Design of an Autonomous Submarine for the RoboSub 2021 Competition

Prepared By:
Amaar Quadri, Dante DiGiuseppe, Dhrumil Parikh, Andy Xu

200 University Ave W
Waterloo, Ontario, N2L 3G1
June 27, 2021

Table of Contents

List of Figures

Summary

Waterloo Aquadrone is a student design team at the University of Waterloo that is working on creating a fully autonomous submarine for the RoboSub competition. Since the submarine must be fully autonomous, a need existed for a sophisticated software and controls system with many different components to successfully control the submarine and complete competition objectives.

A needs analysis was performed considering the requirements for the software as well as the limitations due to the available hardware. Next, several different systems were developed in parallel. A machine vision system was developed for detecting, identifying, and calculating the relative distance to competition objects in the environment. A simultaneous localization and mapping system was researched and implemented to ingest data from sensors and the machine vision system and build a map of the environment including the submarine's location within it. A modular hierarchical path planning system was designed and implemented to provide the decision making behind the complex series of tasks required to complete the competition challenges. A multi-step controls system was designed, implemented, and tuned to convert the movement instructions from the path planning system into thrust commands for each of the 8 propellors to successfully execute the desired maneuvers. Finally, the entire software system was joined together using Robot Operating System (ROS) and tested in the Gazebo simulation environment.

1. Introduction

1.1 Waterloo Aquadrone Design Team

Waterloo Aquadrone is a student design team at the University of Waterloo that is working on creating a submarine to compete in the RoboSub competitions. The team was started in the March 2019 by a group of dedicated students from the mechanical engineering department but has since grown to include many students from across several different disciplines.

Despite being a relatively new team, Aquadrone has already built their first submarine and tested it underwater. The team has also designed a torpedo firing system, and a robotic arm for use in the various competition tasks. The team has also designed a sophisticated software system for controlling the submarine.

Figure 1: The SolidWorks model of the submarine (left) and the physical submarine (right).

1.2 Competition Strategy

Since the team has not competed in RoboSub before, our goal was primarily to design and build a functional submarine with basic navigation, movement, and vision capabilities. This would allow us to complete the basic tasks such as the gate, buoys, surfacing, and torpedoes challenges. The submarine was nonetheless designed with a robotic arm, although achieving accurate enough autonomous control of the submarine to use it effectively was left as a stretch goal.

Since this is our team's first submarine, this report outlines the overall design.

2. Mechanical Design

The mechanical design of the submarine was primarily done in SolidWorks. The following image shows the computer aided design (CAD) of the submarine.

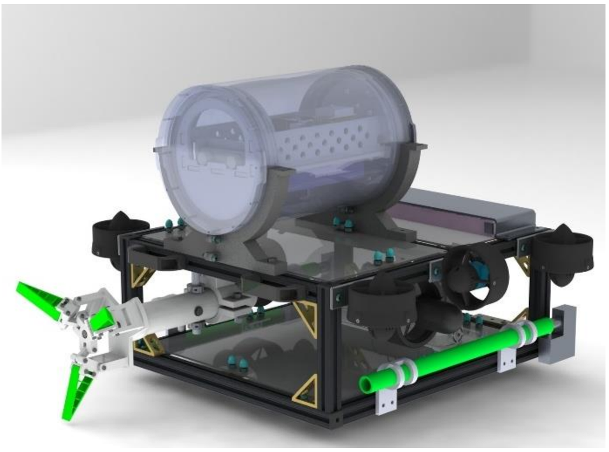

Figure 2: SolidWorks model of the submarine.

2.1 Frame Design

The frame was designed with the goal of being a robust platform for mounting the rest of the components of the submarine. The main structure consists of 12 extruded aluminum rails in the shape of a rectangular prism. The aluminum chosen was galvanized to protect against corrosion from salty water. The top and bottom surfaces have sheet metal plates bolted to them to securely mount other components. The frame also includes four handles for easy carrying of the submarine.

Another key consideration when designing the frame was to manage the buoyancy of the submarine. Volume calculations were done in SolidWorks and compared against the mass of the submarine to ensure that the submarine would be positively buoyant. This is necessary for safety and logistics considerations so that we can avoid a scenario where the submarine gets stuck at the bottom of the pool.

Also, the center of volume of the submarine, which is where the net buoyancy force will be applied, was calculated in SolidWorks, and components were positioned such that the center of volume is slightly above the center of mass of the submarine. This ensures that the torque due to buoyancy will be small, and act towards stabilizing the submarine to an upright position.

2.2 Torpedoes Design

The torpedo was designed to perform the “survive the shootout” task. The two identical torpedoes are loaded into two barrels, located on either side of the sub. A set of magnets on the barrel and ferromagnetic wire in the torpedo hold the torpedo in place until fired. To fire the torpedoes, the sub computer opens a solenoid and compressed air from the pneumatic system accelerates the torpedo out of the barrel.

The torpedo is machined from two pieces of 6/12 nylon. It has a blunt front to minimize the chance of bruising someone, and laser-cut acrylic fins at the rear. The purpose of the fins are to stabilize the torpedo by moving the center of pressure behind the center of mass. The two nylon pieces come apart to access a hollow chamber on the inside, filled with a neoprene foam chord. The foam chord is sized so that the density of the torpedo is slightly less than water, making it slightly buoyant.

2.3 Robotic Arm Design

The robotic arm has the goal of enabling the submarine to interact with objects in the environment, such as by pulling the many levers in the competition challenges. The arm consists of 3 fingers that can be opened and closed using a pneumatically powered linear actuator.

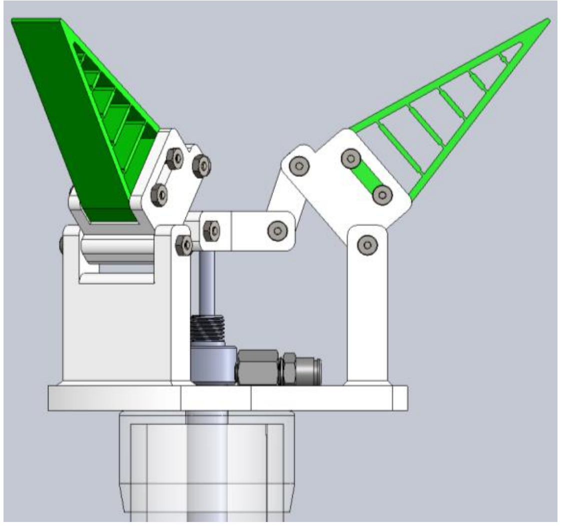

Figure 3: The linkage used to convert the movement of the linear actuator into rotation of the fingers.

The fingers themselves were 3D printed using thermoplastic polyurethane (TPU), which is a flexible material. This allows the fingers to deform to better fit the shape of the object being grasped, as shown in the following figure.

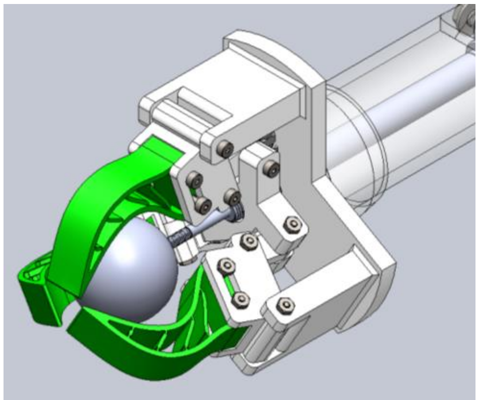

Figure 4: SolidWorks simulation showing the deformation of the fingers to better clamp around objects.

2.4 Pneumatic System Design

The pneumatic system powers the torpedoes and manipulator arm, which require a minimum pressure of 50 psi to function properly. The pressure of the system will drop as air is used, so the initial pressure of the system is 115 psi. The air tank was chosen to be 547 mL. Flexible polyurethane tubing was used to connect parts of the system across the sub. A pressure release valve is included that prevents the system from exploding by releasing air if the pressure exceeds 125 psi. Solenoid valves connected to the sub computer control the flow of air to the torpedoes and manipulator arm. The sub computer uses data from the digital pressure gauge to calculate the length of time to open each valve. An analog pressure gauge is included to read the system pressure while charging and while testing without a computer.

Several tests were performed to assess the performance. A charging test tested all the valves of the system. Also, a leak test was performed to locate air leaks. All threaded connections were resealed with PTFE tape which successfully reduced the leaking to a negligible rate.

3. Electrical Design

4. Software Design

4.1 Robot Operating System

4.2 Machine Vision System

4.3 Simultaneous Localization and Mapping

4.4 Path Planning Framework

The path planning system represents the submarine's core decision making logic. It is responsible for inputting the submarine's location as well as the map of the environment from the EKF and using that to decide what commands to send to the submarine's movement control systems. To do this effectively, the path planning system needs to have short term plans, like moving forward a few meters, as well as long term plans such as maneuvering through the gate, finding the torpedo target, aligning itself to shoot the torpedoes etc. To make this wide range of complexities manageable, a hierarchical and modular state-based methodology was devised.

The path planning system consists of several independent states. Each state has a very small, specific task to carry out such as diving to a certain depth, moving forward a specified amount, or simply waiting for a specified amount of time. Each state is represented by a Python class, which specifies that state's behaviour. All states are derived from a common superclass called the base state.

First, several simple states can be defined including:

- A waiting state, which waits for a specified amount of time
- A go-to-depth-state, which moves the submarine to a specified depth in the water
- A go-to-orientation state, which rotates the submarine to a specified orientation
- A travel state, which moves the submarine to a specified location
- A relative travel state, which moves the submarine to a specified offset from a known object in the environment

Next, several meta-states called state machines were defined. These state machines take other states as parameters and combine them in some way. For example, the sequential state machine takes as input a list of other states. It then runs those states one after another. Similarly, the parallel state machine takes as input a list of other states and runs them at the same time. Other state machines can conditionally execute different states depending on the exit code of the first state, allowing for more complex decision making. Importantly, these state machines themselves act as regular states, which is reflected in the code by them being subclasses of the base state. This allows for states and state machines to be combined and nested as much as desired, which leads to much more sophisticated behaviour.

4.5 Control Systems

The control system for the submarine receives 3 commands from the path planning system: the target depth, the target orientation, and the target x-y position. Together, these three commands dictate the target for all 6 degrees of freedom of the submarine.

The submarine can also send instructions to open and close the claw of the robotic arm, and to fire the torpedoes. However, these tasks simply consist of sending the appropriate command to the attached hardware device and will be handled by a different section of the software, so they will not be discussed here.

The first step is to convert each of the three movement commands into wrenches. A wrench is a common data structure used in ROS which simply consists of a list of 6 numbers: the forces in the x, y, and z directions (in Newtons) followed by the torques about the x, y, and z axes (in Newton-meters). For the linear degrees of freedom, this is done using a proportional-integral-derivative (PID) controller.

The rotational degrees of freedom are more complicated. One common method for working with orientations uses the Euler angles roll, pitch, and yaw. Although this representation is convenient in that it is easy for humans to understand, it suffers from non-linearities and gimbal lock: a situation where one degree of freedom is lost due to unfortunate placement of the other two degrees of freedom.

A superior method uses quaternions, which are a 4-dimensional number system that extends upon the complex numbers. A PID controller done in the quaternion coordinate system will avoid many of the issues faced by a PID controller that uses Euler angles, and leads to overall smoother rotations of the controlled object.

Finally, the wrenches are converted into a list of thrusts for each of the individual propellors based on the propellor geometry. This is done using basic linear algebra and a matrix inversion.

6. References

Solà Joan. “Simultaneous localization and mapping with the extended Kalman filter.” 2014. http://www.iri.upc.edu/people/jsola/JoanSola/objectes/curs_SLAM/SLAM2D/SLAM%20course.pdf.

Graf Basile. “Quaternions and Dynamics.” 2007. https://arxiv.org/pdf/0811.2889.pdf.

Fresk Emil, Nikolakopoulos George. “Full Quaternion Based Attitude Control for a Quadrotor”. 2013. http://ltu.diva-portal.org/smash/get/diva2:1010947/FULLTEXT01.pdf.

Van Biezen Michel. “Special Topics 1 – The Kalman Filter.” 2015. https://youtube.com/playlist?list=PLX2gX-ftPVXU3oUFNATxGXY90AULiqnWT.

Boyd Stephen. “Least-norm solutions of underdetermined equations.” 2007. https://web.stanford.edu/class/archive/ee/ee263/ee263.1082/lectures/min-norm2.pdf.