Montana State University Autonomous Underwater Vehicle Design and Implementation

Jacob Jenks, Alison Combs, Levi Hartman, Colton Crist, D.J. Landwehrle, Rick Chen, Jacob Smith, Reda Alqarqoosh, Justin Marquart, Kirby Overman

I. Introduction

The RoboSub club at Montana State University is a new entity on campus. Previously the RoboSub team at MSU was nothing more than a capstone project for seniors in mechanical and electrical engineering. Last year we decided that the team needed better integration of students studying other disciplines as well as students other than graduating seniors so that the knowledge learned during one year of the competition could be applied to the next. Last year at the competition we were fortunate enough to win the *Pay it Forward* award, which we were able to put to use creating the club and recruiting students to join.

Since then we have been able to recruit engineering students of all experience levels, from freshman to grad students, and the team has benefited greatly for it. Next year we hope to recruit a business team to help seek out more sponsorship opportunities.

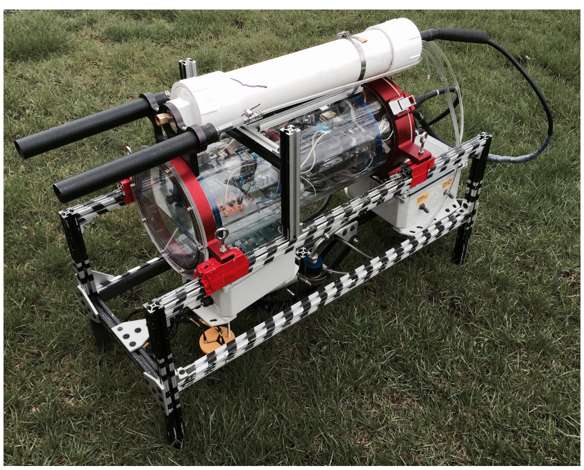

Fig. 1: Montana State University's 2016 Autonomous Submarine

II. Design Strategy

This year we set out to design and build an entirely new autonomous submarine capable of undertaking every task at the RoboSub competition. In previous years we built a sub that was only capable of attempting a small subset of the tasks. We knew that building a sub capable of attempting every task at the competition in a single year would be a difficult challenge, but we wanted to create a platform flexible enough to be reused in subsequent years.

We also decided to pioneer a new approach for handling the difficulty of accurate object detection at the competition. Our approach makes use of recent advances in the field of machine learning that allow for very powerful real-time object detection using a neural network. However, although use of this technique potentially promises very good results, it comes at the cost of additional complexity. Running the object detector in real time requires dedicated hardware to do so. This hardware comes in the form of a Graphics Processing Unit (GPU), which is a piece of computer hardware designed to perform many computations in parallel. The parallel processing capabilities of a GPU are what enable the performance enhancements that make this approach possible.

Deciding to include a GPU on our robot was a difficult decision that came late in the design process. The GPU is our largest single electrical component on the sub, and a custom mounting for it had to be designed to hold it in place inside the electronics capsule. The GPU also requires as much power as the rest of the computer components combined, meaning that it needs its own power supply. Despite all of these special concessions necessary for it to work, we decided that the potential power and reliability provided by this object detector would be worth the effort. This approach is described in more detail in the following design section.

III. Mechanical Design

A. Frame

The best choice of frame design was dependent on a variety of factors. To adjust for the possible changes in competition requirements the frame must have flexible placements for accessories such as the dropper, mechanical arm and batteries. The frame can also enhance the subs stability by placing the centroid at a desired location. It also supplies an area for adjustable weights, or lighten the overall weight with the proper materials. The frame must also be simple to assemble as well as strong.

The best option was to build a frame around a waterproof electronics capsule. The design includes a clear capsule which holds the electronics, and an external cage-like frame which holds the thrusters and battery cases. It consists of a series of T-Slot extruded aluminum bars,with a simple connection the attachments can be easily placed and moved. Advantages of this design are its stability, usability and ease of assembly. Disadvantages of this design are an unnecessary amount of material, and weight.

To fix some of the original design problems, more thought was placed in simplifying the frame and generalizing where

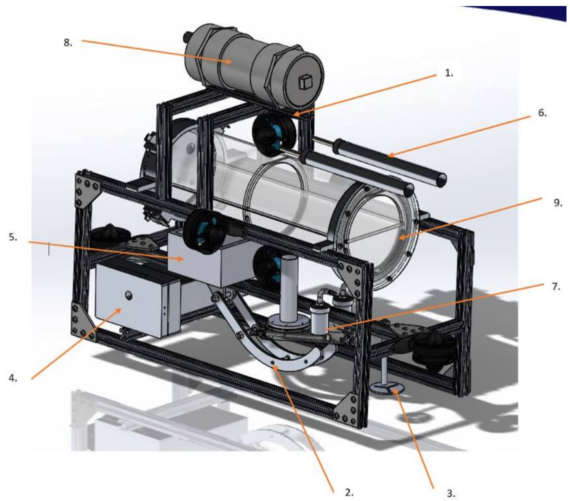

Fig. 2: Sub features: 1) 6 thrusters, 2) Mechanical arm, 3) Hydrophone array, 4) Waterproof battery capsule, 5) Waterproof hydrophone case, 6) Torpedo launcher, 7) Pill dropper, 8) Pneumatic system, 9) Waterproof capsule, 10) Waterproof camera casing

the components will be placed. Much of the unnecessary material can be removed which will lessen the weight and cost. This would also heighten the centroid without compromising its stability. This is the best design choice through the decision matrix. A new radial design is being considered which would consist of a frame symmetrically around the center electronics tube. This design would keep the centroid of the submarine in the middle of the capsule, this would grant the submarine more twisting maneuverability, but would cause instability in turbulent water. Some advantages would be simple assembly, light weight, and less materials needed than the original design. It would also require a complete frame redesign and new components to machine.

B. Capsule

The electronics capsule is made from CNCed aluminum plates and arcylic plastic tubing and sheets. It is essential that this capsule always be reliable as to not compromise the onboard electronics which resemble a large investment. To do this the front plate is sealed with 8 screws that hole an o-ring between the arcylic and aluminum plates. A high grade epoxy is used to create a seal between the arcylic tube and aluminum plates. The end plate required a bit more design as the capsule must be opened and closed frequently. A rubber gasket is used to create a waterproof seal thats held in place with four clamps. It is necessary that there are electrical pentrations that go through the endcap to allow for access to and from the onboard electronics. Wetconnects were found to be best suited for this and were purchased through SeaCon.

C. Pneumatics

Previous Montana State teams created iterations of subsystems to put on the submarine, but didnt incorporate them into the final sub design. This years team improved and implemented the designs for a marker dropper, torpedo launcher, and a pneumatics system for control. One of the initial components of pneumatic system is an air compressor, which will take ambient air and compress to a required pressure. In our design, we will simply use a refillable bottle of compressed CO2, as opposed to an air compressor. The compressed CO2 source is fed into a pressure regulator, where the pressure of CO2 can easily be increased or decreased. The compressed CO2 is sent to mechanical components using a solenoid. A solenoid is an electromechanical device that is used to control the flow of a fluid [3]. They are controlled by electrical current, which energizes the coil. This causes a barrier between the two conduits to move, allowing the fluid to flow to the opposite side. Similarly, when the current is no longer sent to the coil, the barrier will once again obstruct the flow between the two channels. The electrical current is commonly sent using a command from an Arduino microcontroller. After the solenoid directs the compressed CO2, it finally reaches the acting cylinder. This is where the CO2 is converted to linear motion. The CO2 is sent into a small chamber and has such a high pressure that it displaces the piston therefore increasing the volume of CO2. Depending on which type of cylinder is used, a spring or an additional CO2 inlet may be used to reset the cylinder to its original position. A single acting cylinder uses just one CO2 inlet and a spring to reset the cylinder. This causes loss of energy, as the CO2 force has to overcome the spring force in order to move. A double acting cylinder uses an additional CO2 inlet on the opposite side of the cylinder, which requires an additional solenoid valve. The marker dropper utilizes steel cylinders with weak magnets attached which were the payloads themselves. This was placed in a bore connected to the pneumatics tube that purges air into the bore, overcoming the magnetic force and freeing the payload to fall to the bottom of the pool. The torpedo launching system stores a torpedo inside of a tube. To fire, a solenoid is opened and causes a back pressure and propels the torpedo forward. Torpedoes are accurate for about 3 feet before floating to the surface, which makes them easy to retrieve.

D. Arm

As previous years submarines did not incorporate a mechanical arm, this years team sought out to build one that could complete general tasks possibly implemented in the competition. For sake of simplicity and reliability, it was designed with a single degree of freedom movement. This allowed the arm to retrieve and release objects below the submarine with confidence.

Utilizing simple geometry, anodized 6061 aluminum, and stainless steel hardware, the arm has a jaw gap of 7.5 inches (adjustable) and can lift up to a 35 pound load, out of the water. The muscle in the system is a pneumatic linear actuator which runs on 100 psi of pressure, regulated and controlled by the pneumatics capsule.

This system was designed under the principle of simplicity, whether it be manufacturing or controlling the arm. As the arm can fail in hardware or in control systems, it was critical to maintain a straightforward, reliable design with as few moving parts as possible. In addition, there was sparse real estate on the sub. This forced the height of the arm to be very sleek, but a wide jaw gap was important to successful retrievals. As a result, the arm was build to be modular, allowing the overall length to be widened or shortened and optimizing maximum jaw gap.

Objects under the sub are first identified by the bottom facing camera. Once it observes the object and begins to track it, the sub aligns itself above the object. As it then descends upon the object and reaches a specified height above the object, air is released from the actuator, allowing the arm to open. Last, the sub descends into the range of the object and repressurizes the actuator to close the jaws upon it. As a failsafe, the camera continues to track the object, ensuring it does not move with respect to the sub as it is transported to its destination.

IV. Electrical Design

A. Hydrophones

Three Teledyne TC-4013 hydrophones are mounted on an external hydrophone external case. These three hydrophones will be used to calculate the direction and distance to the acoustic pinger. The two front hydrophones are used to find the direction of the pinger and when the two hydrophones H1 and H2 are in line with each other, which are orthogonal and coplanar to the pinger, the signal received is in phase with a zero reading. In this case the sub is facing the pinger as shown in Figure 3. The third hydrophone, which is located in the back, is used to calculate the distance between the submarine and the pinger. A wave carrying an intensity of 162 dB hits the hydrophones with a bare sine wave with a 2 mVpp.

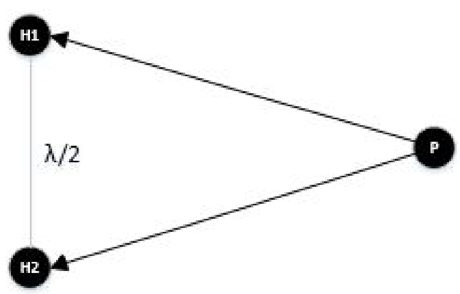

Fig. 3: H1 and H2 are in line with each other which are orthogonal and coplanar to the pinger

The distance between each hydrophone is equal to one half wavelength. Let wavelength λ be the distance between identical points in the sine wave of a waveform signal. Let the frequency of a sine wave representing f, the wavelength be λ, and the speed of sound in water be c. λ can be expressed as $\lambda = \frac{c}{f}$. The distance between hydrophone one H1 and two H2 would be one half wavelength which is the maximum distance to be in phase for a half cycle of a sine wave.

B. Pinger Locator

The design for the pinger locator system is comprised of three parts: amplification, phase measurement and calibration.

The amplification stages were required to amplify the signals from the hydrophones into readable signals that could register with logic level voltages. This stage was comprised of two LF411s in a non-inverting configuration. The gain that was required from this stage was 1000 V/V. This allowed the small signals from the hydrophones to be amplified to the 5 V rails powering the operational amplifiers. Thus, the sine waves were converted into square waves which helped filter out residual noise which could get in the way of measuring phase.

The next stage, a CD4046 was implemented to measure the phase. This device outputs a square wave that has a duty cycle directly proportional to the phase between the two signals. From there it was a simple matter to obtain a DC voltage by means of a low pass filter.

The last stage comprised of calibrating the hydrophone array to the different distances and turn angles. This was accomplished by measuring the voltage level on the output of the circuit at different turn angles and distances. This data was used to generate equations in terms of the DC voltages.

C. Computer

The motherboard is the ASRock B85M-ITX Mini Motherboard which will provide different ports and slots so all the on board components can be connected and installed. Once powered on by the voltage controller, the motherboard will provide power for other components such as cameras and Arduino. The Arduino MEGA ATmega128 that we will be using it for the submarine. They will be used for collecting data from the pressure sensor and hydrophone system as well as controlling the thrusters and pneumatic subsystems. A GEDC-6E IMU was selected for localization of the sub. It is able to directly communicate with motherboard through a USB2.0 cable. All programs being run on the submarine can directly communicate with this IMU through the serial port. shows the two ELP wide angle cameras are the main visual sensors on the submarine. There are two different sets of cameras placed on the submarine to assist with the object recognition program that will be running. The cameras support 1080p video processing at 30 frames per second. The field of view on the camera is also adjustable. Both cameras interfacing with motherboard with USB cables. The downward facing camera was placed in a separate housing nearby the linear actuator hook, purpose for this camera is to observe the operating environment for the machine arm. Front facing camera is the eye for the sub, it constantly send visual data back to motherboard for further image processing. A 3D printed bracket was printed so the camera can be mounted on to the submarine. Lastly the submarine houses an Nvidia GTX 1080 graphics card that provides additional computing power for the object detector

V. Software Design

This year's software was written to make use of Robot Operating System (ROS) running on top of Ubuntu 14.0.4.

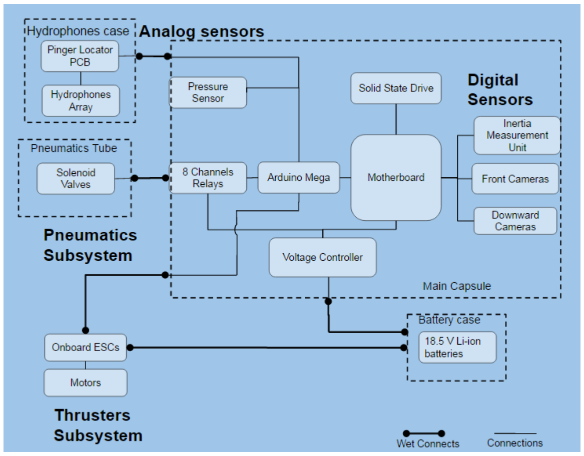

Fig. 4: Electrical Design

ROS is a collection of tools to make developing software for robotics applications simpler and faster. We made the decision to switch from last year's pure C++ custom made application to python in order to make it easier for less experienced computer science volunteers to contribute.

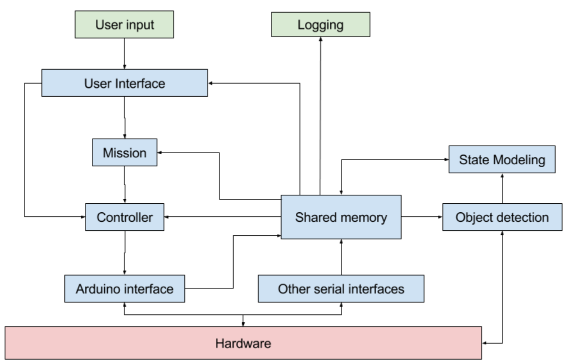

Fig. 5: Software Design

The sub's mission planning module uses a hierarchical priority queue to perform tasks in the order in which they become available. This way less important tasks are not prevented from executing if a higher priority task is unavailable, such as when the location of a near target is not known, but the location of a far target is.

Modeling the state of the sub is performed via an extended Kalman filter fusing data from the IMU, the depth sensor, and the downward stereo cameras. The downward stereo cameras provide odometry information using an library called libviso2, which is able to construct a 3D point cloud of the bottom of the pool using the disparity between the images. Tracking the location of the competition objects relative to the sub is performed via a ROS package that performs probabilistic modeling with sensor fusion called hector_object_tracker[1].

A. Object Detection

Our object detector is an exciting innovation for our team, and we believe it is a fresh approach to the problem of visually identifying competition objects for the RoboSub competition as a whole. Correctly identifying the location of a target object in an image is a difficult task, especially in the context of the RoboSub competition. The RoboSub competition is held outdoors, and is therefore subject to constantly changing weather and lighting conditions. As we found at last year's competition, this can drastically change the appearance of the competition objects, and can easily confuse simplistic object detection techniques, such as the color thresholding and edge detection approach employed by many teams at the competition. Steps can be taken to somewhat enhance the robustness of this technique, such as use of an ensemble of these detectors and careful parameter tuning, but it is fundamentally limited by the fact that the exact color of the target object at an arbitrary competition time must be known, and that color must be easily separable from background elements. This is especially difficult with competition objects such as the green buoy, which is a very similar color to the rest of the pool when the lighting conditions are right. Truly robust object detection requires a more nuanced approach that is capable of examining the geometric properties of objects, as well as their color.

Our object detector makes use of a brand new algorithm in machine learning known as Faster R-CNN [2]. Faster R-CNN uses a convolutional neural network to draw accurate and precise bounding boxes around target objects. A convolutional neural network is a biologically inspired approach that mimics the structure of the visual cortex in animals, and it has proven to be a powerful technique for image classification. It is trained by feeding it labeled images of the desired output, which in this case is hand-labeled bounding boxes around competition objects in images. It uses these training images to learn the desired output, which it can then extrapolate to previously unseen images. An example of the results produced by our object detector is shown in Figure 6.

A neural network uses layers of simulated ”neurons” that take input from the previous layer, runs a function on that input, and outputs it to the next layer. The first layer in a neural network is typically the input and the final layer of the network is the output. In our case the input is an image, and the output is the coordinates of a bounding box in the image, and a label telling us which competition object is contained in the bounding box. A neural network is a good choice for this application because they primarily learn to identify the geometric properties of a target object to be able to recognize them. This should, in theory, make our object detector much more resilient to the constantly changing conditions of the competition pool.

Running large neural networks can be very computationally taxing. Fortunately, Faster R-CNN is able to harness the parallel computing power of a GPU, which enables it to run in real time. In practice we are able to process $\sim$7 frames per second, which is more than sufficient for a slow-moving vehicle.

VI. Experimental Results

A. Pneumatics

Both dry and pool testing was required for all systems. For each of the marker droppers and torpedo the solenoids only

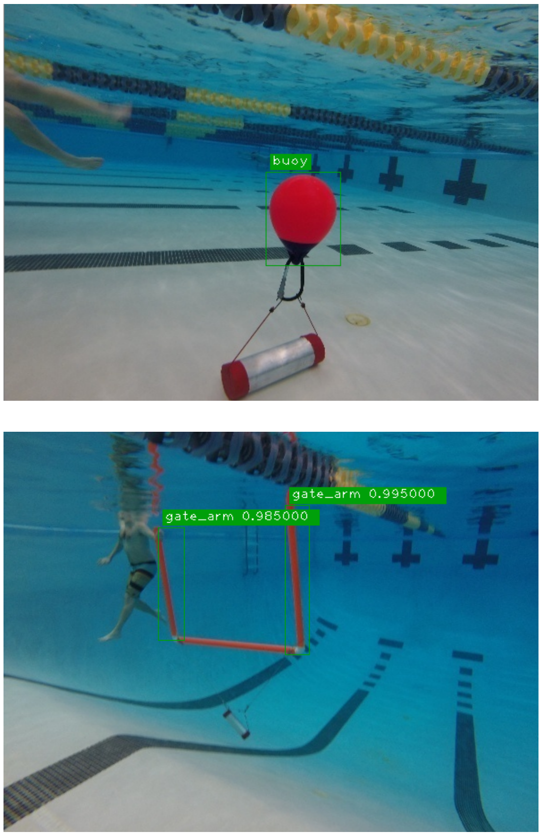

Fig. 6: Object detection using Faster R-CNN. The number next to the class label indicates a confidence score.

need to be opened once to fire. The pneumatic actuator was a new addition to the system and for the first test, only the actuator was used. The actuator required two solenoids, one to force air into the cylinder to push the piston back, and one to evacuate the chamber and allow the spring to assist the piston to the open position. Once this test was proven to be successful, it was tested on the frame using the full arm. This test proved that the arm was able to pick up the object used in the competition. The final test included integrating the system into its final configuration, with the arm being controlled and powered through the wet connect, into the Arduino inside the computer capsule. This test was conducted at a pool test and included testing the marker dropper and torpedoes. All subsystems were found to work, but 20 seconds were needed between each opening to re-pressurize the system. The main recommendation for improvement would be refining the pneumatics capsule to decrease unnecessary buoyancy and weight.

B. *Object Detection*

As mentioned in the description of our object detector, we must train our neural network in order for it to work. A neural network is trained by feeding it input where the correct output is known. The network uses the difference between its own output and the correct output to correct itself, and by doing so slowly learns the desired pattern. In order to get good results with a neural network it must have many examples to learn from, and it must review them many times. Practically speaking, this means that we had to build a training dataset by manually draw bounding boxes around competition objects in hundreds of images. This labeling was done using a simple custom built program.

For our proof-of-concept experimental approach we decided to take pictures of just a buoy and a gate in the pool at the gym at MSU. For the actual competition we will need to train on images of each and every competition object, but in order to test the feasibility of this approach we decided to just use these two competition objects. In total we took 650 pictures, of these 528 are of the buoy, and 122 are of the gate. We ended up with more pictures of the buoy than the gate simply because the gate is larger and more awkward to position to take pictures. Having a disparity in the number of training images for each object is actually a good thing for testing, since we would like to manually label as few images as possible while still getting good results. Comparing the performance of the neural network on each object can give us a rough idea of the minimum number of images required to get good performance.

Training on this dataset took approximately 14 hours. However, the results were well worth it. Our object detector was able to draw bounding boxes whose area was 99% accurate in the case of buoy detection, and 89% accurate in the case of gate detection. This indicates that around 500 labeled images per competition object is a good target to aim for in order to achieve good performance.

The results we got were very encouraging, but it should be noted that our work on our object detector will not be complete until we get to the competition. The images taken in our pool have a much different appearance than those from the competition pool in San Diego, and our object detector is unable to account for this difference without help. As a result we will need to collect a new dataset of images of competition objects once we arrive. We hope we will be able to collect images manifesting all of the various lighting conditions present at the competition, which will allow our object detector to be as robust and well-prepared as possible for anything the competition can throw at it.

VII. Acknowledgments

We would like to extend our thanks to NAVSEA as a whole for providing us with the funding necessary to undertake this project. We would also like to specifically thank Mike Kapus from NAVSEA for advocating for us, and for advising us on this project. We are very thankful to have had this opportunity, and it has been a great learning experience for all of us.

References

[1] J. Meyer, P. Schnitzspan, S. Kohlbrecher, K. Petersen, O. Schwahn, M. Andriluka, U. Klingauf, S. Roth, B. Schiele, and O. von Stryk, "A semantic world model for urban search and rescue based on heterogeneous sensors," in *RoboCup 2010: Robot Soccer World Cup XIV*, ser. Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence, J. Ruiz-del Solar, E. Chown, and P. G. Ploeger, Eds., 2011, pp. 180 – 193.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.