Montana State University Robosub Design

INTRODUCTION

The RoboSub team at Montana State University is a capstone project for seniors in mechanical and electrical engineering along with a team of volunteers. The capstone team is responsible for all of our hardware development while the volunteers, members of MSU's RoboCats team, is responsible for the software development and assisting with testing and machining.

ELECTRICAL SYSTEMS

A. General summary

Bobcat receives its power from four 18.5V 10,000mAh LiPo batteries. These batteries are hot-swappable and provide approximately 47 minutes of run time with thrusters at peak efficiency (1 lbf) and the vision recognition system running.

These batteries then source power to Bobcat's two main electrical systems. Its first is housed inside the main electronics capsule and its primary draw is its ASRock B85M-TX motherboard and GeForce GTX 980 GPU – the powerhouses behind its vision recognition system. The other large electrical subsystem is Bobcat's eight T200 thrusters (it's key to enhanced mobility).

B. Battery backplanes:

As mentioned before, Bobcat has two battery capsules (two batteries per pack). The first contains Battery Backplane One (BB1) which provides power to the main electronics. Battery capsule two with BB2 sources power to the eight thrusters. The backplanes are identical in that they both are hot-swappable, have star grounds, and utilize an Arduino Mega 2560 with two tip 120 mosfets to trip two heavy duty Bosch relays. However, BB1 outputs 37.5 volts at up to 50 amps (necessary for the main electronics DCDC custom computer power supply). BB2 outputs 18.5V at up to 50 amps for the eight thrusters (necessary for the ESC's input voltage). Figure AA shows how the backplanes were physically assembled.

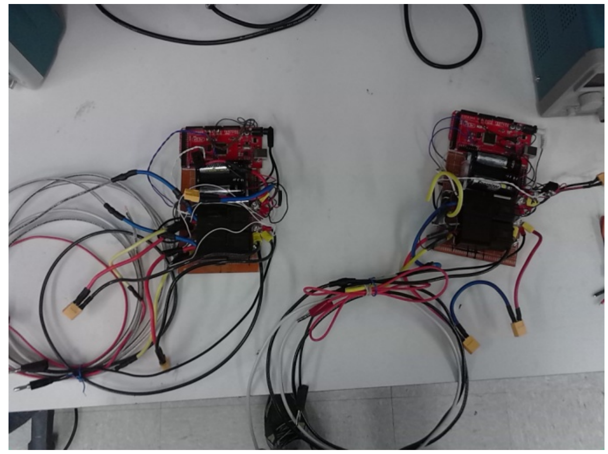

Fig. AA. BB2 on the left, BB1 on the right. Both utilized 10AWG wire to carry their high currents.

C. Main Electronics:

The main electronics capsule is pictured below in Figure XX.

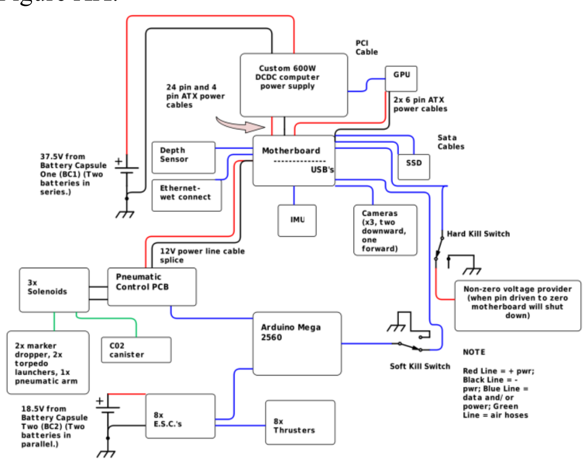

Fig. XX provides simplified details regarding the various connections running to and within the main electronics capsule.

Two sets of power wires are fed into the main capsule (more details on wire routing of the end-cap can be seen in Figure YY and Table XX). Set one runs immediately into a 600W custom DCDC computer power supply from Powerstream. This powers the Motherboard and the GPU. The motherboard is then the hub for distributing control and power to the remainder of Bobcat.

Inside the capsule the wet-connect Ethernet runs to the motherboard for live debugging. Additionally, the soft and hard kill switches run to the motherboard and Arduino. (Soft shuts off thrusters and pneumatics, hard powers down the motherboard.)

Bobcat's eyes are two Microsoft Life-cam Cinema cameras (one downward and one forward facing). An SSD stores data from the cameras such that Bobcat can "remember" items and successfully identify them, e.g. odd round object with pink on top equals "a donut". Also, this system replaced our hydrophone system for locating pingers due to its advanced recognition abilities.

Additionally, an IMU and depth sensor allow Bobcat to determine its location and speed in space.

Lastly, the control of the pneumatics and thrusters were dictated by an Arduino Mega 2560. Control wires from the Arduino are run to a pneumatic control PCB. This then transmits commands via wires routed via through-connects on Bobcat's end-cap to the pneumatics case where the solenoids are housed. The thrusters are controlled with the Arduino and receive power from BB2's wire-harness. (This harness consists of an 18 pin ATX connector with BB2's four connection wires on one side and the eight pairs of +/- power wires for the ESC's on the other.

D. Thrusters:

T200 thrusters were chosen for their proven power and reliability. Eight thrusters were used over the previous six for greater maneuverability and less ESC strain on the robot. Also, these ESC's (Afrojack 30A Race Spec Mini ESCs w/BEC) were upgraded to handle more current than the previous Afrojack's and the built in Blue-Robotics Thruster ESC's. These older ESCs had a tendency to burn out. With the newer Afrojack's, they guard against burnout by being beefy enough (30A) to handle a greater current than the max draw of the T200's (rated up to 25A). The only downside of these newer ESC's was the required additional purchase of a USB programmer to successfully flash them with Blue robotics forward/ reverse firmware. However, once flashed they performed successfully!

.

E. Pneumatics:

Pneumatic solenoids are responsible for the torpedo launchers, marker droppers, and pneumatic arm. The solenoids are housed within the pneumatics case in the center bottom of Bobcat. Air hoses and power wires are run from this container. (The C02 canister is mounted outside of the Pneumatic case for ease of access.)

The control PCB is housed inside the main electronics capsule and utilizes a series of Tip 120 transistors to trigger the solenoids. (The solenoids positive line is provided from the DCDC computer power supply.) For an illustration of this, please see Figure WW.

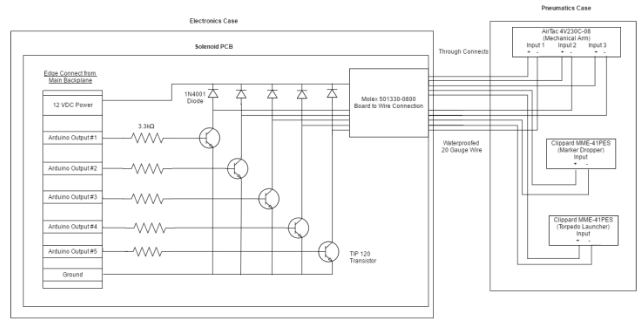

Fig. WW. Above contains the wiring diagram and necessary parts to build the pneumatics control PCB.

F. Outer electronics and wiring routings:

Shown below in Figure YY is the routing diagram for Bobcat's end-cap. All connections shown (barring CD, CF, D, K1, K2, and EN) are made with 6mm through-connects. CF and CD are clear half-spheres that allow the cameras (housed inside) to see out into the water. K1 and K2 are water proof switches. D is the depth sensor and EN is the wet connect for the Ethernet.

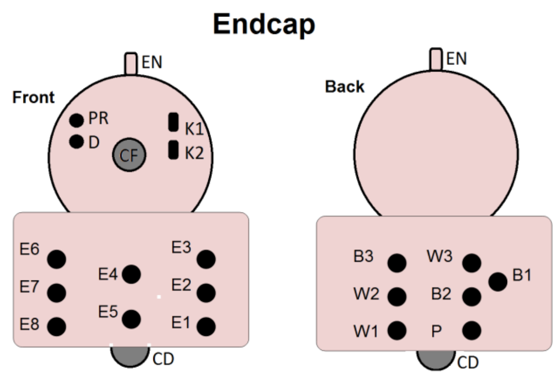

Fig. YY. Visual companion to Table XX.

Table XX. Below are the descriptions of the end-cap connections and their respective locations. (Battery capsule = BC)

Route name	**Abbr.**	**Route name**	**Abbr.**
Neg./ Black wire from BC1	**B3**	ESC 8	**E8**
Neg./ Black wire from BC2	**B2**	ESC 7	**E7**
Neg./ Black wire from BC2	**B1**	ESC 6	**E6**
Positive/ White wire from BC1	**W3**	ESC 5	**E5**
Positive/ White wire from BC2	**W2**	ESC 4	**E4**
Positive/ White wire from BC2	**W1**	ESC 3	**E3**
Pneumatics control wires	**P**	ESC 2	**E2**

Pressure release valve	**PR**	ESC 1	**E1**
Depth sensor	**D**	Ethernet	**EN**
Camera front	**CF**	Kill switch soft	**K1**
Camera down	**CD**	Kill switch hard	**K2**

Mechanical

A. *Frame*

This year we had a full overhaul in our mechanical design compared to previous iterations of this MSU capstone. The full redesign was by recommendation of the team's primary sponsor, NAVSEA. This year, primary capsule slides off of the main electronics rack for ease of access to the sub's main electronics. Because of this change we no longer have to unplug and re-plug all of our interior electronics which, in previous years, put strain on our wiring and may have even contributed to the damaging of our Ethernet coil pairs. In addition, last year we sealed the open end of our capsule with a single rubber gasket. This year we introduced a double O-ring seal and 3D printed latches.

Our frame is aluminum cut with 10 inch cross sections and holes drilled nearly every two inches in each cross section for modularity. This way we can attach our thrusters around our center of balance yet still make changes based on power differences and design changes. In addition, we can move our torpedoes around with our 3D printed brackets. On the fore of our sub, we have included an aluminum cross bar for our hand. The aft section of our frame also unlatches and open up for easy access to our two battery packs.

Our battery packs feature a 3D printed rack internally to hold our Arduino and two batteries (per pack).

Both the battery packs and the main capsule's bulkhead also include blue robotics through connects, a new addition to our team's sub this year in which we exclusively used Seacon wet connects. This year we still use one 8-pin wet connect for our Ethernet connection.

B. *Pneumatics*

On the belly of the submarine we have our pneumatics with a water tight gasket seal. Our pneumatics system is used to power our torpedoes, marker droppers, arm, and hand.

C. *Mechanical Arm*

As previous years' submarines did not incorporate a mechanical arm, this year's team included one that was constructed by another senior capstone team at Montana State University. 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 built 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.

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. 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.

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 □7 frames per second, which is more than sufficient for a slow-moving vehicle.

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 Alex Read, Aaron Darnton, and 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. [1] J. Meyer, P. Schnitzspan, S. Kohlbrecher, K. Petersen, O. Schwahn, M. Andriluka, U. Klingauf, S. Roth, B. Schiele, and O. von Stryk, "A se- mantic 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. [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.