A **proportional–integral–derivative controller** (**PID controller** or **three-term controller**) is a control loop mechanism employing feedback that is widely used in industrial control systems and a variety of other applications requiring continuously modulated control. A PID controller continuously calculates an *error value* as the difference between a desired setpoint (SP) and a measured process variable (PV) and applies a correction based on proportional, integral, and derivative terms (denoted *P*, *I*, and *D* respectively), hence the name.

RoboCat: A Self-positioning platform for stationary measurements in the Aquatic environment

Lily Lamb[1], C. Papageorgiou[2], G. Voulgaris[2], and I. Rekleitis[1]
[1]Computer Science and Engineering
[2] School of the Earth Ocean and Environment

Research Question

Can we develop technology that facilitates measurements near the surface of the ocean?

Objectives

The objective of the project is to improve Eulerian Ocean measurements near the surface of the ocean (upper 2m) through the development of a cost-effective, self-adjusting, position-keeping platform. The design should improve data quality through minimization / elimination of interference by a nearby vessel or tether.

Project Overview

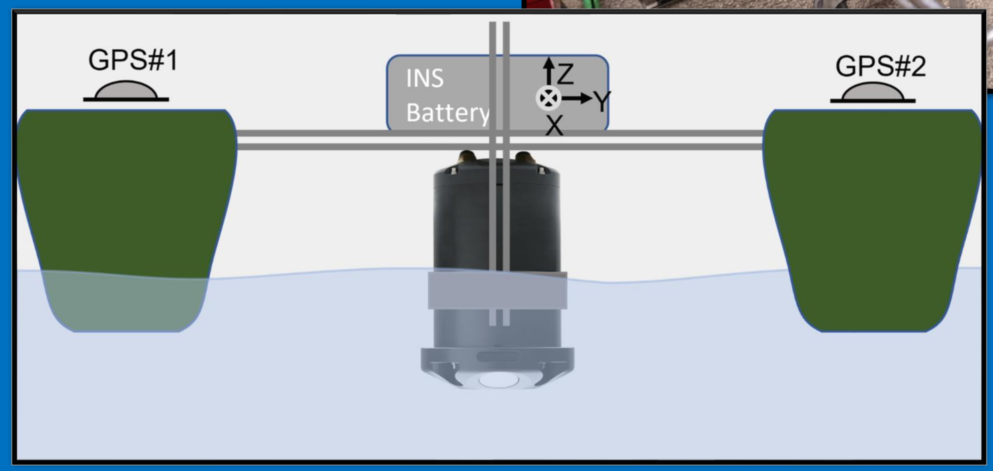
With the Magellan grant, we conceptualized and implemented a self-positioning platform for aquatic-based data collection. In the ocean, there are several areas of study (coastal plumes, for example) where the usage of a boat is suboptimal. Due to their size, boats interfere with fine structures in the water column leading to data collected not been representative of the real conditions; however, the platform we created (RoboCat) generates indigently less disturbance while having the ability to maintain position automatically. RoboCat is a catamaran platform fitted with four electric thrusters arranges in an Omni-X configuration and controlled by a Cube Orange® autopilot board utilizing the ArduPilot firmware. The default loitering option available in Ardupilot contained inefficiencies as described in the challenges section (see below). In order to resolve the issues and improve the ability of the RoboCat to maintain its position (loitering), I modified the source code of Ardupilot controlling the vehicle. These modifications can be reviewed in the solution section. Once bench testing and the modifications were complete, we transported the platform to a local lake and conducted a series of successful tests, followed with additional testing in the marine environment off the SC coast.

Challenges

In order to maintain a thruster turbulence-free environment in the middle of the platform, the 4 thrusters were arranged in an Omnix-X (i.e., 45 degrees off the main axis of the vehicle).

UNIVERSITY OF SOUTH CAROLINA
College of Engineering and Computing

Built platform

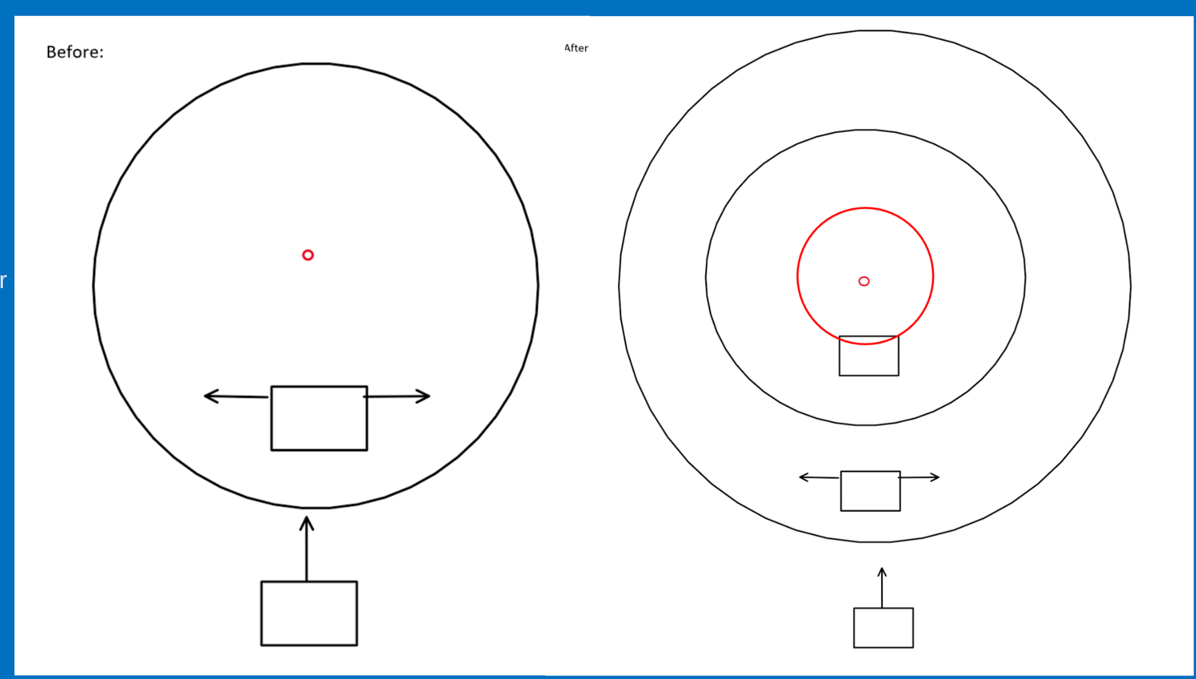


Initial Design

Testing code changes in USC Reflection Pond

Programing Loitering Mode (Before and After):

Original loitering code tasks:
1. Define a loitering point and a loiter circle of radius r.
2. Inside the loiter radius (r), the platform rotates to face the wind direction (and in turn the central loiter point – small red circle).
3. Outside the loiter radius, the platform moves forward until it is inside the loiter radius.
4. Then the platform returns to preform task 2.



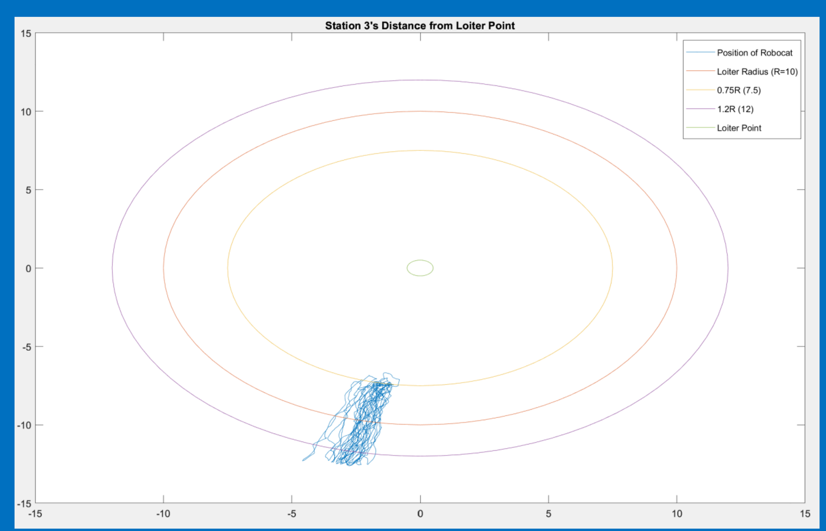
Modified loitering tasks
1. Define a loitering point and a loiter circle of radius r. Internally the code generates two circles (an inner with radius 0.75*r and an outer with radius 1.2*r).
2. When Inside the inner loiter circle the platform maintains the desired speed to counteract the forces (wind or current) that try to move it away (calculated by the software), while maintaining its heading (yaw) [but would not attempt to change yaw].
3. When outside the inner loiter circle but within the outer circle, the platform aligns itself to face the wind /current which in turn faces itself towards the central loiter point (small red circle).
4. When outside the outer loiter circle the platform moves forward toward the loiter point (until inside 75% of the loiter radius - red circle).
5. Once inside, the platform would continue to preform task 2.

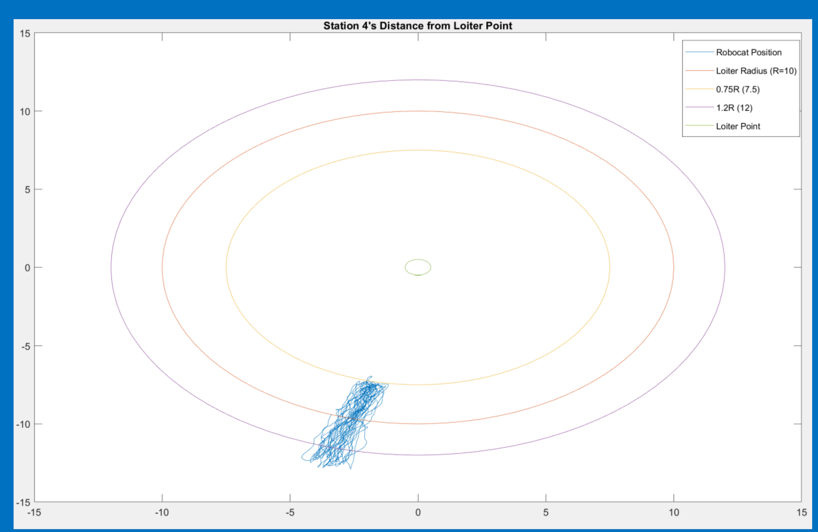
The Code in Action:

The figure to the right shows position data during testing in the marine environment. For this experiment, the loiter radius is 10 meters.

The three circles in the figure represent the different stages of loiter mode.

The smallest circle (0.75r or 7.5m) represents when the platform recognizes it has re-entered the loiter radius and maintains the desired speed (calculated by internal systems). We chose to maximize the time the platform spent within the loiter radius by creating a delay from when the platform re-enters the loiter radius and when the platform maintains the desired speed. Creating a smaller circle within the loiter radius allowed us to have this desired effect.





Furthermore, the central circle (r=10m) represents the user defined loiter radius. When inside the loiter radius (between 0.75r and r), the program maintains desired speed but once leaving, constantly rotates to face the current/wind direction.

Finally, the outside circle (1.2r or 12m) represents the location the platform recognizes it is away from the desired loitering location. It starts moving forward until it reaches the inner loiter circle. Separating when the platform rotated and when the robot moved forward simplified the processing and decreased the chance for error.

The field data (blue dots) show the platform leaving the outer circle, re-entering the inner circle and repeating this process during the loitering mode.

SCAN ME

Scan this to be brought to the Github with the platform's code

Scan this to be brought to video taken during first stage testing

SCAN ME

Challenges Cont.

This configuration, did not allow use of the default loiter option available in Ardupilot which was created for the standard sailboat design. The loiter option requires defining a loitering point and a circle of a given radius around the loiter point (loiter circle). The tests showed that the platform could not maintain a constant orientation, but it would be oscillating around the mean direction consistently. Upon investigation of the log files, it was revealed that the platform underwent one of two actions: turn to focus on the central loiter point (the center of the circular area the platform stayed within) or move forward until reentry into the loiter radius has been achieved.

Solution

(1) Using the default Loitering option. Initially, my efforts focused on calibrating the vehicle to achieve a more stable loitering behavior. This included trials in the water and fine tuning of the controller's automation by adjusting its PID (proportional–integral–derivative controller) values. These values optimize Ardupilot's control loop mechanism through a continuously modulated control. The improvements achieved through PID optimization were significant but not satisfactory as the platform continued to exhibit occasions of constant yaw movement when in loitering position.

(2) Software modification. Given the fact that the Ardupilot code is open source, allowed me to get insights in the code and investigate methods to eliminate the oscillating problem. **The code was modified,** and a second loitering radius was introduced The description of the code before and after the modifications can be seen to the left (noted in white). Essentially, the new code minimized the time when the platform centered on the loiter point and created a space within the loiter area where the platform did not move. Allowing time where the platform did not move decreased the disturbance and made the platform more stable.

Conclusions

A stable self-positioning platform for stationary measurements in the marine environment has been developed and tests in lake and the marine environment.

This platform would allow the installation of current meters and other ocean measuring sensors for the collection of data free of mooring lines or large boat interference.

UNIVERSITY OF SOUTH CAROLINA
College of Arts and Sciences
School of the **Earth, Ocean and Environment**