ECE 17
Autonomous Quad-Thruster Submersible Object Location Retrieval

Jose Coreas, Stephen Jardim, Elizabeth Moonjelly, Harold Park

Special Thanks: Nolan Ballew, Romel Gomez, Jay Rana

A. James Clark School of Engineering | Department of Electrical & Computer Engineering

PROBLEM DEFINITION

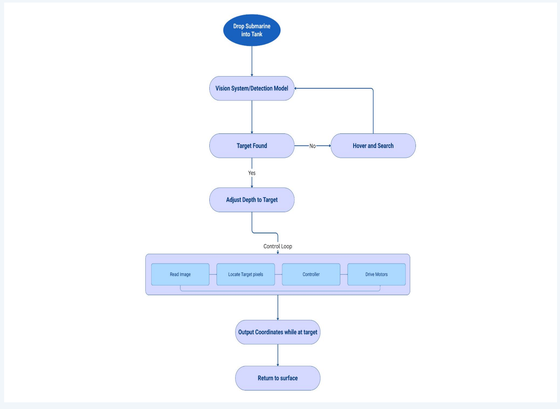
Design and develop an autonomous underwater vehicle capable of detecting, navigating to, and reporting the location of a target using vision-based control within a constrained environment. Challenges:

- **Diminished visibility**
- **Risk of water ingress**
  - Water pressure increases with depth, exacerbating this challenge
- **Difficulty of data transmission**
  - Radio and other wireless methods of communication are often ineffective underwater

SYSTEM DESIGN

We aim to develop and construct an autonomous submarine which can…

1. ***Identify*** an object that is underwater
2. ***Navigate*** towards said object
3. ***Transmit*** the location of the object



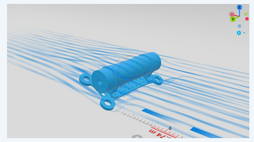
Two-processor architecture:

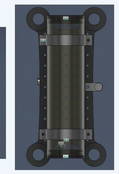
1. RPi 4: Vision, Control, SSH Transmission
2. Arduino Uno: Motor Driving

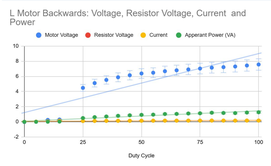
DESIGN CALCULATIONS AND ANALYSIS

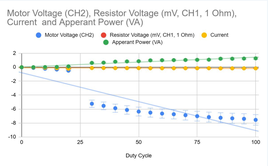
Displaced Mass

- The tube has a height of h = 12 inches and an outer diameter of d = 3½ inches, giving an internal volume of V = hπ(0.5d)^2 = 115.5 in^3 (2.54 cm/in)^3 = 1892 cm^3.
- At STP, 1 cm^3 of water has 1 gram of mass, meaning our tube displaces about 1.892 kg of water.









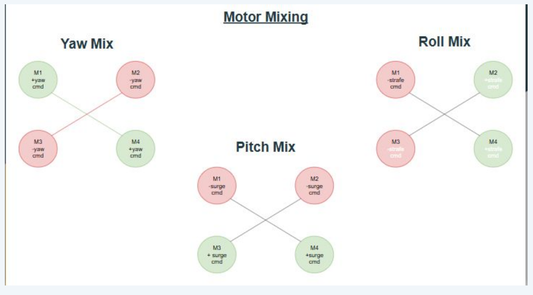
Power Consumption

- Battery capacity will be 2000 mAH.
- According to Pi Dramble, the Raspberry Pi 4 B consumed up to 1280 mA.
- From our motor characterization data, the motors typically consume up to 170 mA. For four motors this consumption becomes 680 mA.
- For 0.5 hours (30 minutes) of use, the battery will have 1020 mAH to power all other systems.

ALGORITHMS AND COMPUTER VISION

Navigation and PID Control

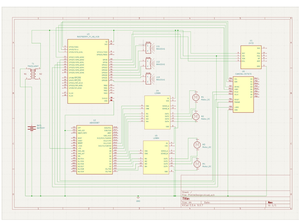
- Roll: driven by horizontal pixel error
- Pitch: driven by target apparent size to regulate approach speed
- Yaw: angular velocity held at zero to maintain forward orientation
- Surge bias: Nose-down pitch bias applied once target is centred within ±30px deadband
- Hover throttle: driven by vertical pixel error
- Motor mixing:
  - M1=hover−pitch + roll +yaw,
  - M2=hover−pitch - roll −yaw,
  - M3=hover+pitch -roll +yaw,
  - M4=hover+pitch + roll −yaw
- All motor commands clipped to 0–255 PWM range

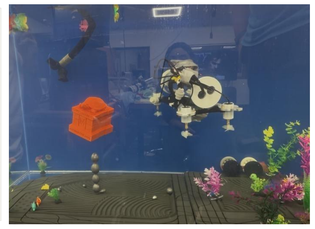


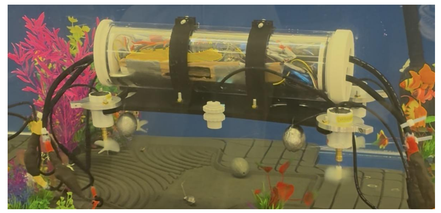
Computer Vision

- HSV color masking: H∈[0,5], S∈[120,255], V∈[20,200]
- Morphological open+close to remove noise (5×5 kernel)
- Largest contour above 350px² selected as target
- Centroid smoothed with α=0.7 exponential filter
- Pixel error (u_err, v_err) fed directly into PID controllers
- Frames captured at 360×360 resolution, target centred at (180, 180)

FINAL DESIGN







TEST RESULTS

Test 1: Demonstrated effective waterproofing
- No water ingress after submerged for 30 minutes

Test 2: Demonstrated constant depth using ultrasonic depth sensor
- Held a constant depth of various set points

Test 3: Demonstrated using Computer Vision to hold depth
- Held target at center of image as target moved

Test 4: Demonstrated navigation towards target