The Design of Iron Man: The RoboSub Club of the Palouse's Autonomous Underwater Vehicle

Abstract

The RoboSub Club of the Palouse's 2014-2015 autonomous underwater vehicle (AUV) is the culmination of vast improvements made to previous years' designs. The team built upon and modified the 2013-2014 submarine model to be more modular, efficient, and to lay out a better foundation for future teams to work on and learn from. The team, comprised of 30 members from electrical engineering, mechanical engineering, and computer science, completed a ten-month design cycle to complete this year's model. Members spread across two universities, Washington State University and the University of Idaho, and ranged from freshman to senior class standings. Though the design of the AUV poses many electrical, mechanical, and software challenges, our team was prepared to take on whatever challenges came their way. This year we are proud to present our new an improved AUV, Iron Man!

Electrical Engineering

Circuit Board Redesign

One of the key changes within the electrical subsystems revolves around a redesign of the internal circuit boards within the sub. In the previous design, a single board acted as a master control board with most of the functionality for the sub implemented on it. This board was supported by external regulators for voltage supply and small prototyping areas with additional circuits. This design was simple and worked well, however issues arose when changes to the design were required. One of the negative aspects of having a single board is that whenever a single change needs to be made, the entire board must be refabricated. To combat this issue, we split apart the main functionalities of the previous master board into a number of separate boards. In the new design, we have created separate boards for: UART-USB communication, sensor input, voltage regulation, thruster kill signals, master kill signals, pneumatic control, battery management, and notification LEDs.

Having boards with a single function simplifies the process of upgrading and changing the system as a whole, as smaller elements of the whole system can be changed and replaced while retaining other parts of the subsystems. One of the issues encountered with utilizing multiple boards is that a larger amount of area is required. To increase modularity and reduce space constraints, we have implemented edge connections on the circuit boards to allow vertical slotting of the boards in the sub. This frees up a large section of the interior for additional prototyping space for further development and allows for easy replacement of boards. If components on any of the boards fail or the design must be updated, it is a very simple matter to pull the board out of the slot and place a new board in its place without removing any screws or having to wire connectors.

After implementation of the edge connectors, it was discovered that wiring had not been simplified; often, a single signal would need to be fed to multiple boards. This large amount of redundancy would cause issues in understandability and maintenance of the electrical systems. To address this issue, we have elected to create a motherboard that acts as a connector for all of the other boards. With this implementation, a single wire can be connected to the motherboard, and the motherboard can then break that signal apart into multiple traces to be fed into the different boards requiring it. A perfect example of this is the 3.3V power. Without the motherboard, 3.3V would have to be broken out into 8 separate connections and then connected to each of the other boards. With the motherboard implementation, all of this wiring can be consolidated within the board itself. A single power input line can be supplied and sent to the regulator board, which then in turn gets regulated down. The motherboard then traces from the regulator board the 3.3V output to each of the other boards 3.3V inputs, resulting in an immense reduction in overall required wiring.

One of the drawbacks of the motherboard design is that it reduces overall modularity in the design. Because of the specialized nature of the motherboard, it will be more difficult to change pin mappings on future boards, as only certain pins are connected within the board itself, and changing the routing post-fabrication is not easily possible. To address this issue, a standardization for the circuit boards was implemented, dedicating certain pins on all boards as voltage inputs of 12V, 5V, 3.3V, GND, UART receive, and UART transmit. By dedicating these pins, a larger level of modularity on the motherboard can be achieved. Future plans for the motherboard include universalizing the entire board for year-to-year sustainability by providing on-board jumpers between each of the different non-reserved output pins on the board to allow for custom interconnections based upon design.

Sensor Suite

A primary component of the submarine is the capability to track its own location, permitting it to make decisions about its movement to approach obstacles and achieve objectives. This requires a suite of sensors capable of providing sufficient information to determine a state and a robust localizer module to provide sufficient confidence in the determination.

The sensor suite consists of a 3-axis digital accelerometer (Adx1345), a 3-axis angular-velocity gyroscope (ITG3200), a 3-axis magnetometer (HMC5883L), an analog depth sensor, and a hydrophone array configured for differential time-of-arrival measurements.

The three 3-axis sensors are contained on DFRobot IMU chip, and report to the submarine through an I2C channel. The accelerometer has 13 bit resolution and has a dynamic range of +/- 16g. This provides the submarine with an external orientation, an internal angular velocity, and an internal acceleration.

The depth sensor is mounted inside the submarine, and samples the water pressure through an aperture in the bulkhead. It receives a 12V signal and its pressure transducer encodes the information about external pressure into a current, which is run across a resistor to be measured as a voltage. It is interpreted to report absolute depth.

The Hydrophone array consists of three hydrophones: an x-axis hydrophone, a y-axis hydrophone, and a reference hydrophone, arranged together in an L-shape. This provides two differential time-of-arrival measurements that each map to a hyperboloid of potential of signal origins in the x-y-z axis. Together, these two measurements provide a very accurate bearing to the pinger. When combined with information pulled from the depth sensor, the system can perform rangefinding as well, mathematically resolving mostly-unambiguous xyz coordinates of the source of the ping. Having spatial reference to a fixed point, though noisy, provides some additional information to help bound drift of the dead-reckoning system.

At the root of the localization is an Extended Kalman Filter (EKF) which operates on a 12-state state-space model consisting of an x,y, and z location, a roll, pitch, and yaw orientation, and the first derivatives of each. The physical model used for the a priori guess every cycle incorporates drag, buoyancy, and inertia into the transition matrix, and uses the commands sent to the motors as an input, interpreted by the input matrix accounting for thrust, position, and scaling of the motors.

The posterior measurement stage of the EKF incorporates the measurements for absolute orientation, rotational velocity, and translational acceleration, combined with a depth sensor for absolute z-axis location. Fusing this data together several times per second permits a close, reliable tracking of the submarine's location, orientation, and translational and rotational velocities, subject to some drift. When the ping is detected by the hydrophones, the data is combined with the submarine's current position and helps to bound the drift.

Microcontroller Code Redesign

Several major changes were made to the microcontroller code from last year. First and foremost, the code was in major need of cleanup and reorganization. Functions that interfaced with on-chip peripherals were redone in a library format, and rewritten in a way that maximizes code reuse. Emphasis was placed placed on efficiency and modularity. Our resultant library allows us to use our PIC32 microcontroller's timers, UART, ADC, and SPI modules via a consistent interface. All modules use interrupt service routines (ISRs) such that hardware is being used to its full potential. Multiple processes can utilize these peripherals at once via "work queues" that are then processed by their respective ISRs, with long-running tasks occurring in background processes. This modularity was made possible via the extensive use of callback functions and circular byte-queues.

Another large change was the packet-based communication scheme with our computer, which is necessary for the proper interpretation of data between the computer and microcontrollers. Last year we used a fixed-length packet format, but we found this to be limiting. We have since developed a new packet format that allows for variable-length packets up to 255 bytes large. This allows us to combine more relevant data into the same packet, increasing overall system efficiency. Our microcontroller library also includes support for sending and receiving these packets, making the protocol transparent to the user.

Using this library, software development for each microcontroller was rapid. Each board's software was able to be written and completely tested in a matter of hours.

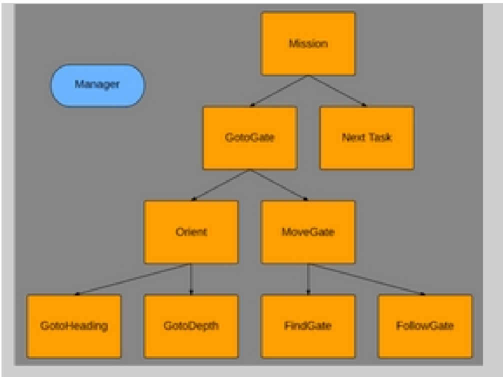
Computer Science

Background

The Washington State University Robosub Computer Science team has been dedicated to working on completely changing most of what was done in previous years. This includes completely rewriting vision, artificial intelligence, communication, and testing systems. The underlying mindset when developing this year's software was creating a strong framework that could be built upon in future years and would not need to be modified heavily or replaced. The software from previous years was inflexible, non-extensible, and overall incomplete. In particular the previous artificial intelligence (AI) and vision implementations were simply static objects that resembled more of a library than a runtime architecture. It is this issue that led us to developing our new vision and AI systems.

Artificial Intelligence

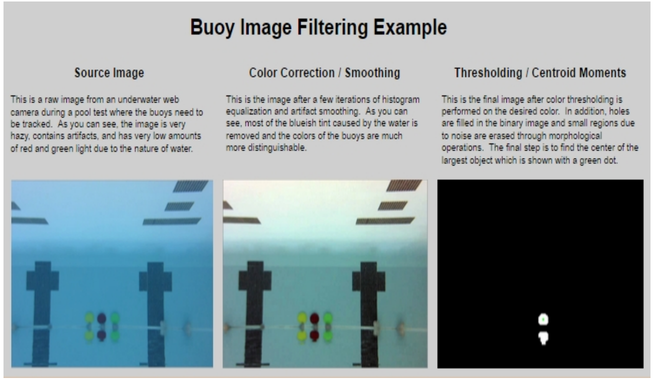
Because last year's team had a very minimal AI system, we decided to start from scratch on a new system, choosing to write the AI in Python because it is a scripting language. We designed the AI using a tree like structure with a visitor class that runs through the leaves in the tree. At the root of the tree is the Mission, the competition objectives we wish to complete. These objectives correspond to the pool obstacles, eg. the start gate, the buoys, etc. Under the Mission are the tasks, which are broken down into various steps for the mission, such as turn and move toward the start gate. Under those are the subtasks, which are divided into more subtasks until each is a primitive Robosub command that can be sent to movement, such as turning left. The mission manager, the visitor class for the tree, then traverses the tree and sends commands to the movement module which tells our submarine what to do. The completed tasks are then marked and the mission manager moves on to the next task until every subtask is completed, at which point the mission is marked as completed. Any of these missions can be easily modified or replaced without having to rewrite the Mission Manager class.



Localization is another module we are designing which has a close relation to AI. Localization allows the AI to find the Robosub's location relative to a known point in the pool. Before making movement decisions the AI uses the localization system, as well as information from the vision system, to make a decision as to where to move to. We accomplish localization by taking the data from the pinger in the pool and estimating the Robosub's position. Localization will also be accomplished by creating a map of the pool and marking the locations of the different missions with (x,y,z) coordinates.

Vision

The computer vision system was designed around a multithreaded environment that could be controlled through a central module. Processes can be spawned through broker commands and utilize a shared memory segment for frame buffers obtained through each of the cameras. To meet strict performance requirements, vision processes also utilize custom built image filtering trees that minimize copying of data, and allow runtime optimizations of algorithms through OpenCL. Based off the currently selected AI mission the vision loads up the corresponding filter tree and grabs the filtered image, demonstrated below is an image run through the filter tree, and grabbing the center of any object found (indicated by the green dot). In competition the vision would grab the (x,y) coordinates of an object on screen and send that to the AI module.



Movement

Our movement module is similar in design to last years movement code, but this year our Computer Science team consolidated the code and decided on using a fuzzy logic control library called FuzzyLite. Fuzzy logic is perfect for aquatic navigation because it is able to operate in an environment full of uncertainty. The movement system's main purpose is to convert high level commands sent by the AI module into primitive thruster values.

To best accomplish its goals the movement system is conceptually separated into stabilization and movement. The stabilization system takes location data from the localization system, and a desired location from the AI, and calculates the best path to take. The system then converts this into actual thruster values.

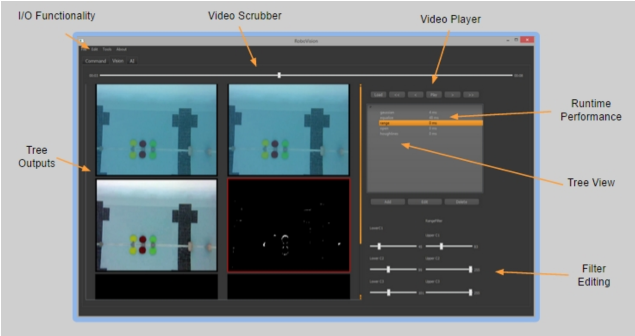
Broker System

The communicator is an object used by the modules to trade messages with the broker. Both the broker and communicator use ZMQ for their communication using sockets to send strings over ports. The broker receives all messages from modules then sends the correct message to the recipient module. All of these messages can also be serialised into C++ objects using RapidJSON.

This years decision to implement a broker system was done for a few reasons. In the previous year the communication system used a publisher-subscriber message passing design. Information for each module was held in a global settings file. The downside of this design was that it was fragile and hard to change. If a single module was removed or renamed every reference to it would have to be removed from every other modules settings. By having a broker, we can dynamically add or remove modules to the system on the fly. Additionally modules can now also send messages to specific recipients, increasing performance of the sub.

Tools

This year the Computer Science team also designed several tools to aid in testing and improving our software. One of our software tools developed is a graphical user interface (GUI) written in C++ using QT Creator. The GUI displays mp4 files or video streams run through several image filters. The GUI utilizes a tree structure to hold these different filters which allows for testing multiple combinations of filters at a time to find the optimal combination to complete tasks at competition. Another useful tool created this semester is our logger, which logs all communication made between modules. The logger was deemed necessary because of its utility in debugging our overall communication, as well as debugging individual modules.



Mechanical Engineering

Slim Redesign

When trying to repair and upgrade systems on the sub it became apparent that the layout was inefficient. Reaching some parts was difficult due to space constraints. Also removing some systems required other systems to be removed first. To abate this problem, the sub chassis was redesigned to allow for a more disc shaped layout. This gives easy to reach access to all systems without the need to remove other systems. In addition to the increased usefulness, the new chassis design also moved the thrusters to provide a more distributed moment to help avoid barrel rolling.

Camera Housing

With vision upgrades being one of the top priorities for the team this year, the Computer Science and Electrical Engineering teams chose to upgrade our cameras to Point Grey's Flea3 for the 2014-2015 submarine. With this the mechanical engineering team has designed and built waterproof housings that properly stabilize the cameras while protecting them from not only water, but other types of physical damage. Three cameras will be mounted in different locations around the sub. Stable mounting will be achieved via 3D printed hinge brackets. Locations for camera's includes two forward facing for main vision and one on bottom facing for location of target/deployment of markers with the newly designed marker dropper system.

Marker Droppers

The previous years marker droppers were integrated into their forward claw which we wanted to separate and design independent systems. The design uses a two way pneumatic actuator fixed on one end and attached to a sliding gate on the other. When the submarine enters the water the position is closed. Once the sub reaches the obstacle requiring it's operation, which the camera and AI identify, the cylinder is actuated, pulling back the gate and letting the marker drop into the designated target location. With how simple this system has been built, we not only hope to have high target efficiency, but also high reliability.

Hoist Connection

Last year's team used 1/4" rope for the hoist connections when picking up/setting down the submarine into the pool. This configuration confused the divers and lift operator at last years competition, which nearly cause one of our thrusters to break while the sub was being lifted. To avoid this confusion, as well as making a safer system, we have decided to simply remove the rope. Instead we use eye bolts. Not only do they have much more strength but this will be easier for the divers to identify how to hook up the sub.

Battery Tube

While the previous design included an aluminum cylinder that housed the battery, we decided to change the material to acrylic as well as include blue LED's for a more aesthetically pleasing design. We also chose to remount the battery tube in the aft of the submarine allowing for the subs height to be condensed and the weighting to be more evenly redistributed.

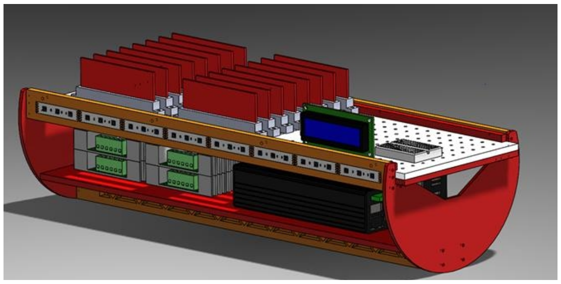
New Pneumatic Housing

The pneumatic housing is now a hybrid 3D printed/acrylic sheet box what is capable of holding 6 solenoids. It was originally planned to be completely 3D printed however, limitations of the available printers required the top flange and bottom base be added after printing. We currently have four in total, two for the torpedo launcher, two for the marker dropper system, and two extra in case they are needed for next year's competition. The solenoid we've decided on is the same one that was used two years ago by the 2012-2013 team (Clippard MME-41NES-D012). It proved to be reliable for their competition and with its simple mounting procedures it will make for an easy addition anywhere we choose to place it.

Brush Guard

Since many of the systems on the submarine have been rearranged/redesigned, the center of mass and center of buoyancy of the submarine have changed as well. To offset the differences in weighting, a brush guard was designed and implemented. While it's main purpose is to promote fluid movement of the submarine underwater, it also doubles as a protective guard in case the submarine has any collisions, as well as a stand for the submarine when set on a flat surface. With the sub frame redesign, the bottom thruster interfered with the submarine's base. To counteract this the bottom of the guard was designed to protrude down to form leg-like stands which allow for the submarine to be easily and safely be placed on flat surfaces. The brush guard was also bent around the side-mounted thrusters, which allowed for the fluid flow to these sections to be undisturbed during both dive and surfacing operations while adding limited drag to the forward, back, and strafe movements.

Circuit Rack



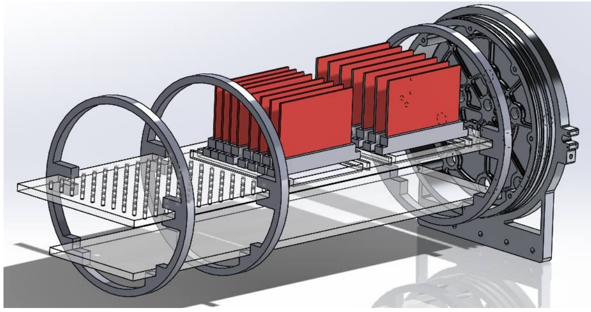
The circuit racks purpose is to house the electronics that control both mechanical and vision components for the autonomous robotic sub. The electronics rack more specifically holds motor controllers that control 6 independent Seabotix thrusters and a small computer that receives and processes vision data and controls various PCB boards within the circuits rack. The circuit rack also includes various PCB boards that our custom made to manage sensors, control pneumatic solenoids and give feedback on various systems on the sub. With this many parts and wires comes a mechanical challenge to keep the electronics waterproof, organized for accessibility and ascetically pleasing to practice commercial techniques.

The first step to meet our goals was to understand what practices are out there to manage electronics systems. Previous teams have made their electronics racks completely removable, have multiple degrees of axis or have stayed with the traditional static mounting system.

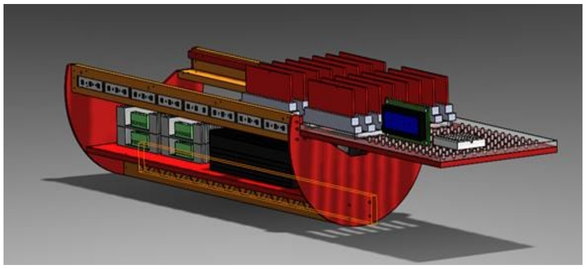
The idea was to separate the entire electronics rack from the hull when doing maintenance on it. This allows for easy access to electrical components and therefore makes maintenance on our submarine a lot easier.

The design of the circuit rack went through different stages. Each of these stages allowed the team to weigh the pros and cons so we could fix the parts that we discussed would be a problem.

The first orientation gave us a general idea of what we wanted this fixture to be. In the render below, you can see how there are no electronics placed on the rack yet. This render's goal was to see if the drawer concept would be possible given our restricted size requirements.



After concluding that the electronics rack would indeed fit within our size restrictions the addition of electronics took place to see how much space we had for each of the components.



As the semester came to an end and the CAD reached its final steps, there was a whole other mountain to climb over. As of November 30th the parts for this fixture were ordered and arrived on December 9th, 2014. On January 5th the team planned to start machining all necessary parts and assemble it all before March. Once completed with the rack the team will finish the presentation for it and complete an animation to present to judges. This fixture still has a lot more to go and will require many more days, and nights of work.