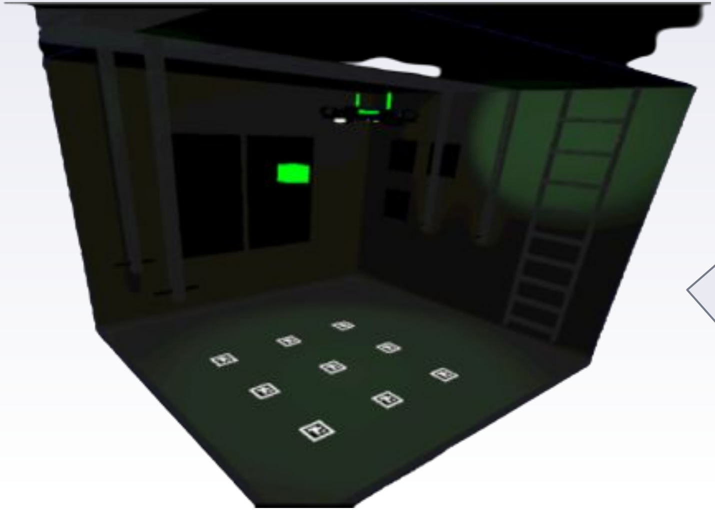


AUV simulator with the 3x3 grid of tags.

Reinforcement Learning

Physics

unity



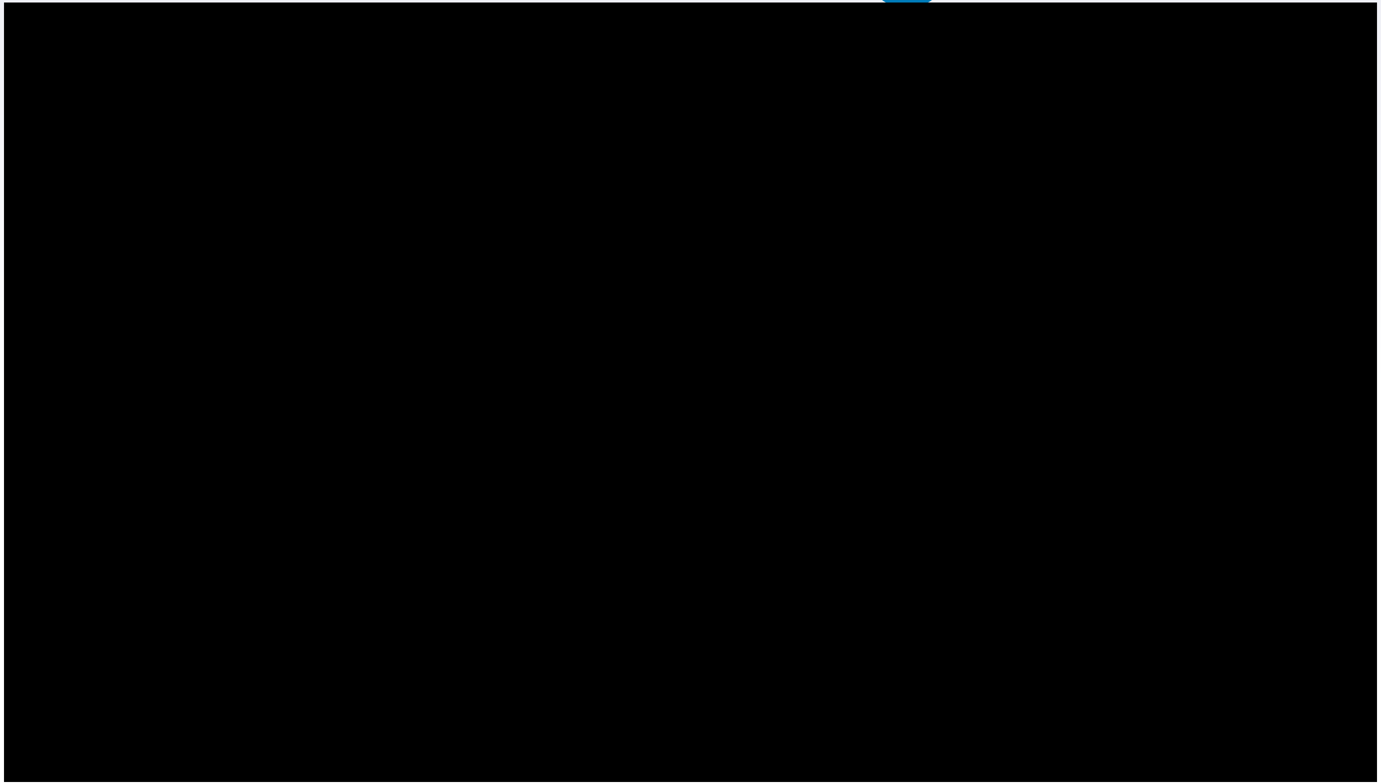
Environment

Action(s)

State & Reward

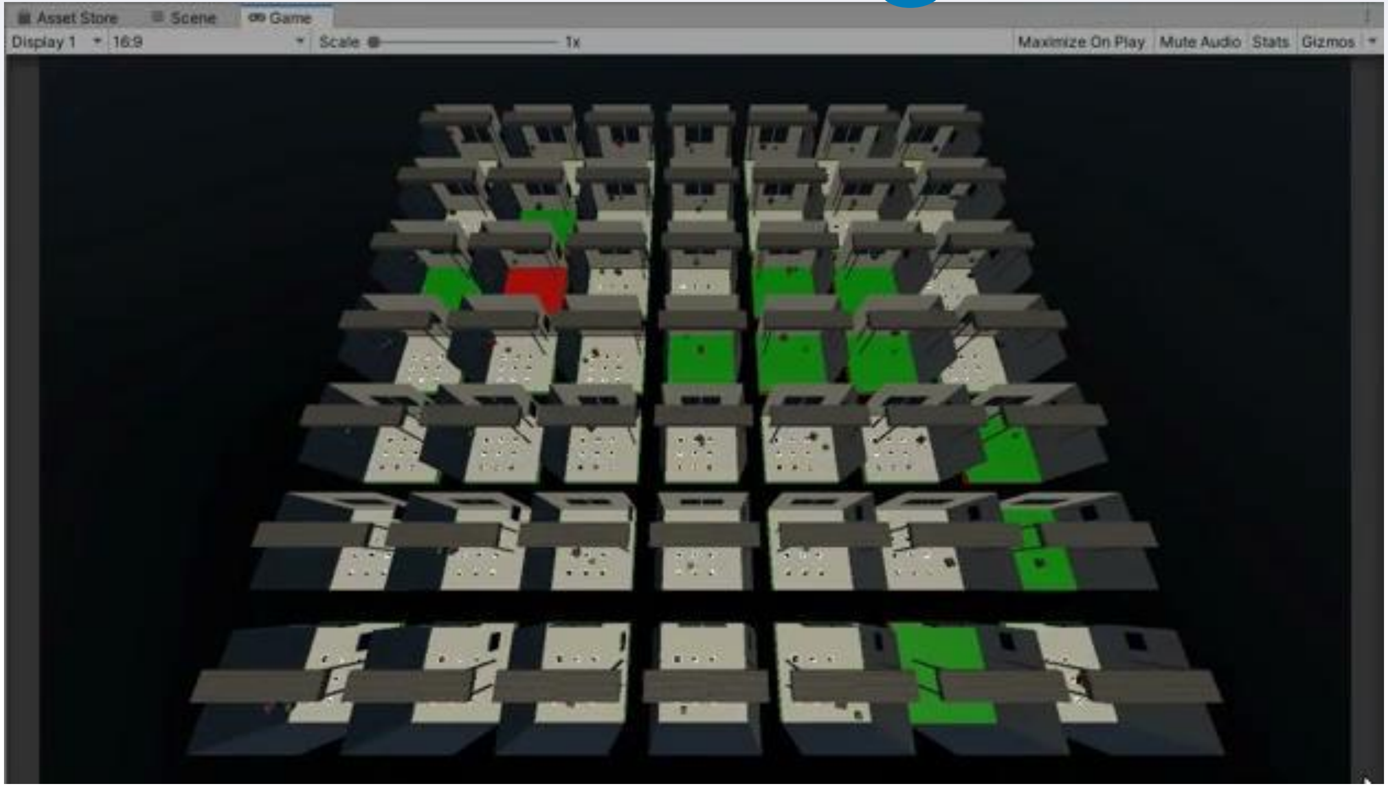
Agent

Auto Pilot Training



Green = Success, got to target
Red = Failure, hit tank wall

Auto Pilot Training

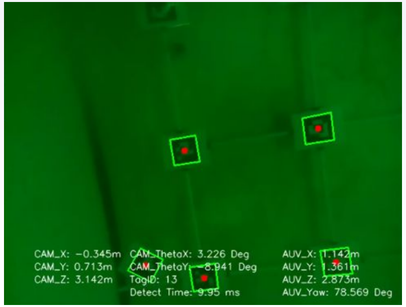


Green = Success, got to target
Red = Failure, hit tank wall

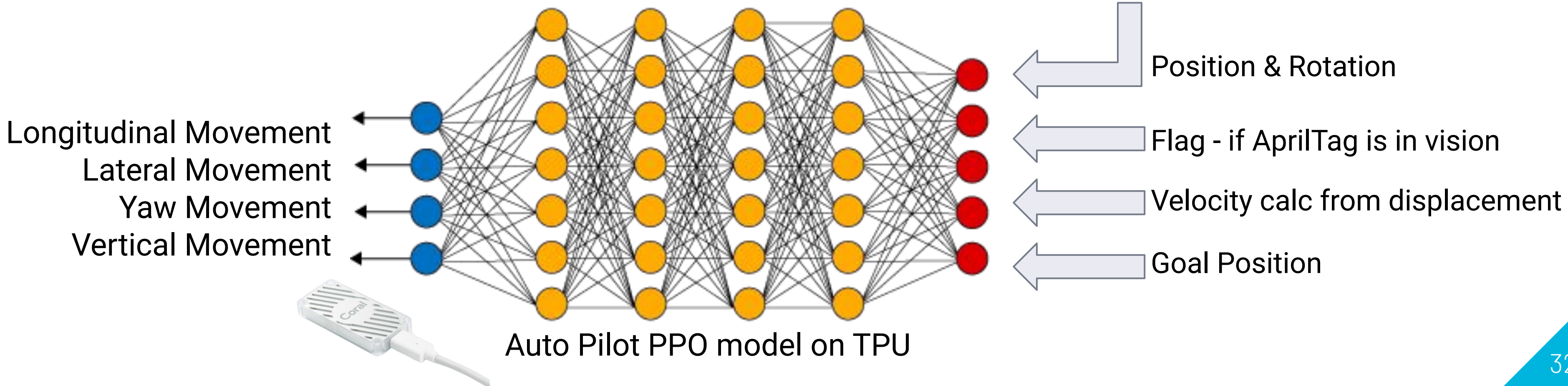
End to End Auto Pilot

Stream Bottom Camera

Grey Scale Image

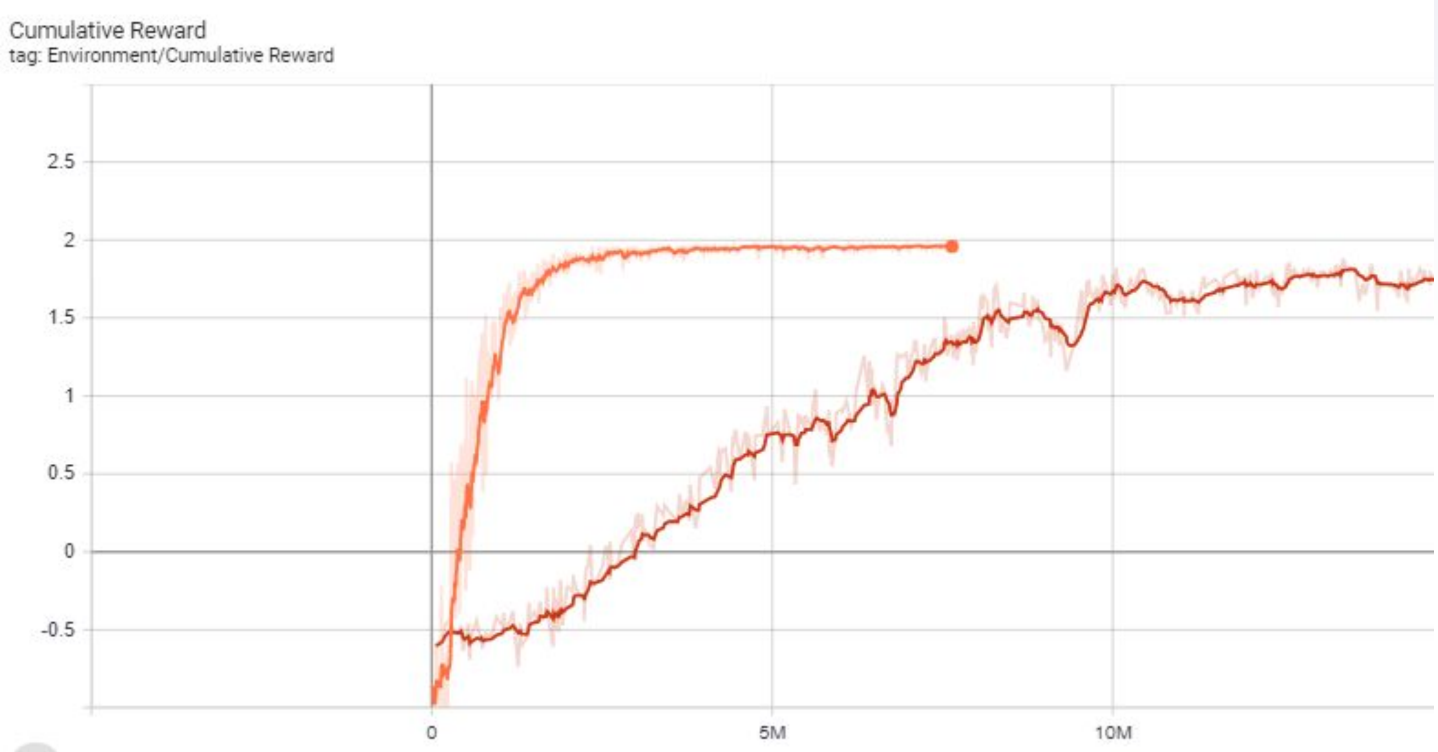


Localize from AprilTags



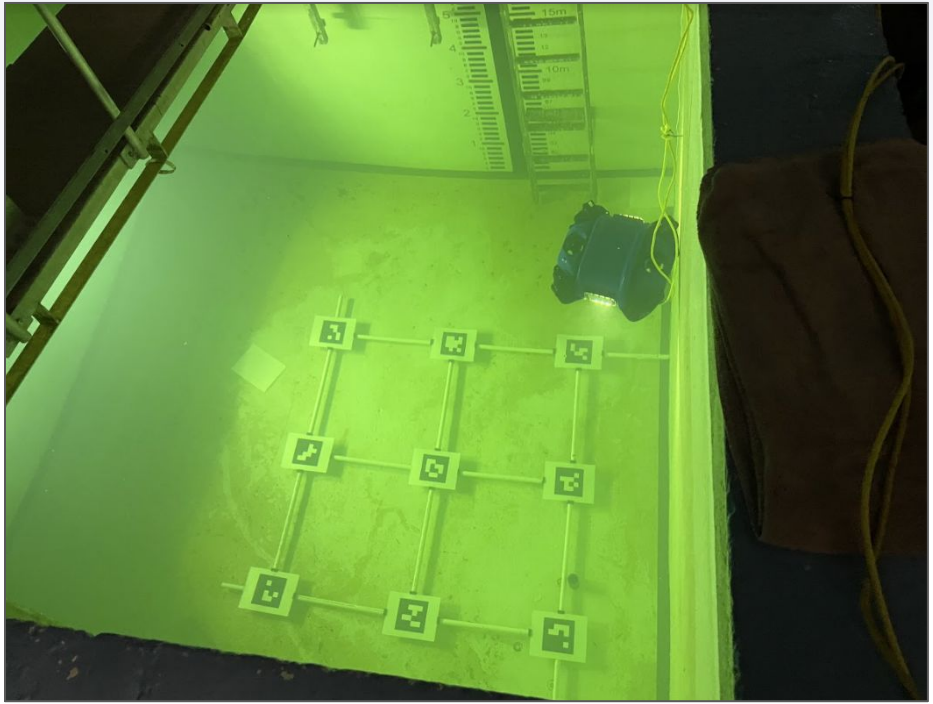
Auto Pilot PPO model on TPU

AprilTag Disadvantage



Orange = DVL (known position and velocity at all time)
Red = AprilTags (Noisy position sometimes, no velocity)

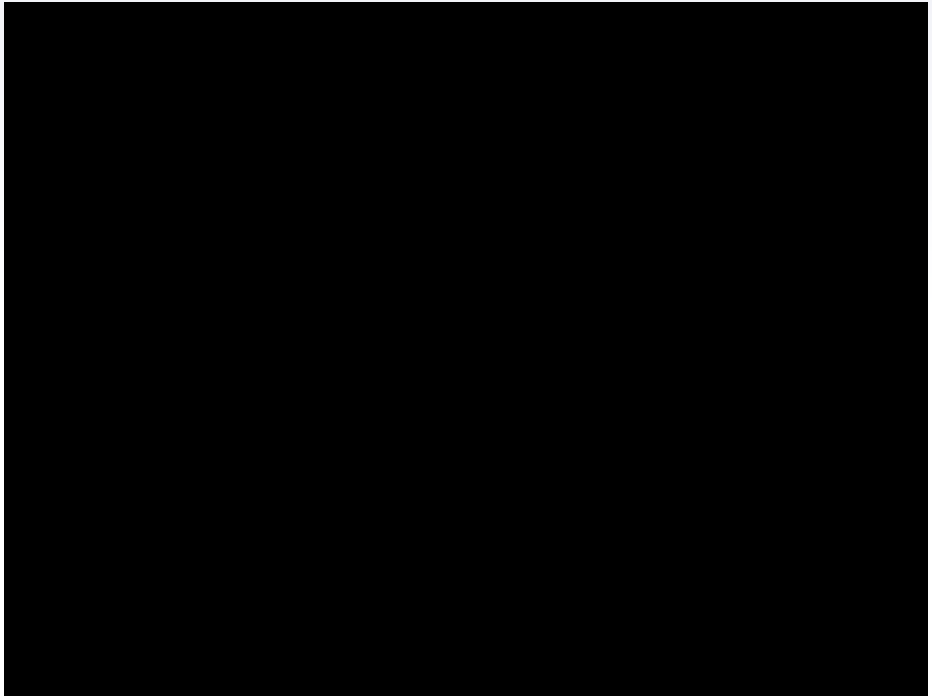
Tank Testing

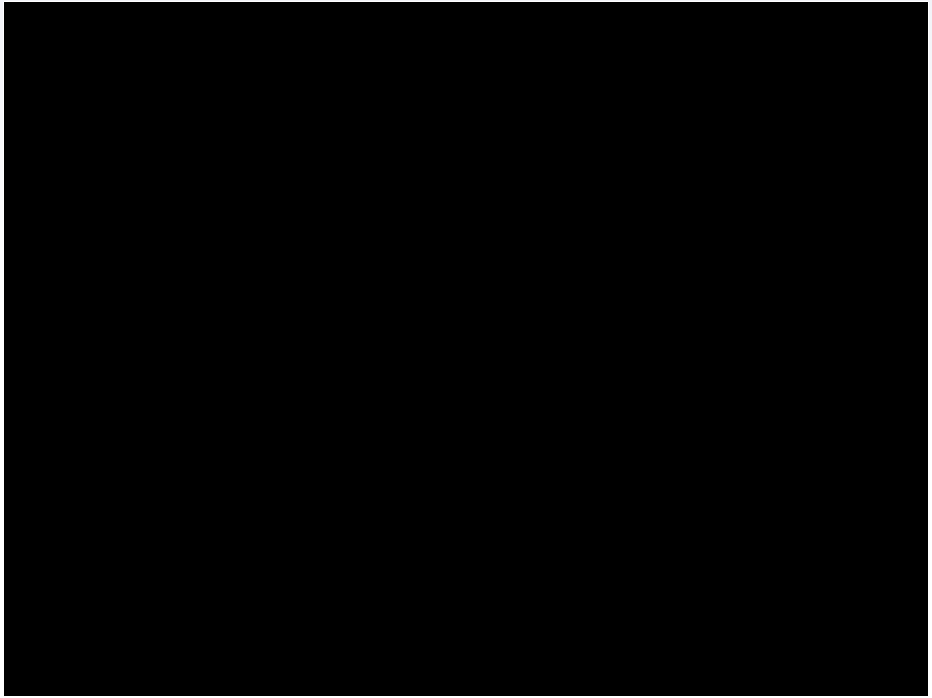


AUV using the AprilTag grid for localization.

Team photo, missing Frank and Mo. Taken Feb. 02, 2021

March 18 Autonomous Runs

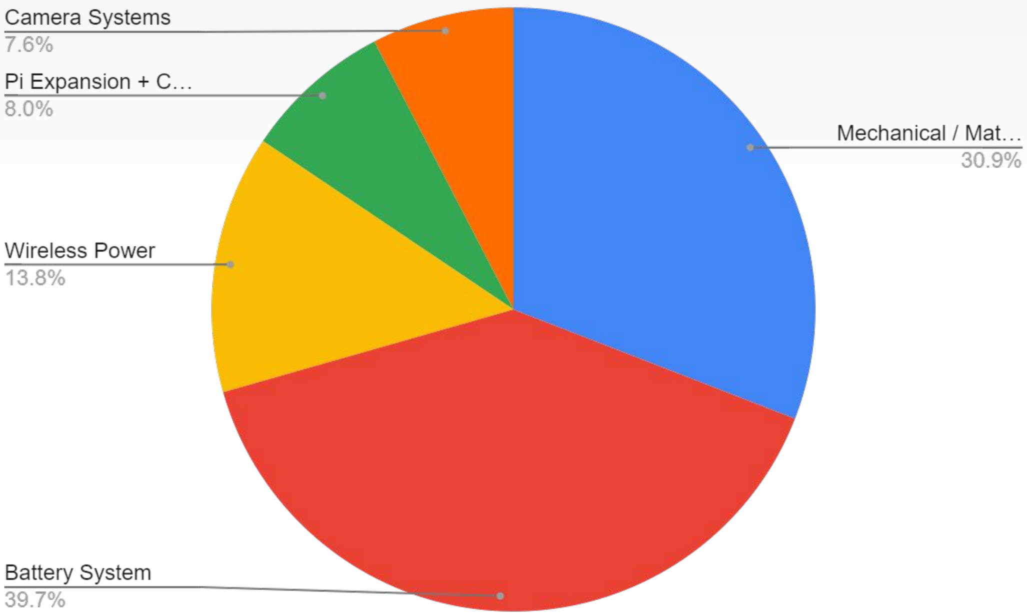




Budget & Funding

Budget Breakdown

Source/Sink	Value
Standard Course Funds	+$353.78
IEEE Special Grants	+$1200.00
Donated / Borrowed Materials	+$2000.00
Mechanical / Materials	-$778.32
Battery System	-$1000.2
Wireless Power	-$348.67
Pi Expansion + Coral	-$200.42
Camera Systems	-$192.15



Special Thanks

- Client: AOSL, Dr. David Molyneux and Federico Luchino
- Supervisor: Dr. Andrew Vardy
- Dr. Dennis Peters
- Fluids Lab Technicians: Trevor Clark, Craig Mitchell, Matt Curtis
- Memorial University
- IEEE

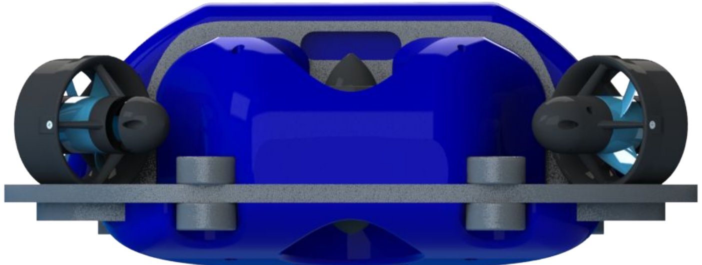
Prototype Results

- Results:
  - Electrical systems worked as intended.
  - Coral TPU + Pi Threads exceeded software efficiently expectations.
  - PPO reinforcement learning was able to solve underwater driving.
- Able to complete an autonomous run and redock!

- Next Steps:
  - Improved XYZ Localization and Velocity (pseudo lidar)
  - Improve simulator realism (drag / buoyancy)

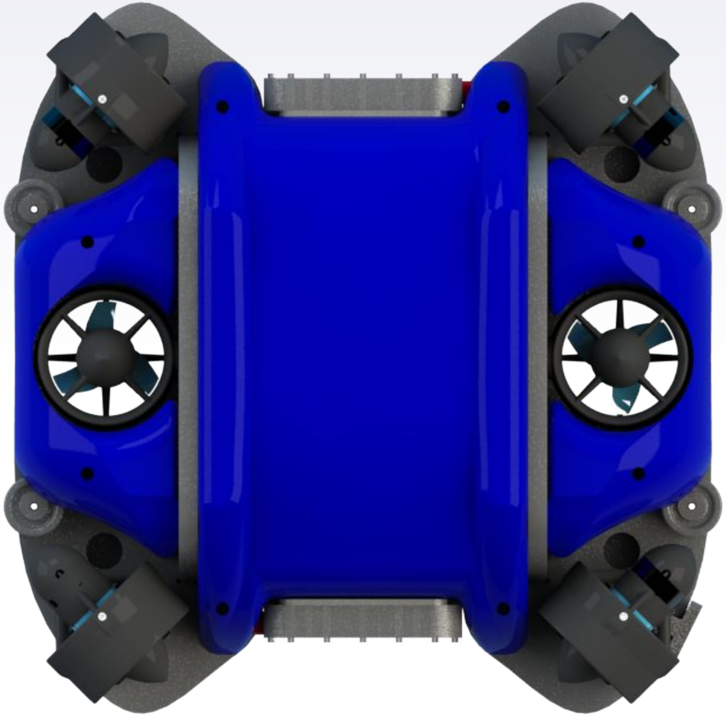
Recommendations

- ML models on a TPU can work for subsea autonomous flight
- Vision recognition is possible for localization underwater
- Wifi for network data transmission is possible for at least 10 cm
- Wireless Power needs testing
- Subsea Resident AUVs are a possibility



THANKS!

Any questions?



References

Theme: https://www.slidescarnival.com/gaoler-free-presentation-template/9122

[1] E. Olson, "AprilTag: A robust and flexible visual fiducial system," IEEE International Conference on Robotics and Automation, May 2011, pp. 3400-3407.

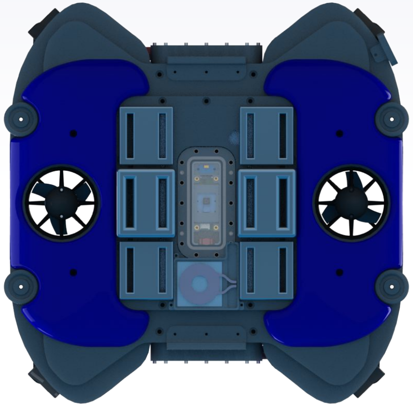
[2] E, Olson and J. Wang, “AprilTag 2: Efficient and robust fiducial detection,” IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Oct. 2016, pp. 4193–4198.

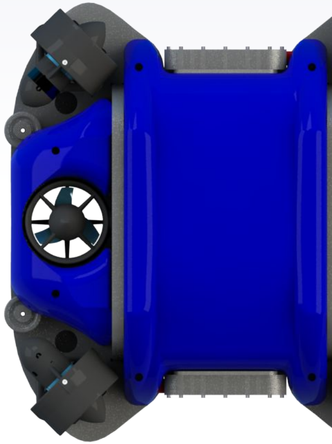
[3] Boston Dynamics Spot https://www.bostondynamics.com/sites/default/files/inline-files/spot-enterprise.pdf

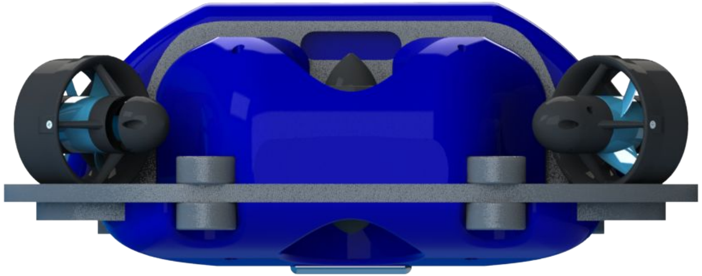
[4] Perseverance Mars Rover, https://www.reddit.com/r/opencv/comments/lq0ffb/news_apriltag_16h5_27_and_28_on_perseverance_rover/

Bonus Slides to aid QA

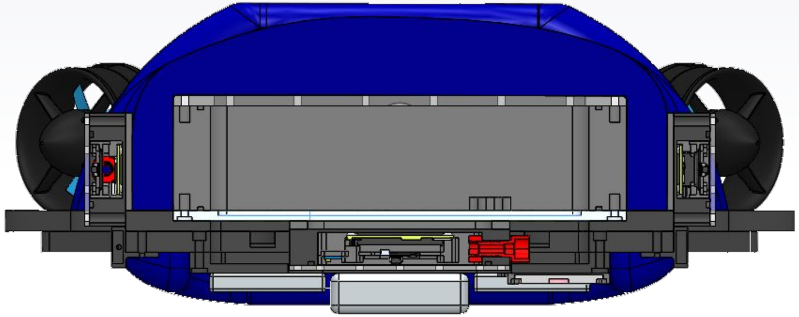
Mechanical Images







Mech Images



DIY Anodizing

Anodized thruster motor controllers at home using common off-the-shelf products (pool ph- and lye)

Great results with a simple process!

Finished Anodized Result

Motor controller pucks in acid solution

Buoyancy

Buoyancy Blocks

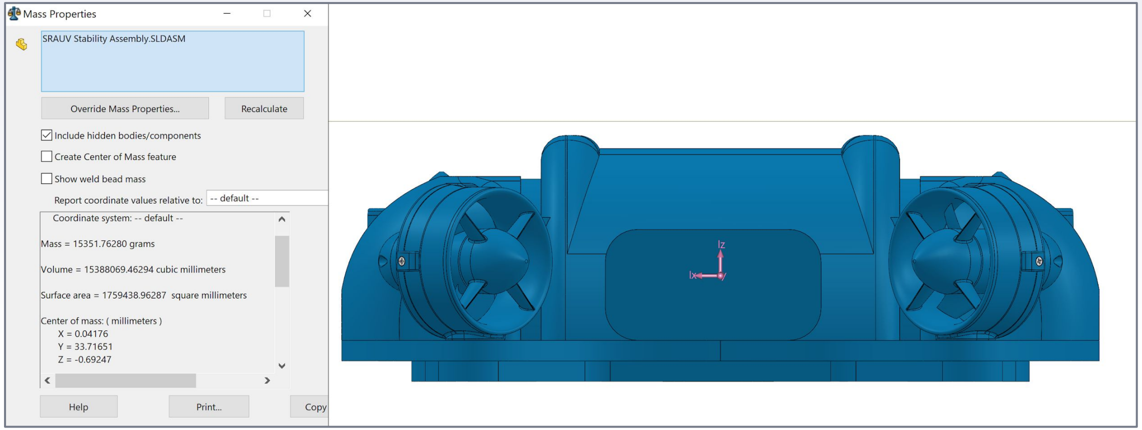
Buoyancy blocks are essential in order to create 'neutral-buoyancy' in water, making the AUV essentially weightless.

These blocks were CNC machined in layers from common rigid insulation foam and then painted / coated for water resistance.

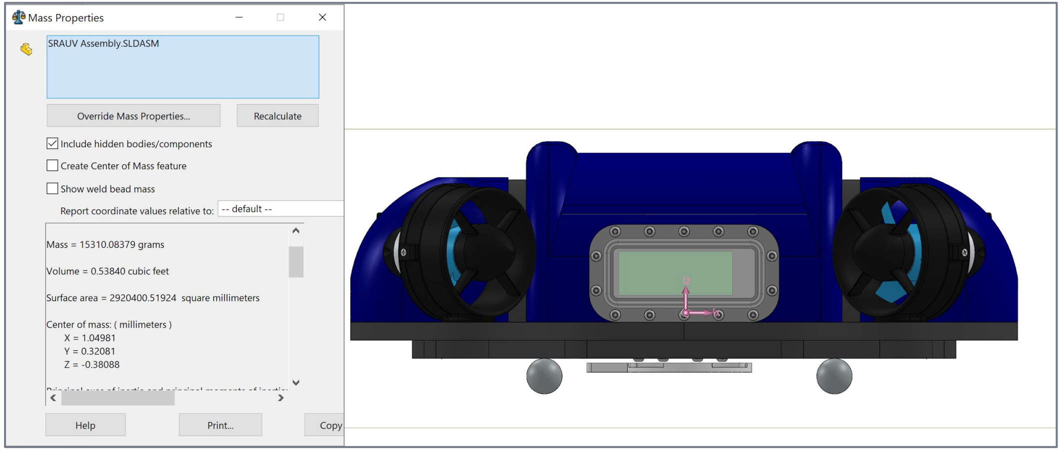
Provides over **7 kg** of positive buoyancy in water.

Total cost: $80
Total Weight: 0.95kg

Stability Analysis



Center of Volume Calculation



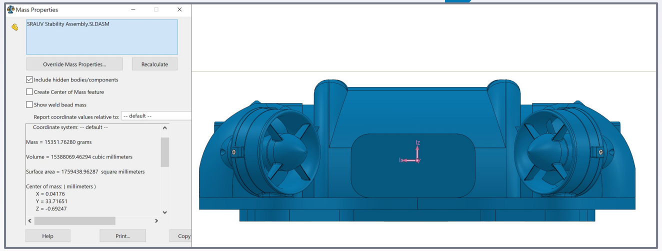
Center of Mass Calculation

Using SolidWorks, the volumetric center and center of gravity were found.

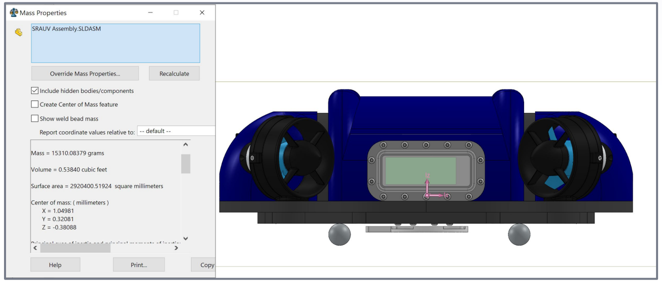
The stability of the vehicle is directly proportional to the distance between these centers.

This *separation distance* was maximized by placing all ballast weights at the bottom of the vehicle and the majority of the buoyancy at the top.

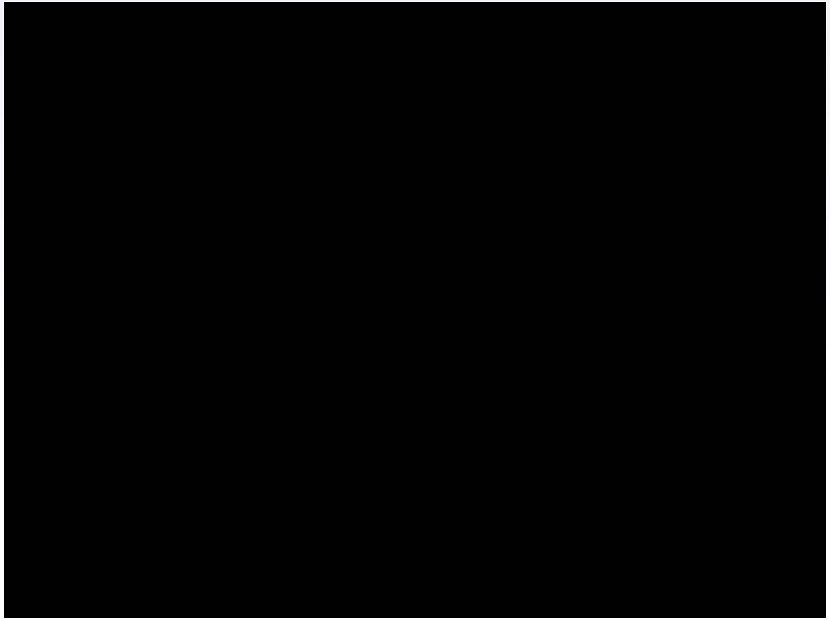
Stability Analysis



Center of Volume Calculation



Center of Mass Calculation



Using SolidWorks, the volumetric center and center of gravity were found.

The stability of the vehicle is directly proportional to the distance between these centers.

This *separation distance* was maximized by placing all ballast weights at the bottom of the vehicle and the majority of the buoyancy at the top.

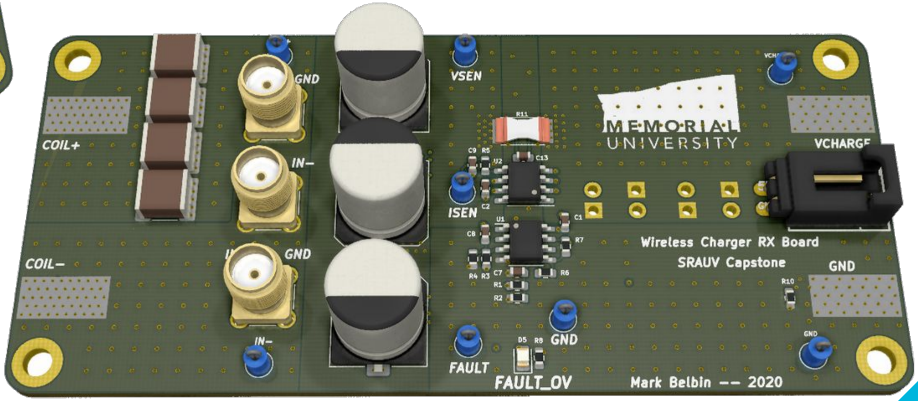
Dock Charging

How to charge battery system underwater?

1. Mechanical Wet-mate Connector
2. OTS Wireless Power
3. Custom Wireless Power

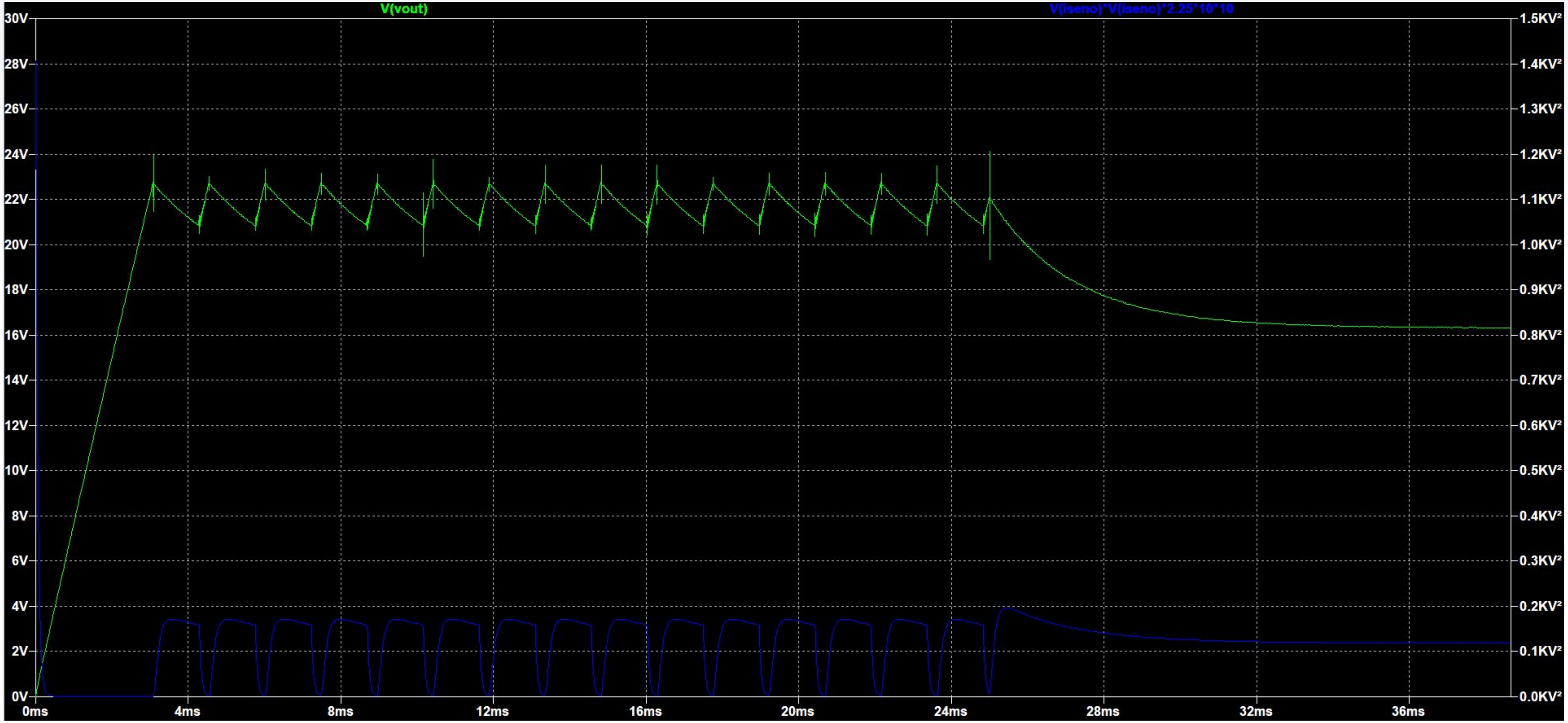
Wireless Charging Specifications

Input Voltage	24V
Max Current	3A
Output Voltage	18V
Max Power	~50W
Operating Freq.	300-100 kHz
Charge Time	~ 2.5 Hours



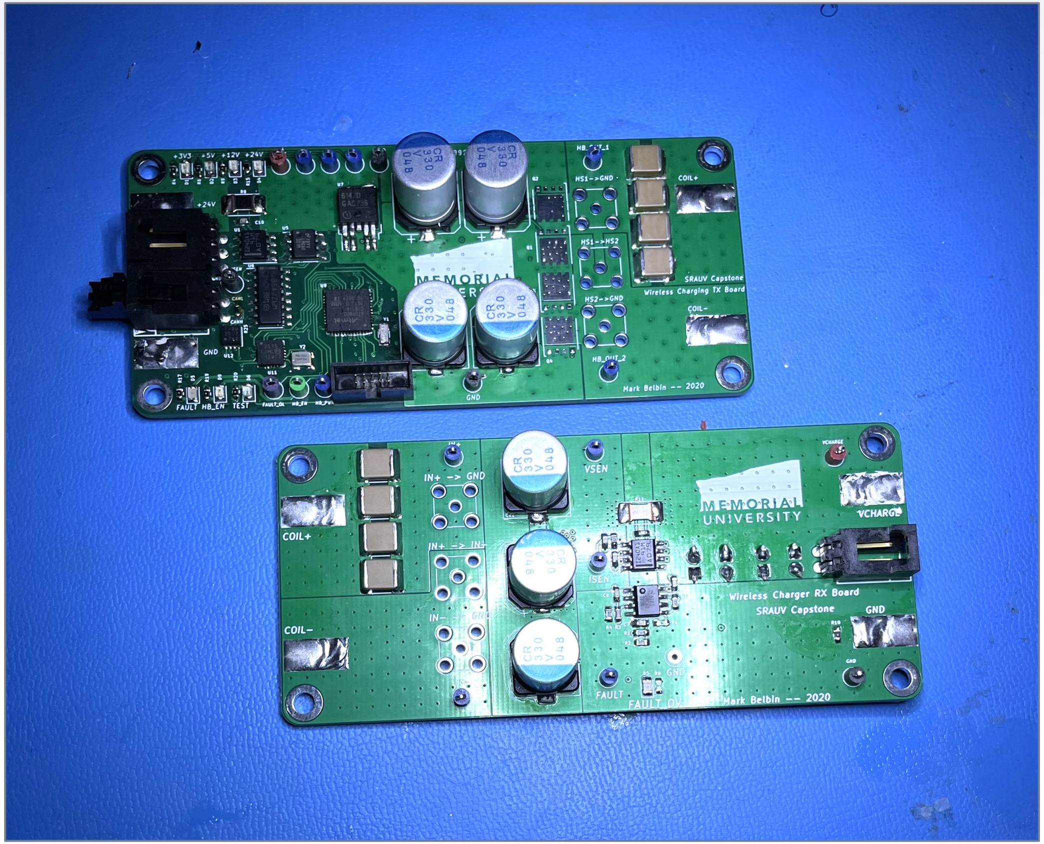
Above: Wireless Power Transmitter
Left: OTS Wireless Power Coils
Right: Wireless Power Receiver

Wireless Power Simulation



LTspice Wireless Power Simulation Results

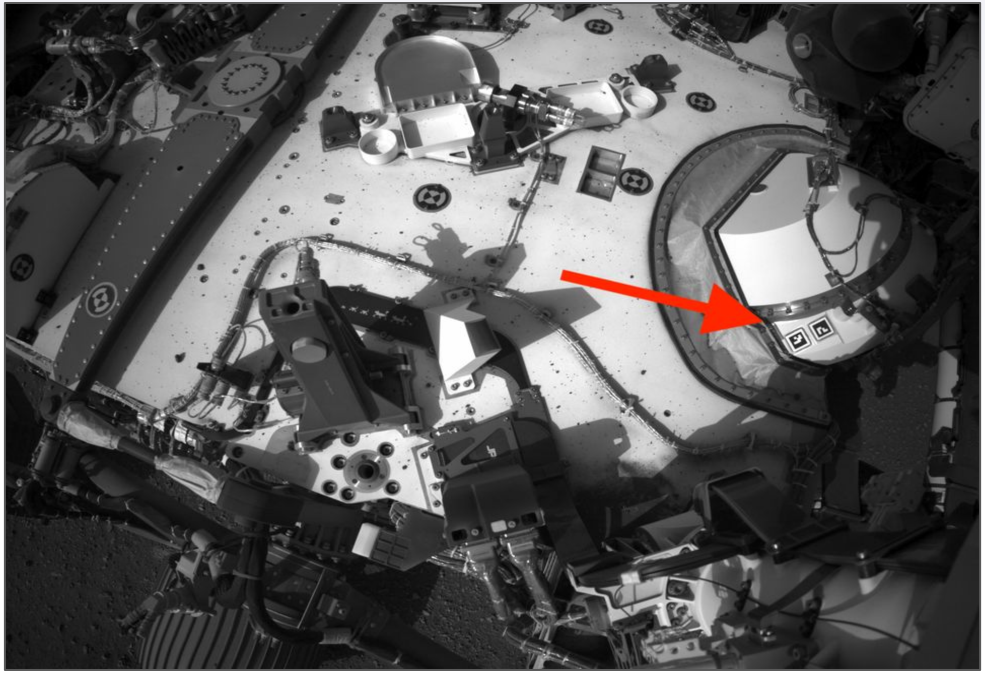
Wireless Power Boards



Wireless Power Assembled Boards

April Tags in the Wild

April Tags used to dock the new Boston Dynamics Spot Robot Dog. [3]

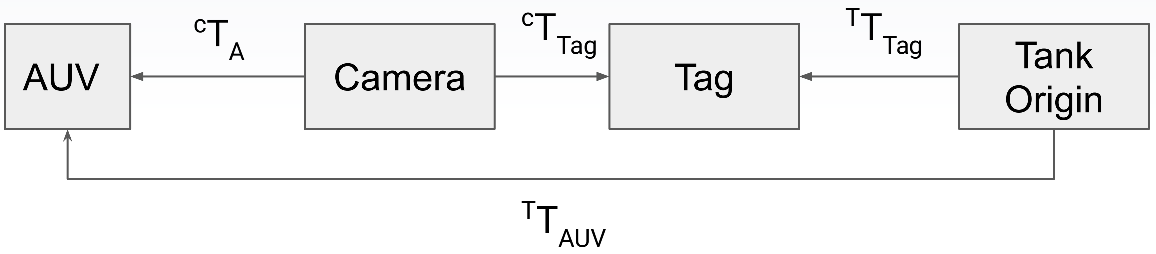


April Tags used on the latest Mars Perseverance Rover, using the same 16h5 tag family. [4]

AUV Localization

The AprilTag algorithm gives us the camera pose relative to the tag.

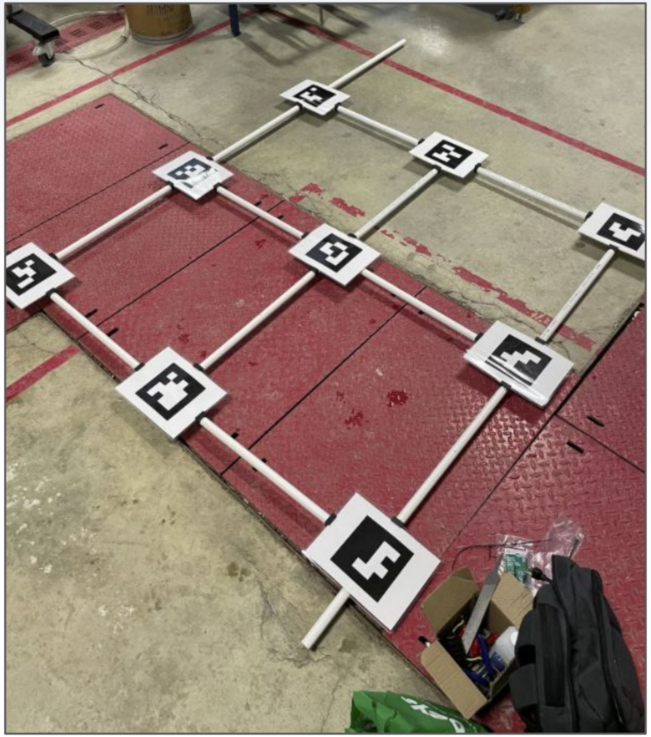
Using coordinate transforms, we can then find the AUVs global position relative to the tank!



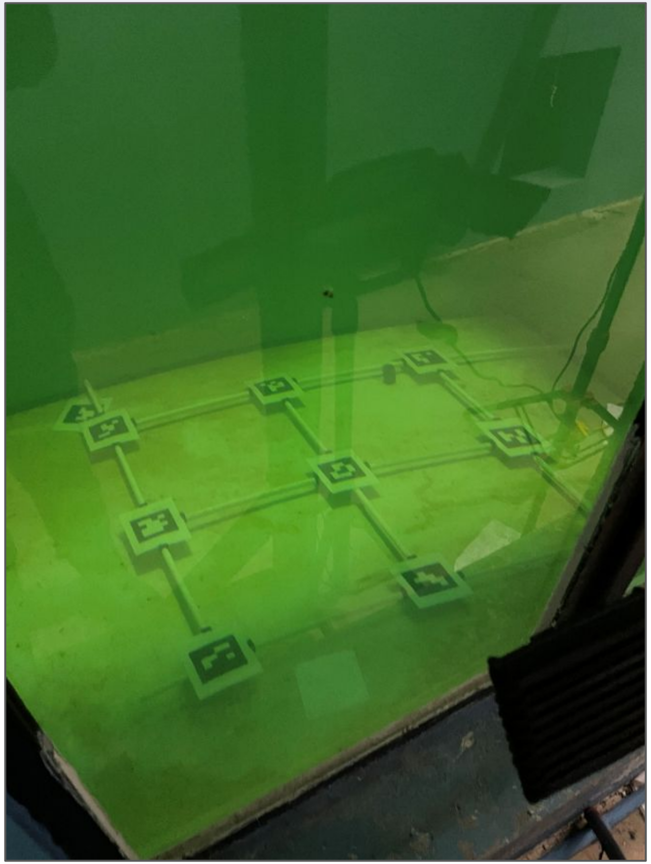
$^{c}T_{Tag}$ Is given by the april tags library

$$\boxed{^{T}T_{AUV} = {^{T}T_{Tag}} * inv(\,^{c}T_{Tag}) * {^{c}T_{A}}}$$

April Tag Grid

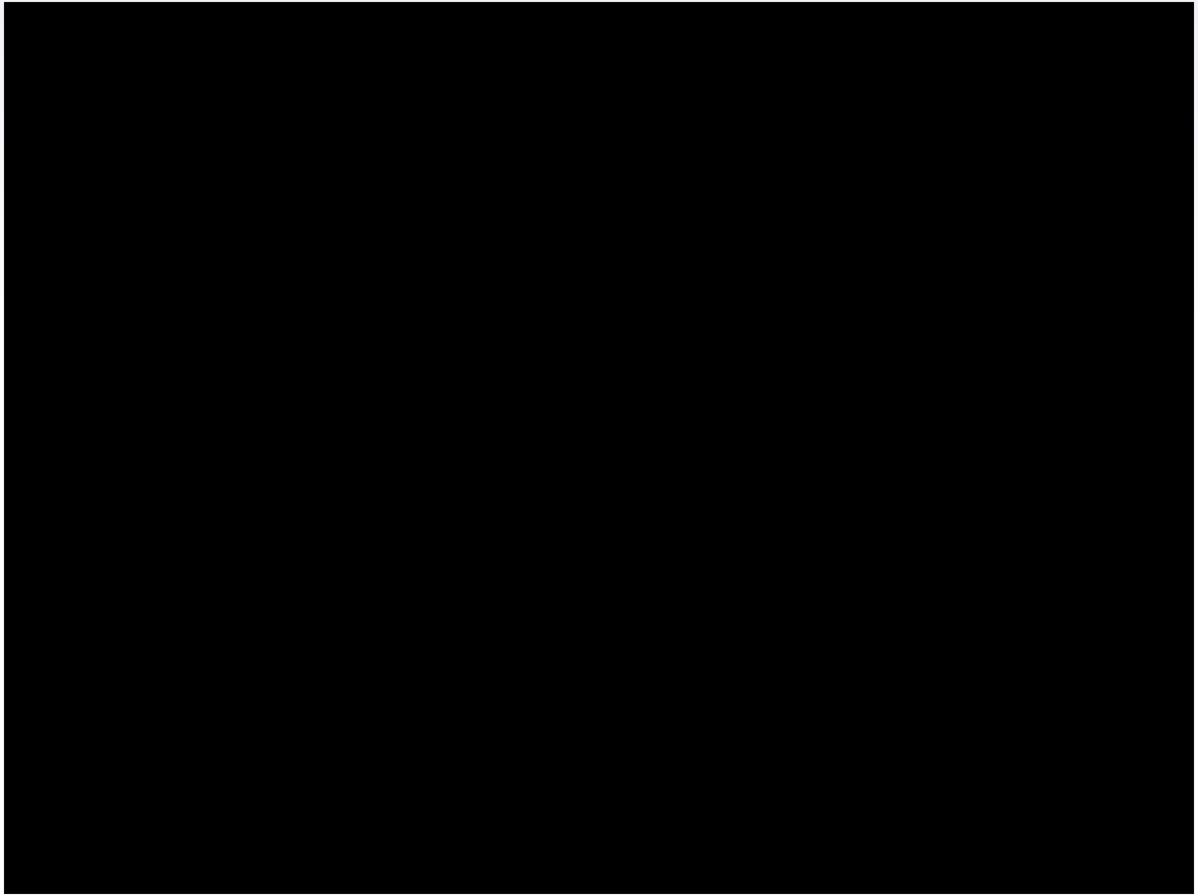


Constructed AprilTag grid on the surface. It measures about 1.75x1.75m.

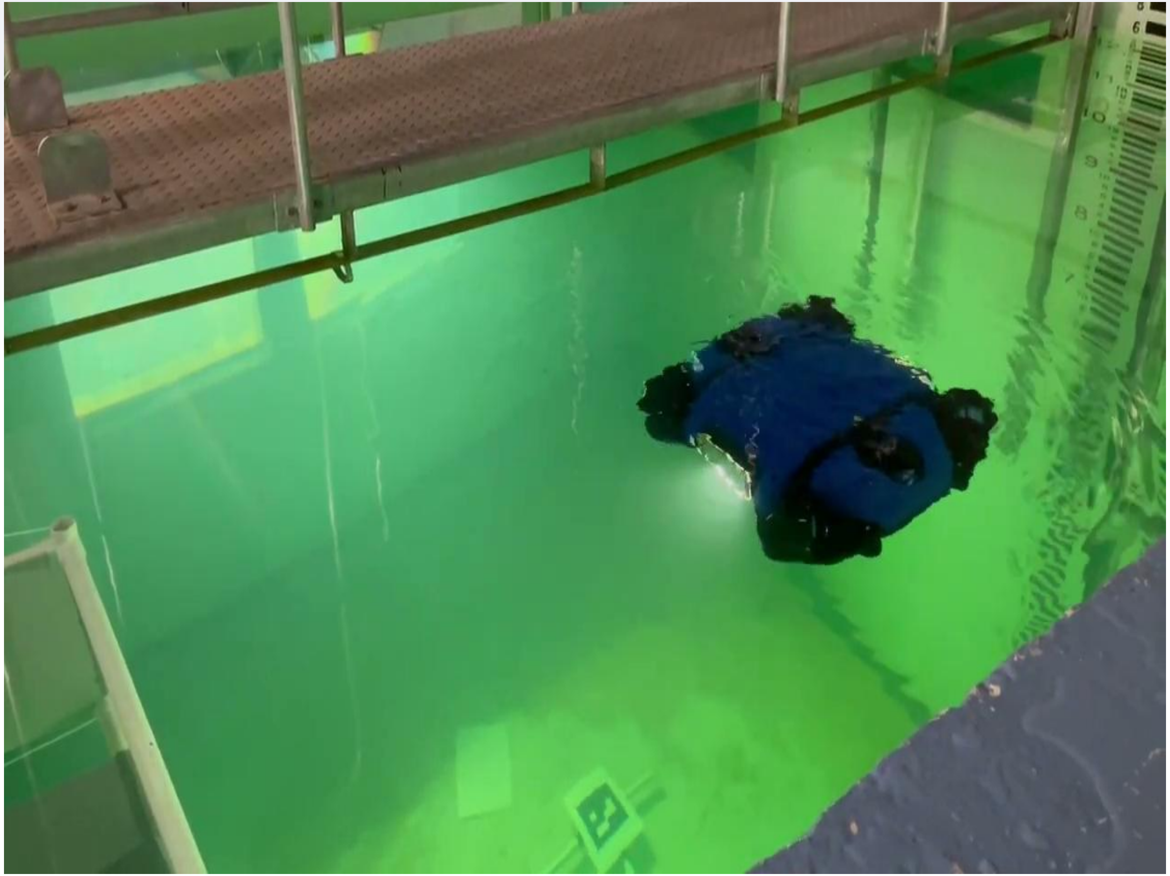


Constructed AprilTag grid at the bottom of the Deep Tank.

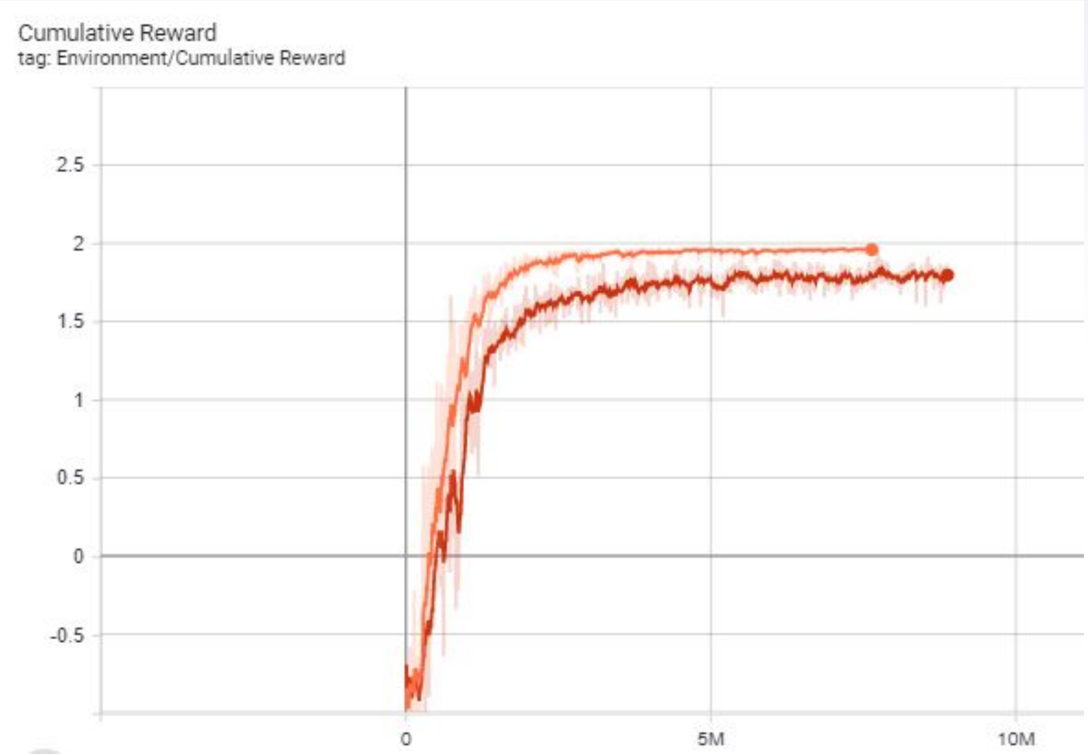
Real World Integration



Real World Docking



Obstacle Difficulty



Orange = DVL + Distance Sensors
Red = DVL + Distance Sensors, with Obstacles in env

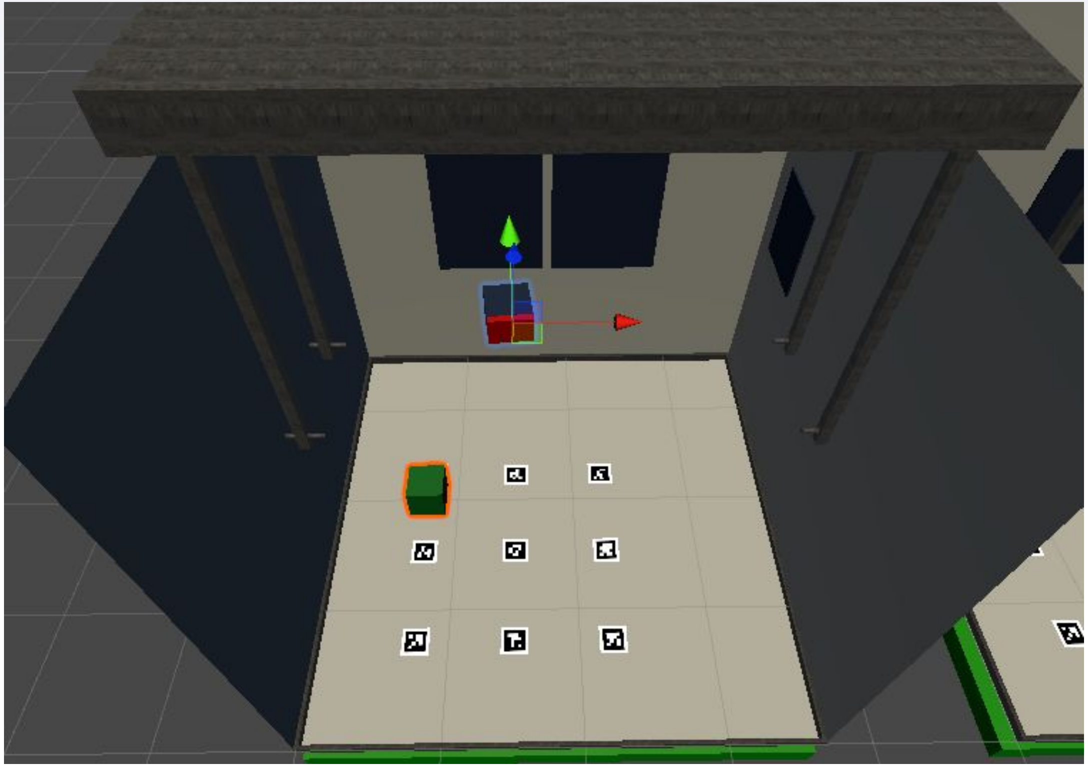
Curriculum Learning

$$X-1=9$$

$$2X-2=18$$

$$2X-2=38-2X$$

$$X^2-50=50$$

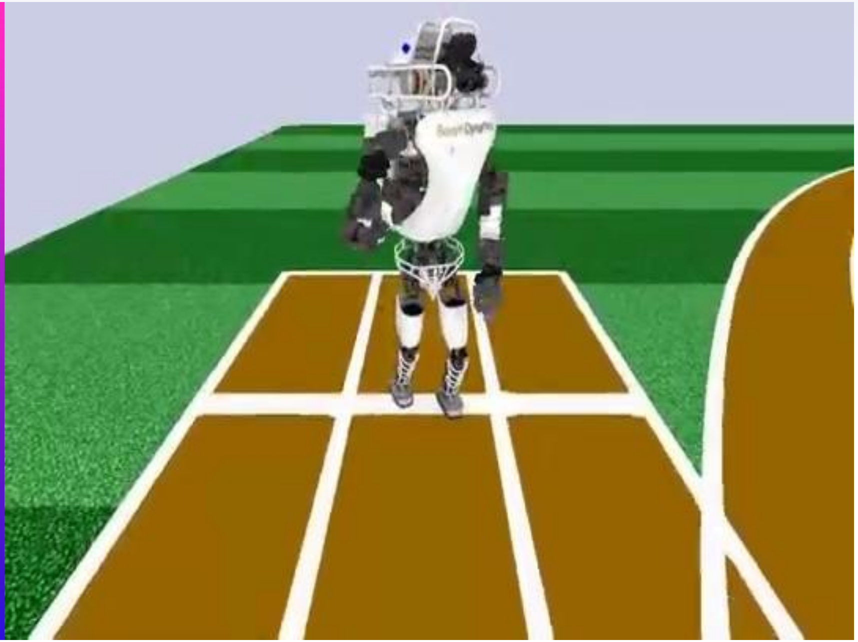


RL Algorithm of Choice

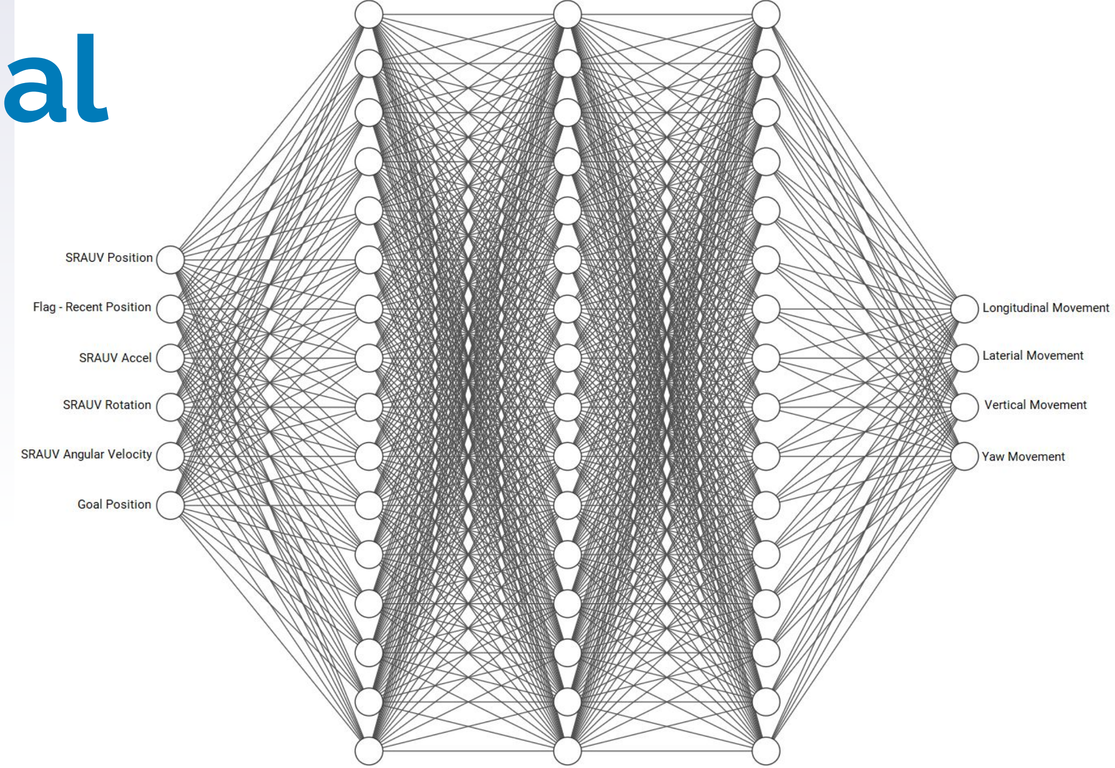
Proximal Policy Optimization

We're releasing a new class of reinforcement learning algorithms, Proximal Policy Optimization (PPO), which perform comparably or better than state-of-the-art approaches while being much simpler to implement and tune. PPO has become the default reinforcement learning algorithm at OpenAI because of its ease of use and good performance.

July 20, 2017
3 minute read



Neural Nets



SRAUV Position
Flag - Recent Position
SRAUV Accel
SRAUV Rotation
SRAUV Angular Velocity
Goal Position
Longitudinal Movement
Laterial Movement
Vertical Movement
Yaw Movement