

San Diego Robotics 101: Design and Implementation of CubeCeption 2.0

Ryan Ganton (Captain), Jason Ganton, Rahul Keyal, Jason Ma,
Tushar Pankaj, Patrick Paxson, Rahul Salvi

Abstract - Cubeception 2.0 is an autonomous underwater vehicle (AUV) built by a team of seven high schoolers. As an improvement over Cubeception 1.0, Cubeception 2.0 was rebuilt over a five month design cycle with the use of CAD software and 3D printing. Cubeception 2.0 addresses several design flaws in previous designs while maintaining a unique cube layout which allows for multiple orientations and rotations to take up the same space underwater. Featuring new connectors, a redesigned easily removable electronics rack, upgraded processor boards and six degrees of freedom, Cubeception 2.0 is markedly more robust and capable than its predecessor.



I. Introduction

San Diego Robotics 101 participates in the Annual International Robosub Competition presented by the AUVERSI (Association for Unmanned Vehicle Systems International) Foundation. This entails designing and constructing an autonomous underwater vehicle (AUV) to prioritize and perform various tasks similar to those performed by professional AUVs. This year, tasks range from detecting and touching colored buoys to dropping markers in bins to shooting torpedoes at marked targets, challenging teams to exhibit feats of mechanical, electrical, and computer engineering as well as collaborate in order to build competent AUVs. Each AUV must accomplish these tasks independently, which requires the use of computer vision algorithms, precise maneuverability, and the ability to handle certain objects well. To create such an AUV, San Diego Robotics 101 is a highly interdisciplinary team in which many members have multiple roles in Mechanical, Electrical, Business, and Software subteams due to the relatively small team size.

II. Mechanical

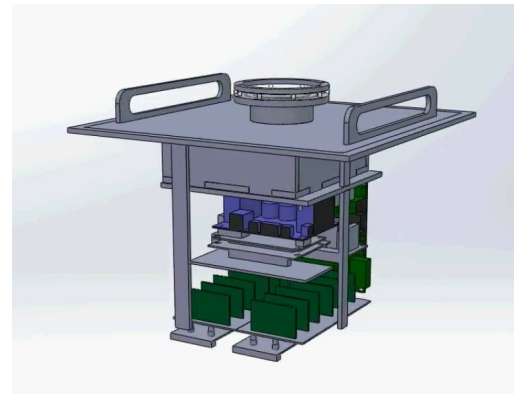
Cubeception 2.0's mechanical design provides protection to water sensitive electronics while maintaining a large surface area for connections from the inside to outside of the central Aluminum cube. Motors and other external components are also shielded from impact by a PVC frame. The design was assembled in SolidWorks and then 3-D printed or manufactured.

A. Aluminum Cube

This makes up the Hull of our robot, separating the electronics, and batteries, from the water outside. We noticed many competitors from last year had designed flanges on their opening ports. We found through our own research that this would help solve some of our leakage problems. Overall the sides of the cube are 3/16th of an inch thick, except for the flange/lid which is 3/8ths of an inch.

B. Core/Electronics Rack

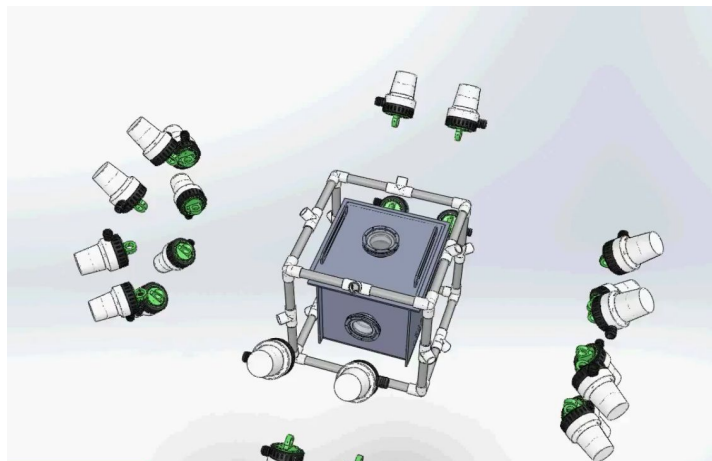
A new feature this year, a necessary upgrade from last year, implementing a rack has proven to be very useful. The core can now be easily removed from the hull of the robot and allows for maintenance and change of batteries. The circuits, and wires, are all set on different 'levels' and are an attempt to make the electronics fit in the smallest area possible, to optimize size and volume.



C. Frame

*Outer Frame: The outside frame is designed to act as as a bumper, and shield the hull and bilge pumps from impact. Future plans include using the frame as a 'tactical rail', attaching things like manipulators, torpedoes, landing markers, flashlights, etc.

*Inner Frame: The inner frame is designed for attaching to the Hull of the robot, and is also currently used as a 'tactical rail' for attaching all the bilge pumps.



D. Bilge Pumps

Last years Bilge pumps were a great success, and as such, we decided to use them again. They were a cheap, and decent

alternative to thrusters. 24 were needed since they only fired in one direction, 4 for each face, so with 24, we get all 6 degrees of freedom.

E. Connectors: For the wiring in and out of the robot we are going with a unified port system that when screwed together, tightens on an O-ring and makes the seal. We had each wire utilize a separate connector, so that if one fails, it's easy to replace. The connector design was done using SolidWorks, and the final version was 3-D printed, and uses an M14 nut and an O-Ring o ring of 14mm inside diameter, and 2mm thick.

III. Electrical

- A. Power: A custom circuit board was designed to provide power to the bilge pumps and electronics on the vehicle. It features clean 3.3V, 5V, and 12V supplies for the digital components. The "dirty" power side for the bilge pumps is separated as it is harder to keep smooth and sudden changes in voltage could damage the sensitive electronics.
- B. Bilge Pump Control: Each bilge pump is controlled individually by the PWM controller in the FPGA. To produce the signals for each pump, several PID controllers are used. PID control is an effective method of calculating the output required for the desired motion. Lateral motion in 3 axes is produced by running all four pumps on a side simultaneously, and rotations are produced by running only a select few on each side. Combining inputs from the IMU and pressure sensors allows the vehicle to remain at the desired orientation and depth while moving laterally.
- C. Power: A custom circuit board was designed to provide power to the bilge pumps and electronics on the vehicle. It features clean 3.3V, 5V, and 12V supplies for the digital components. The "dirty" power side for the bilge pumps is separated as it is harder to keep smooth and sudden changes in voltage could damage the sensitive electronics.
- D. Bilge Pump Control: Each bilge pump is controlled individually by the PWM controller in the FPGA. To produce the signals for each pump, several PID controllers are used. PID control is an effective method of calculating the output required for the desired motion. Lateral motion in 3 axes is produced by running all four pumps on a side simultaneously, and rotations are produced by running only a select few on each side. Combining inputs from the IMU and pressure sensors allows the vehicle to remain at the desired orientation and depth while moving laterally.

IV. Software

- A. Computer: We use the Nvidia Jetson TK1 computer running Ubuntu Linux as our primary processor for high level instructions such as camera image processing, keeping track of the current task, and communicating to the lower-level processor to drive the motors. We use the Xilinx MicroZed with an FPGA as the lower-level processor. The FPGA has PWM outputs to the motor controllers and SPI inputs from the various sensors, such as the pressure sensor and the IMU. The board also has a PetaLinux environment that runs the communication to the Nvidia board over Ethernet using UDP, and the PID control loops that use the processed sensor data from the FPGA to produce the PWM outputs to the motors. The board does not have knowledge of the missions, but instead receives commands from the Nvidia board telling it which direction and speed to move and rotate.
- B. Image Processing: The image processing is done on the Nvidia board using the OpenCV library. We have a USB 3.0 camera that the board records images from and uses algorithms such as thresholding, edge detection, and contour detection to determine the location of the robot relative to visual landmarks and to determine the configuration of

the landmarks (i.e. the color of the buoys). The computer vision algorithms are not complete at this time as the competition is not for a month, but if time permits, we are planning to use neural networks to process the images of the various landing bins.

- C. Logging: Logging is done by both the Nvidia board and the MicroZed board in order to analyze the data generated by the test runs.
- D. Mission Planner: Mission planning is done by the Nvidia board, as this is the only board that has knowledge of the missions. It keeps track of which missions are done and sends low-level commands to the MicroZed.
- E. Movement: Movement is controlled solely by the MicroZed. While the Nvidia board sends commands such as the desired translational and rotational velocities to the MicroZed, the MicroZed runs PID control loops to convert the desired velocities from the Nvidia board into PWM commands while incorporating the sensor data from the FPGA to account for drift and other unpredictable errors. The MicroZed then sends the PWM commands through the FPGA to motor controllers, which provide the connection from the motors to the battery.

V. Sensors

- A. Pressure: Two MS5803-14BA sensors are used to accurately determine the water pressure. This data is then used to calculate the vehicle's depth.
- B. IMU: A MPU9250 9-axis IMU is used to determine the vehicle's orientation in the water. This sensor includes an accelerometer, gyro, and magnetometer. Data from each of these components is combined and filtered for accuracy.

VI. Acknowledgements

First and foremost we would like to graciously thank SolidWorks. Our entire robot is cadded in their software. This made things simpler for machining the aluminium and for the 3-D printing; Apollo Manufacturing for allowing us to use their facility to stuff our custom boards; Vinatech for their awesome welding job to make sure we stay waterproof; and MetalEtch for their stencil services.