Summer 2018 Report

MIT PavLab

Isaac Perper

Summer Goals:
- Learn C++ and MOOS-Ivp
- Become familiar with REx USV and Harbor Challenge
- Develop battery monitor MOOS app for REx computer battery
- Improve vehicle camera hardware, and develop computer vision program to identify stoplights at Boston Harbor locks.
- Perform various day-to-day robotics work

Most of the below information can also be found in our Sea Grant Final Presentation

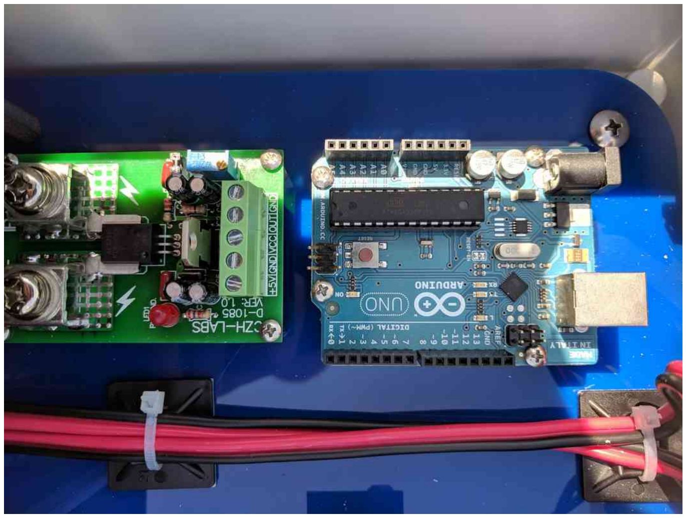
Summer Achievements and Timeline:

June – July: Learning MOOS and C++

In the first month of the summer I worked through the 2.680 course labs, which taught the basics of C++ and the MOOS autonomy platform. I completed the 10 C++ labs and 2.680 labs 1-10, 15, and 16, out of the 18 total. This got me familiar with MOOS vehicle setup, Pablo boxes, and the general process of making MOOS apps.

Late June: REx Battery Monitor

Once I had a general understanding of MOOS, I began developing an application to monitor the status of the battery powering the computers on REx. Prior to this, there was no real-time way of knowing the state of the battery during autonomous missions, which was dangerous since the computer battery powers all the autonomous systems.



Hardware Design:

- Arduino Uno connected to current sensor and voltage sensor
- Current sensor in series with power distribution board, voltage sensor in parallel
- Arduino takes readings multiple times a second.
- Current integrated over time to give total energy used
- Given initial battery charge in mAh, battery state can be accurately determined.

Software Design:

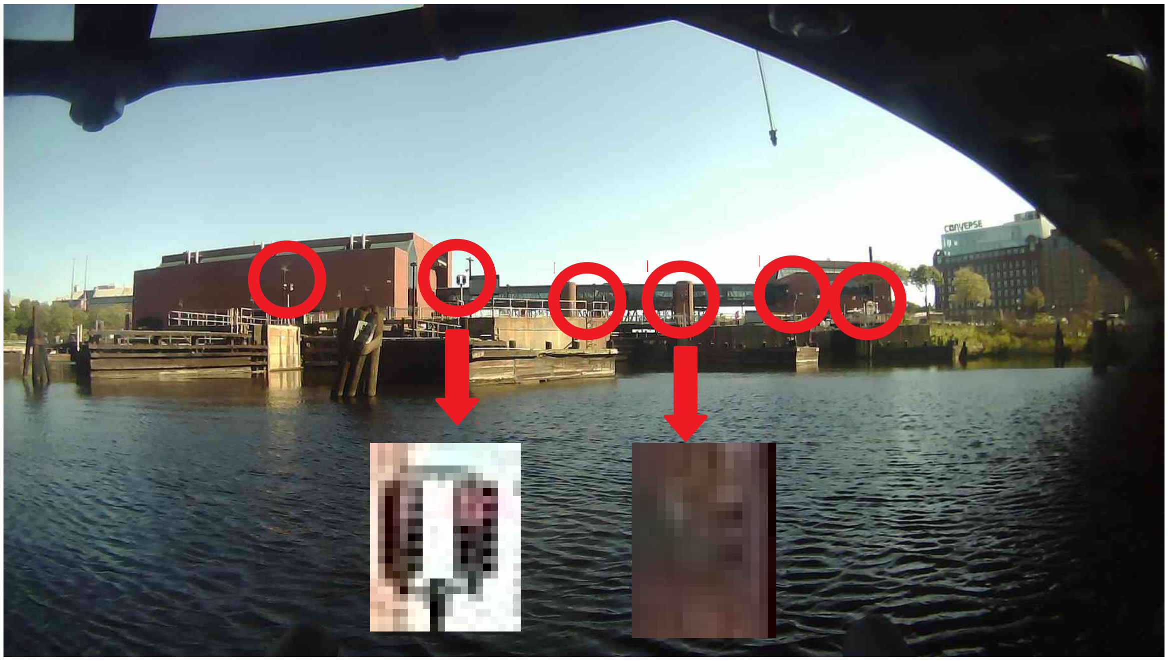
- Arduino sends string of sensor readings over serial port to front-seat computers
- MOOS app parses the string and updates MOOS environment variables
- Battery voltage, current, and charge state posted to pMarineViewer so vehicle operators are aware of battery status.

Outstanding Issues:

- Noise in Arduino readings causes large errors in current readings at low current draws
    - More accurate voltage measurement (read: Not an Arduino) or stable reference power supply needed to improve system accuracy.

Mid-July – Present: REx Vehicle Stoplight Detection

Problem: Current camera setup not adequate for computer vision techniques.



It’s clearly impossible to identify all six of the stoplights.

Even the human eye has a hard time identifying the 6 stoplights in the images, which meant first improving the camera hardware.

Camera Hardware Improvements:

- GoPro Lens covering the camera to remove light diffusion from translucent cover plastic
- Lasercut camera mounts to eliminate shaking and distortion (previously zip-tied).
- Larger camera lens to improve resolution and dynamic range of smaller objects at a distance.

Once the hardware was adequate, it was possible to look into the best computer vision techniques for identifying the stoplights. Note: There was no data with the new hardware to test this with yet.

Trying template matching didn't work well due to the changing conditions and perspective on different harbor missions and times of day.

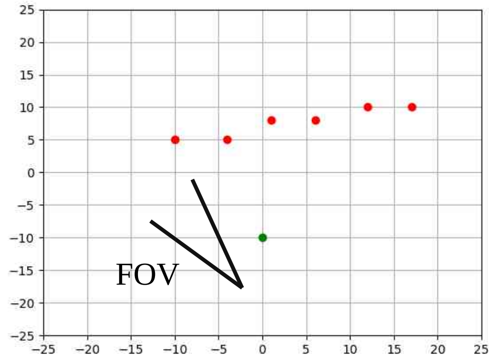
Next Test: Using background buildings as reference to light locations.

Program mislabels the locations of the stoplights based on position of large building in background.

Obviously, using the buildings as a reference didn't work either, as the perspective shift was too dramatic between locations to accurately identify where the stoplights were.

Current Method: Localizing stoplight locations on camera frame based on GPS coordinates.

- Take local coordinates of each stoplight and REx vehicle.
- Project stoplight location onto image plane.
- Use zoomed lens to focus onto each region and use simple color thresholding to recognize stop light signal.



Based on vehicle heading and location, the camera see's some FOV

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

System Projects Real-World Coordinates onto Image Frame Coordinates

The main downside is that all the objects must be pre-mapped, but it makes up for it in reliability and consistency. No computationally intensive tasks are necessary to determine the state of the stoplights.

Work to be Done:

- Develop image thresholding based on location in image frame
- Conduct real world tests.
- Expand program to be useful for other objects.

Summary of Summer

I enjoyed the variety of tasks and opportunities I had during the summer. Working at the sailing pavilion and getting on the water was a great way to spend the summer while learning a lot about programming and robotics. I'm a lot more confident in my ability to develop software to solve problems, and I also have a much better understanding of the work I enjoy doing in robotics.

In retrospect, I should have spent less time on the 2.680 labs and more time on the vision software, which was one of my primary goals for the UROP this summer. However, I am still satisfied with my progress on the stoplight detection and I look forward to continuing to work on it to some extent throughout the semester. Once more data is acquired, I would be interested to test more machine learning based object detection for obstacle avoidance and giving ability for REx to operate on less planned routes.

Additionally, I would like to test more of what I've built on actual missions, which was difficult during the summer due to timing and other events going on in the Pavlab. Prototyping systems is just one step in process, and creating a final product of what I worked on this summer would ensure a noticeable and useful impact on the REx vehicle.