Nautilus Technical Design Report

Jansen-Sanchez
Department of Electrical Engineering
Embry-Riddle Aeronautical Univerity
Daytona Beach, United States
jansenss@my.erau.edu

Mascal
Department of Mechanical Engineering
Embry-Riddle Aeronautical Univerity
Daytona Beach, United States
mascalt@my.erau.edu

Coello
Department of Mechanical Engineering
Embry-Riddle Aeronautical Univerity
Daytona Beach, United States
coelloe1@my.erau.edu

Haeberlen
Department of Mechanical Engineering
Embry-Riddle Aeronautical Univerity
Daytona Beach, United States
haebera1@my.erau.edu

Thompson
Department of Aerospace Engineering
Embry-Riddle Aeronautical Univerity
Daytona Beach, United States
thompm73@my.erau.edu

Wilde
Department of Aerospace Engineering
Embry-Riddle Aeronautical Univerity
Daytona Beach, United States
wildeg1@my.erau.edu

***Abstract*—The documents the design goals for the RoboSub 2023 Competition. With the goals of reliable movement in six-degrees of freedom, and basic object localization while submerged, the team for Project Nautilus integrated a new suite of sensors include a WaterLinked A50 Doppler Velocity Logger, a VectorNav Inertial Measurement Unit, and a Blue Robotics Ping360 Sonar. This in combination with novel integration software, yielded a product that met the design window objectives.**

I. Competition Goals

A. Starting Place

The starting place of this design iterations process, the Autonomous Underwater Vehicle (AUV), had reliable a functional Mechancial Design. The Electrical system was servicable with minor deficiencies. The area where the AUV required the most engineer design work was in the sensor suitre and the control software, and the navigation algorithims. The concequence of the under developed systems in the RoboSub 2022 Competition in 2022 was imprecise movement control coupled inaccurate navigation commands. Our primary goals address these shortcomings.

B. Primary Goal: Have Reliable Movement Control

Our primary goal for the year is to develop a reliable and accurate movement control system. Before any other work and proceed, reliable AUV movement must be obtained.

1) Motor Control

The first design objective was to establish an effective method of controlling the revolutions per minute (rpm) of each of our eight vectored T200 thrusters.

a) Method 1:

The first design streategy for motor control was to rework the code that was prevviously used. We attempted to use a drone flight controller, the Pixhawk Px4, which has an inertial attitude and heading reference system (i-AHRS), as well as PWM signal outputs. This self contained package theoretically had all of the capabilities required to control the AUV. The cost of this system is accessibility. The method to interface with the control software and sensors for use in an autonomous vehicle was extremely obfuscated, making low-level integrated challenging. After spending six months of effort with little success, the team elected to scrap all effort that had been made with the system over the past three years.

b) Method 2:

The Pixhawk PX4 flight controller has several design features that were the foundational requirements of the novel motor control solution. The requirements included:

- A minimum of 8 PWM signal outputs.
- Proportional, Derivative, and Integral (PID) based control software.
- A replacement for the i-AHRS

With these basic engineering requirements, the following solution was established.

Using a ROS implementation of a PID controller [1], a reliable of way mapping sensor outputs to control efforts was created. Through simple tuning and connecting control efforts in ROS to a Pololu PWM signal board via a universal serial bus (USB), a simple but robust method of controlling motor speed was developed.

2) Control Movement in the Rotational Axes

The second design milestone was to establish reliable movement across all rotational axes on the surface of the water. This was required prior to other movement profiles because the tuning of the other PIDs will be dependent on the angle of the thrust vectors. Therefore, the pitch, roll, and yaw must be controlled.

3) Control Movement in the 2D Plane

The third milestone was to sort out navigation in a 2D plane on the surface. While it is a competition requirement to submerge, these tests were conducted on the surface for easier observation and testing. For all practical purposes, navigation on the surface is identical to navigation when submerged.

4) Control Submerging and Surfacing

Finally, the AUV is required to conduct the operations for competition while submerged. This as simply accomplished by adding a control effort to apply a force against the force of buoyancy with the thrusters.

C. Implement a Reliable Sensor for Detecting Objects

The final goal was to implement a more reliable method for object detection. In previous years, we have only used cameras due to their low cost and ease of implementation. However, their low fidelity signals for the target environment causes issues when feeding the data into algorithms. As a result, we were unable to accurately do any object following task with any degree of repeatability.

For this design iteration, we will be focusing on Sonar. It is reliable for the localization of objects in the target environment. This comes at the cost of effective classification, but this is a worthy trade off because our target tasks do not rely on pure classification.

II. Competition & Design Strategy

A. Gate

The first objective to overcome is to solve the gate task. This is a prerequisite to solving other tasks. The current strategy is to use our scanning sonar to find where the gate is relative to the AUV.

Funding Provided by the College of Engineering and Department of Electrical Engineering and Computer Science at Embry-Riddle Aeronautical University

B. Coinflip & Style

The Coinflip and Style tasks complement our updated controls software. The task will serve to demonstrate our new capabilities.

C. Buoys

The buoys tasks are near the limit of our current capabilities. While the identification part of the task if beyond our current capabilities, the localization portion of the challenge will utilize both our updated control system and sonar capabilities.

III. Technical Systems

A. Custom Control System

The control system is based on simple but robust PID controllers. We have one controller mapped to each degree of freedom. The six degrees are:

Rotational

- Roll
- Pitch
- Yaw

Translational

- Heave
- Surge
- Sway

A separate controller must be initialized and tuned to control each of these axes.

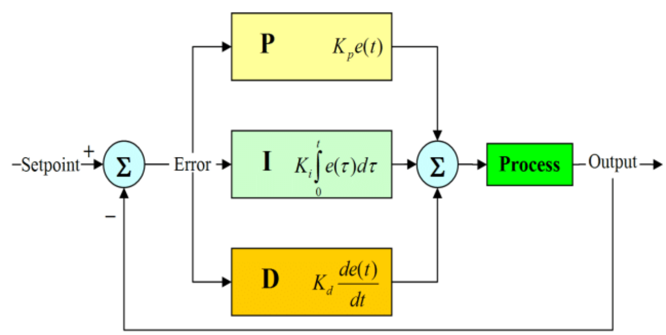

Figure 1 [3]

The PID controller works through a traditional control loop, where an error is calculated through the difference between a setpoint and a plant.

$$S - P = e \quad (1)$$

Where S is the setpoint, P is the plant, and e is the calculated error.

For each of our axis, the control effort is an increase of decrease to the PWM (Pulse-width modulation) signal that drives our T200 Thrusters. The tuning constant, K_p, K_i, and K_d, influence how the AUV will respond to induced error. The proper tuning parameters can be found through simulation or hardware experimentation. For this year, we used hardware experimentation to find functional parameters.

Once tuned, these control efforts were summed for each motor individually. This was done to ensure competition control efforts would reach equilibrium. For example, the thrusters that controlled pitch also controlled depth, therefore, to maintain pitch and depth, the from motors would have to apply more force than the rear thrusters.

$$\sum E = E_R + E_P + E_Y + E_H + E_{Su} + E_{Sw} \quad (2)$$

Where E_R is the control effort for Roll, E_P is the control effort for pitch, E_Y is the control effort for yaw, E_H is the control effort for heave, E_{Su} is the control effort for surge, E_{Sw} is the control effort for sway. These are all summed for the control effort (PWM Signal Gain) for each of our eight thrusters.

B. Sonar System

Our sonar system comprises of one sensor from Blue Robotics, the Ping360 Sonar. Seeing as we haven't implemented a standardized communication protocol, i.e. RS-232 or CAN, for our internal communications network opted to use IEEE standard 802.3u, 100Base-T. This provided us with enough bandwidth and room for growth should our scanning capture and/or processing algorithm require. The driver takes in a value of a max scan distance, start and end angles, and a few other parameters. The proper transducer transmission duration and sample period are then calculated using the speed of sound through water and a few other variables. The sonar then sweeps the degrees specified and broadcasts all resultant scan data onto a ROS Topic. Currently the scan data reception and publishing is done on a single thread. Our hope is to make this driver multi-threaded so we can dedicate one thread to data capture then the other to data publication and processing.

IV. Testing Strategy

A. Dry Testing

Most of the testing conducted took place out of the water and was simplistic. Computer simulations are beyond our capabilities at the moment. To effectively test our systems, we would have individuals hold and rotate the AUV in the air in response to commands sent by our software.

B. Wet Testing

We conducted seven water tests throughout the development season. The goals of these water tests were to tune our control systems before the end of the semester, and we would be barred from working on the project until competition. We successfully tuned our control systems and established reliable movement for the first time in our AUV.

C. Corrective Measures

With the results from each of our experiments, we systematically expanded the operations restrictions to a state where we could control the AUV in six-degrees of freedom without having an error for normal movement.

Acknowledgment

The accomplishments of this year would not have been possible without the help of Wanda Rodriguez, Bill Russo, Dr. Brian Butka, Department of Electrical Engineering and Computer Science, the Lehman College of Engineering, the Embry-Riddle Student Government Association.

REFERENCES

[1] A. Zelenak and P. Bouchier, “PID,” ros.org, http://wiki.ros.org/pid (accessed Jun. 10, 2023).

[2] M. Quigley et al., “Ros Melodic,” GitHub, https://github.com/ros/ros/tree/melodic-devel (accessed Jun. 10, 2023).

[3] N. Mehta, D. Chauhan, S. Patel, and S. Mistry, “Design of HMI based on PID control of temperature,” *International Journal of Engineering Research and*, vol. V6, no. 05, 2017. doi:10.17577/ijertv6is050074