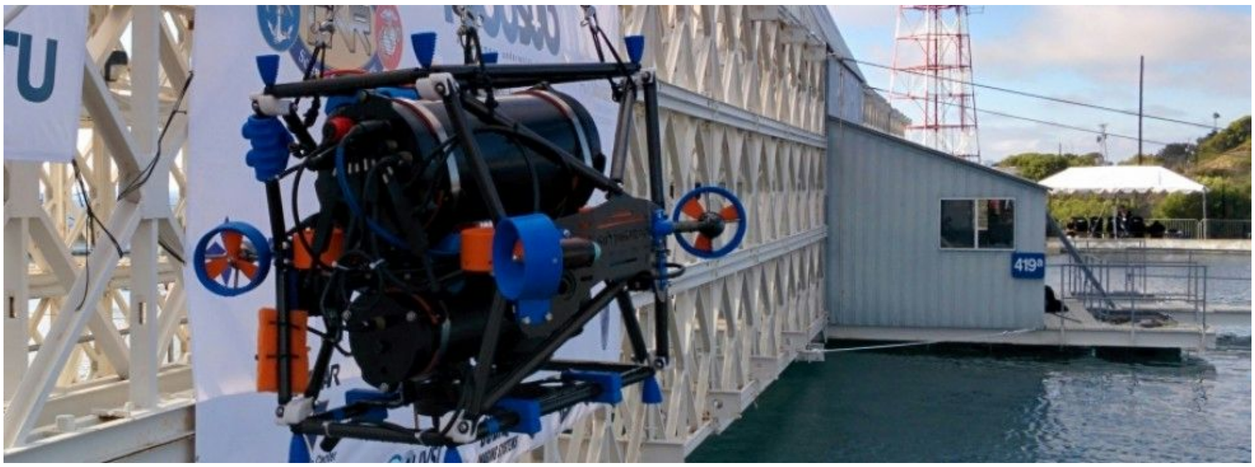
IMDL - EEL 5667

Robot Proposal

Jacob Panikulam



Instructor: Dr. Arroyo

TA: Ralph Leyva

Lab: Monday

Robot: Subjugator

UFID: 8651-5199

Degree: Electrical Engineering

Year: 4th

Abstract

This project is an effort to substantially advance to complexity of the autonomy that the Subjugator robot is capable of. This includes *true* environmental awareness (Instead of simple color thresholding), sparse map generation (As opposed to dead reckoning), real motion planning (Not just straight-line paths) and an effective fault detection/isolation system.

Introduction

Subjugator 8 is an autonomous submarine built by our group in the UF Machine Intelligence Lab. It has 8 thrusters, (Two for redundancy), and a host of sensors outlined in the Sensor Manifest section of this document. In the past, Subjugator's autonomy was largely reactive, the vessel did little planning (almost none to speak of), very little environmental comprehension or mapping, and no contingency response beyond shutting down entirely when thrusters undervolted. My project is to develop a full autonomy suite for Sub8 - this includes obstacle comprehension and avoidance, motion planning, fault isolation, and visualization of vehicle state for human debug use.

The intention here is to develop tools that can be reused on the Navigator robot, for the RobotX competition in Hawaii this coming December. The sub is being used as a development platform because it takes substantially less effort to take the sub out for testing. This paper will discuss what the complete software system will look like, what problems will be solved, and how I intend to solve them.

Robot Goals

The core special sensor will be the single down-camera, as I am most interested in inference from incomplete information (ex: loss of true scale from monocular projection). By the end of the semester, I intend to be able to:

1. Generate a sparse map of the environment, and 3D obstacle association with this sparse map

2. Execute trajectories that avoid collision using a library-based receding horizon controller (Dependent on 1.)
3. Simulate everything before putting the vehicle in water
4. Visualize using Rviz or raw OpenGL the sub's understanding of the world, and its motion plans, so that a human can verify its validity

Mobile Platform

- **Power**: 2 - 4 (4 not yet supported 5400 mAH Lithium Polymer batteries, lending ~1 hr of battery life)
- **Compute:** Mini-ITX PC running Ubuntu 14.04, Intel i7 processor, 8 gb RAM, 1TB Solid-state drive
- **Compute:** (Future plans include a Jetson TK1 for GPU computing in addition to the main computer)
- **Actuation:** 8 VideoRay M5 thrusters, communicated to serially via RS485
- **Navigation:** Teledyne RDI DVL, STM IMU/Gyroscope, magnetometer
- Max depth ~140 ft
- Max speed ~1.5 m/s (Max)

Algorithms

- SLAM will be done using a mixture of Large-Scale Direct [1] monocular SLAM and Semidirect Visual Odometry [2] using vehicle odometry for scale estimation
- Obstacle detection will be done using the marching cubes algorithm on the sparse map generated by SLAM to produce a "polygon wall" that expresses non-traversible region in 3D without requiring a very dense traversability map. This is because the map is inherently sparse, so without analysis there is no inherent notion of "inside" or "outside" a mess of points.
- Motion planning will be done using an RHC library method [3], where direct collocation will generate some huge number of trajectories for a discretized set of initial conditions. Due to the symmetry of the sub, this only needs to be done in one quadrant of the plane, for the parameter **`initial_velocity`**, and **`initial_heading`**
- Visualization will likely be done using RVIZ, or Vispy, a Python interface to OpenGL

- Ideally, simulation will be done using the same tool developed for visualization, and will be an extension to the existing sub simulator

Sensor Manifest

There are other sensors on the vehicle, these are the only ones that I intend to use for my project.

- DVL - Doppler Velocity Log; A device for estimating velocity relative to the bottom of the water environment using the Doppler effect. Accurate enough that double-integration of this signal (simple dead reckoning) will allow the sub to surface within a ~5 meters of its target after 20+ minutes of travel
- Forward Stereo Cameras - Two cameras mounted at the front of the sub produce a disparity map via block-matching, which is converted to a depth estimate by triangulation. This can be interpreted as a colored point-cloud
- Down-Camera - One camera is pointed downward from the sub chassis, and as a result it can track the bottom
- Depth Sensor - Highly accurate water-pressure depth sensor

Conclusion

By leveraging the algorithms mentioned previously, the sensors at our disposal, and the collective ambition of the Subjugator team, this project has an acceptably high likelihood of success by the end of the semester.

Bibliography

[1] Jakob Engel, Thomas Schops, "*LSD-SLAM: Large Scale Direct Monocular SLAM*", Online

[Available] http://vision.in.tum.de/research/vslam/lsdslam

[2] Christian Forester, M. Pizzoli, "*SVO: Fast semi-direct Monocular Visual Odometry",* Online

[Available] http://rpg.ifi.uzh.ch/docs/ICRA14_Forster.pdf

[3] Drew Bagnell, Shreyansh Daftry, "*Semi-dense Visual Odometry for Navigation in Cluttered Environments*", Online

[Available] http://www.ri.cmu.edu/pub_files/2015/5/root-1.pdf