ROBOSUB

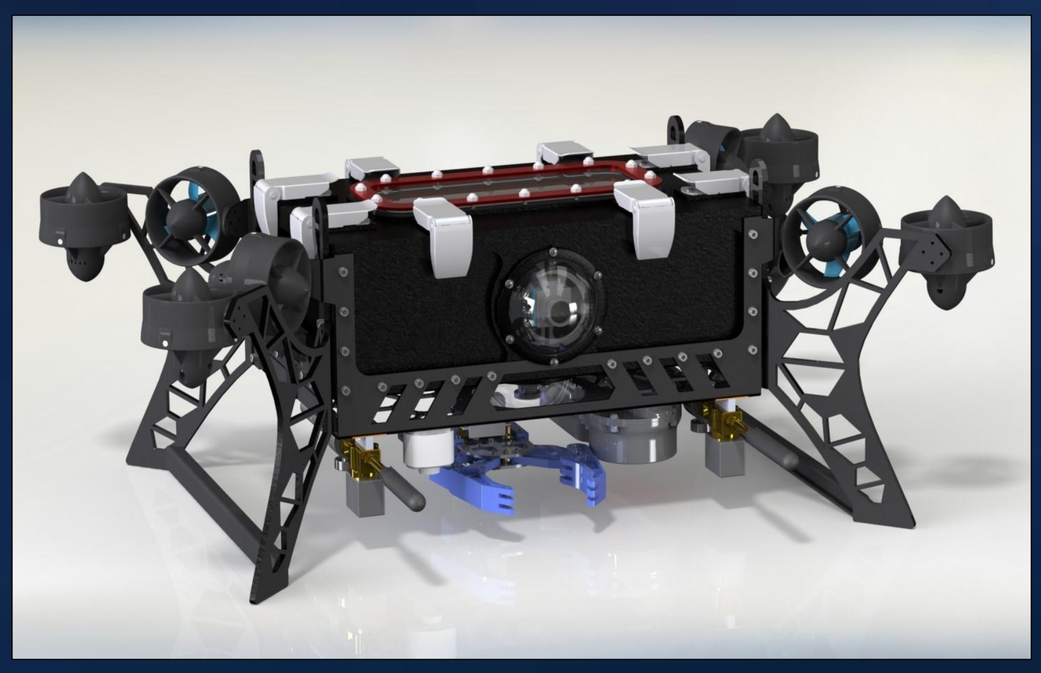
Team Members: Thomas Benson, David Camacho, Bailey Canham, Brandon Cao, Roberto Hernandez, Andrew Heusser, Hector Mora-Silva, Bart Rando, Victor Solis, Milca Ucelo-Paiz

Faculty Advisor: Professor Richard Cross

Department of Computer Science
College of Engineering, Computer Science, and Technology
California State University, Los Angeles

Background

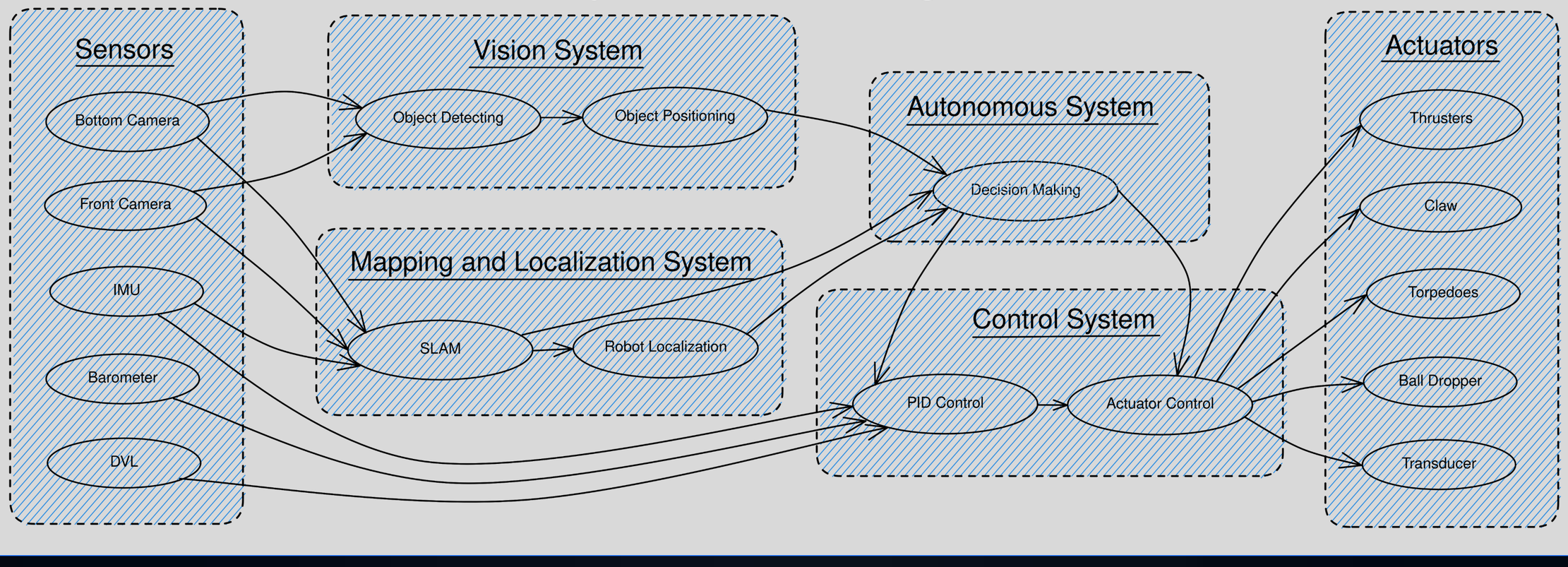
The annual RoboSub competition, hosted by the Office of Naval Research, requires students to design and build an autonomous underwater vehicle (AUV) that can complete specific tasks. This year, Cal State LA's RoboSub senior design team will be competing with the AUV named Lanturn.



Objective

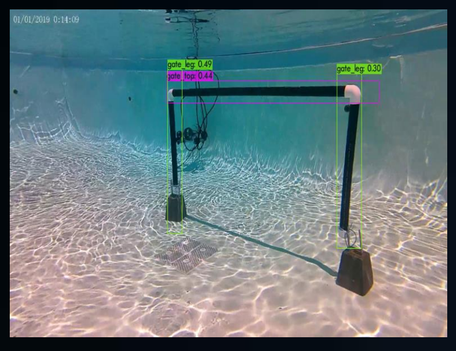
The RoboSub Computer Science team will design and develop the software system for Lanturn. The objective is to enable Lanturn to navigate the competition environment, identify competition tasks and complete them autonomously.

System Overview Diagram



Control System

The Control System gathers data from sensors, processes high-level commands and sends low-level commands to actuators for precise task execution. This is accomplished by writing drivers for the sensors and actuators, and developing a Proportional Integral Derivative (PID) Control System for stable navigation.

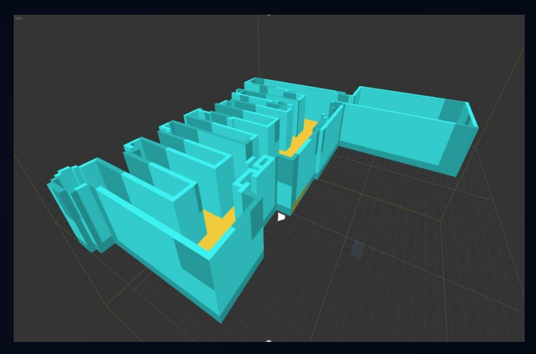


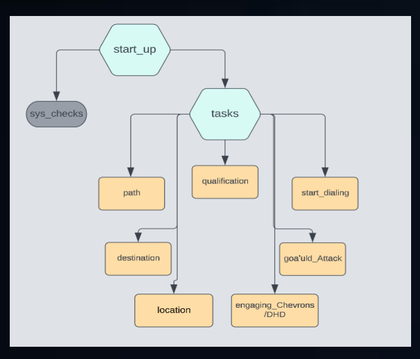
Vision System

The Vision System acquires and processes images, detects objects using machine learning algorithms and deep neural networks with the Darknet library. It processes the data from object detection and uses triangle similarity algorithms to find the relative positions of objects.

Mapping and Localization System

The Mapping and Localization System receives sensor data, uses Visual-Inertial Simultaneous Localization and Mapping (SLAM) algorithms from ORB-SLAM3 library to create a map of the environment and localize Lanturn in the map. It then uses the robot_localization ROS library to integrate additional sensors and improve localization.





Autonomous System

The Autonomous System collects and analyzes data from all other software systems. It uses the information to make decisions to achieve the desired objective. The system employs behavior trees to organize decision-making processes and incorporates fallback and recovery nodes to ensure continued functionality in the event of hardware malfunctions.