

Design and Implementation Report of Hammerhead: AUV NIT Rourkela

Abstract – Autonomous Underwater Vehicle team of National Institute of Technology Rourkela (AUV NIT Rourkela), also known as Team Tiburon is entering it's AUV named Hammerhead into the 2019 AUVERSI Foundation Robosub Competition for the first time. In this report, we have explained the design of the AUV frame and its advantages in addition to the electronic circuit design and the software we use.

I. Introduction

Tiburon is the autonomous underwater vehicle team of National Institute of Technology Rourkela, India. The team, established in December 2014 consists of students who work on Mechanical Design, Software, Image Processing and Electronic Circuit Designing.

We have designed three vehicles so far. In every vehicle, there were drastic improvements in term of mechanical design, electronic circuit design and software design. Finally, we achieved six degrees of freedom, dropping and grabbing of markers, and shooting of torpedo.

II. Competition Strategy

We will start our vehicle with a toss of a coin. To envision this, we have developed an external switch using a reed switch, which will work as if the switch is on the vehicle will start perpendicular to the dock. Otherwise, it will start parallel to the dock. We have targeted to cross through 40% of the area. For gate detection, simple image processing functions like Sobel function have been employed in both x-y directions. We have not developed grabbing feature so, after crossing gate, we will not pick any markers. To drop markers, we will carry two markers our own. For moving forward we must detect the path. We have done that using image processing and will move to slay the vampire. To detect vampires, we have used machine learning. During testing, we will check which vampire is detected with the most accuracy. Next, we will call that vampire during the final run. Along with this, we will touch the face of that vampire and will also touch the single buoy. From slay vampire there will be a path pointing to drop garlic bins. In drop garlic, we have decided to drop marker into the already open bin. Then with the help of Pinger, we will find stake through heart task and using the front camera we will align with heart and will fire torpedo through it. The last task is exposed to sunlight. We will have one marker left with our AUV, so we plan on dropping that marker and grabbing the Dracula to surface inside the octagon.

III. Vehicle Design

A. Mechanical –

- a. **Shape and Size** – The shape of the vehicle is generally cylindrical. The cylindrical hull is housed at the bottom by aluminium trapezoidal frame so that the upper half of the cylinder is easily accessible. The frame is designed in such a way that any additional feature can be added to the vehicle keeping the size and shape of the bot compact. The different views of the bot are as follows-

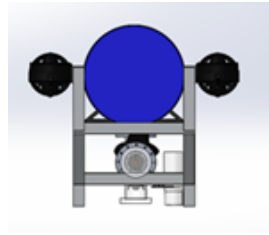


Figure 1: Front View

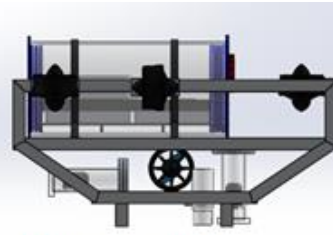


Figure 2: Side View



Figure 3: Isometric View

The dimensions of the bot are as follows -

Length=69.5 cm, Breadth=47.5 cm, Height=40 cm

Hull length=42 cm, Hull diameter=20 cm

Frame length=69.5 cm, Frame width=27.5 cm, Frame height=31.5 cm

Weight = 21.3Kg in air

- b. **Type of Buoyancy** – The vehicle is designed to be 95% submersible while putting in the water without any external power. This allows the vehicle to resurface if there is a condition of breakdown. The centre of buoyancy lies above the centre of gravity along the vertical axis to eliminate toppling.
- c. **Basic vehicle frame design and material selection criteria** – The frame is designed to be compact and modular to reduce drag force to a minimum and provide efficient hydrodynamics.
- d. **Design of Hull** – The hull is designed based on the easy availability of the materials in the market, namely acrylic cylinder of the appropriate dimensions. Both the ends can be opened easily to facilitate easy access of the electronics inside. The hull is divided into two layers – one for electronics and circuitry and the other for battery enclosure. The material of the hull is chosen in such a way that it does not crack at a depth of about 10m. This assumption is verified by carrying out a simulation in ANSYS Static Structural by applying hydrostatic pressure on the hull due at a depth of 10m.

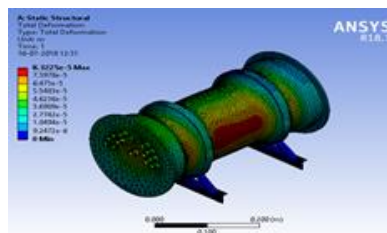


Figure 4: Hull at 10m depth in ANSYS

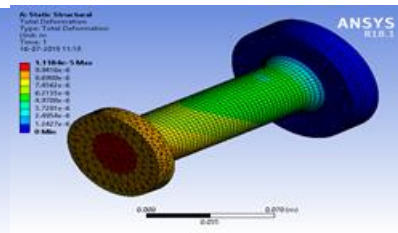


Figure 5: Camera Hull at 10m depth

- e. **Camera Hull Design** – The camera hull is also made of the same material (Acrylic) as that of the main hull. So similar simulations were performed on the camera hull in ANSYS as done in case of the main hull.

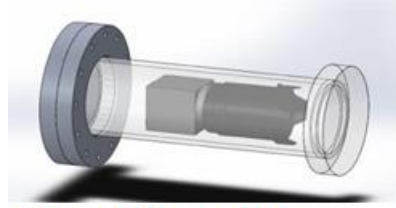


Figure 6: Camera Hull

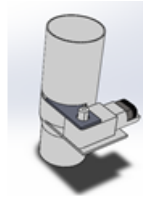


Figure 7: Dropper

- f. Drag calculation and Drag coefficient:** - In order to simulate the motion of the vehicle in water (with an initial velocity of about 1m/s), in ANSYS Fluent, a flow domain(material->water) is considered with the vehicle inside it. An inlet velocity is provided at the water inlet and reset all sides of the flow domain are assigned as pressure outlet with gauge pressure=0. The drag force generated is 37.1N and the drag coefficient is 61.35.
- g. Object Dropper** – A very simple mechanism has been employed to drop the required objects from the vehicle. In our mechanism, a BO motor is mounted to a PVC pipe. When the motor rotates, the obstruction causing the object to stay is removed and hence the object falls.
- h. Torpedo** – Torpedo is in the shape of a hollow cylinder of 10mm diameter and is 3D printed for better accuracy. For launching torpedo, the air tank is to be used to store air at a pressure of about 30Pa. The torpedo is connected to the tank through PU fitting pipe passing through a coil actuating solenoid, which on receiving the command opens for about 10ms thus allowing the air to pass from the tank to the torpedo through the PU fitting pipe.

B. Electronics –

- a. Design of Electronics System** – The electronic subsystem is implemented using two custom made PCBs which are stacked vertically. One circuit board serves as the main control centre of the AUV. It consists of DC-DC converters, Arduino Nano, Servo Controller, Power MOSFETs etc. while the other circuit is used for driving the thrusters. Both the circuits are **electrically isolated** to avoid any damage to the computing units from the high-power rotors. Various voltage nodes have indicator LEDs to make debugging easy. The circuit primarily consists of through-hole components which make their replacement and debugging relatively easier and faster.
- b. Power Budget and type of Battery** – The power infrastructure includes smart monitoring and regulation of power. The Li-Po battery packs are selected as per the power needs of the vehicle keeping the running time of the vehicle in mind. One of the battery pack powers the thrusters and its driving circuit. The other battery powers the CPU (Intel NUC) and its supporting circuit consisting of other controllers for sensor data acquisition and actuators for dropper and torpedo launch systems. Keeping a **run time target of 45 minutes**, the 4-cell Li-Po batteries of **16000 mAh** each were selected. Other factors for selecting Li-Po batteries were their high-power density, simplified packaging, compact yet robust form factor, and less probability of electrolyte leakage.

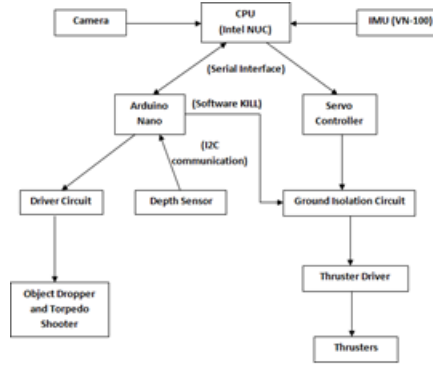


Figure 8: Basic Workflow

Name	Average Power Consumption
CPU (Intel NUC)	30 W
Other micro controllers (Arduino and Servo Controller)	3 W
Thrusters (6 Nos.)	330W

Table 1: Power Consumption Chart

- c. **Power Distribution** – The main control circuit is designed to have 3 different voltage requirements i.e. 5V, 6V and 12V. This distribution of power is done using two DC-DC buck converters for 12V and 6V and an LDO regulator for 5V.
- d. **Kill Switch** – There are two provisions for setting all the thrusters off. A command kill can be initiated to shut down all six thrusters. A manual kill provision, implemented through a reed switch, is also present that disconnects power from the entire circuit and also shuts down all the thrusters.
- e. **Acoustic Navigation** – The Acoustic Navigation system of the AUV uses two **H1C Hydrophones** connected with Raspberry Pi 3B for data acquisition purpose. Currently, the system is in the development phase and algorithm for the localization of the Acoustic Pinger is under-progress. An **Acoustic Pinger (Teledyne Benthos ALP-365 Pinger)** is being used for testing the hydrophones. MATLAB along with liquid DSP and Aquila libraries is being used for algorithm development and debugging.

C. Software –

- a. **Tiburon Commander** – We have developed a GUI application using Qt in C++ called tiburon_commander. The tiburon_commander have a list of processes which are necessary for testing and running of AUV.
 - i. **Control** – In control tab there are two processes one is **roscore** to control(start/stop) ROS master and other one is **thruster_controller** to test specific thruster by changing the speed of a particular thruster.
 - ii. **Hardware** – In this tab, all processes are related to the hardware system. Thruster Driver enables thrusters, Serial Node, Vector Nav gives a reading of IMU (orientation of the vehicle with respect to initial point), Bottom Camera and Front Camera enable the respective camera.

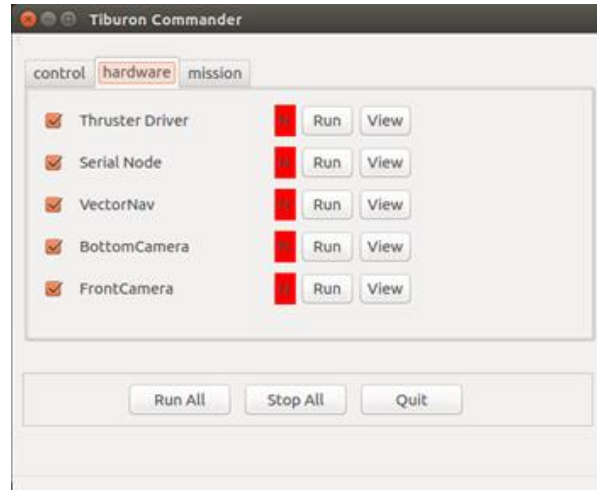


Figure 9: Base Software(Tiburon Commander)

- iii. **Mission** – Mission tab contains VN Synchronizer process which gives a depth of vehicle from the water surface, Controller process is basically controller of thrusters' speed, PID process gives a GUI window to tune PID coefficients
- b. **Vision Processing** – Since we use Intel i7 CPU for all processing, to implement object detection with machine learning on CPU for real-time use was a difficult task. Fortunately, we found a library called OpenVINO. The OpenVINO toolkit enables CNN based deep learning inference on Intel CPU. For normal image processing, we use OpenCV.

IV. Experimental Results

Team has successfully achieved a fully functioning simulator. The simulator is a 3D environment developed in Unity 3D game engine. Before going to test any algorithm, to tune PID coefficients or path planning of AUV into the pool we test it in the simulator. Testing in a simulator helps us to find logical errors along side path planning and tuning PID coefficients. Later, on testing in the pool, we found almost the same results as in the simulator. We performed pool testing regularly whenever the pool was available. In case of any technical issue, we extend our testing time and try to solve it as soon as possible. We have completed the pre-qualification tasks and uploaded a recorded video successfully. Currently, we are testing our training results of object detection in different situations and intensity of light. We found some detection issue in different intensity of light, to overcome which we added that in the dataset and resume training. In this way, we have tested almost all situations and varying intensities of light.

V. Acknowledgement

The Tiburon team would like to thank the Mechanical Department of NIT Rourkela, National Institute of Ocean Technology, NALCO and MATLAB for their constant support and co-operation.

We would like to extend a special token of gratitude to our faculty advisor Prof. Haraprasad Roy who was always willing to support us and believe in us.

Appendix A: Expectations

Subjective Measures			
	Maximum Points	Expected Points	Points Scored
The utility of the team website	50	50	
Technical Merit (from journal paper)	150	150	
Written Style (from journal paper)	50	50	
Capability for Autonomous Behavior (static judging)	100	100	
Creativity in System Design (static judging)	100	100	
Team Uniform (static judging)	10	10	
Team Video	50	50	
Pre-Qualification Video	100	100	
Discretionary points (static judging)	40	40	
Total	650	650	
Performance Measures			
	Maximum Points		
Weight	-	57	
Marker/Torpedo overweight or size by <10%	Minus 500/ marker	-	
Gate: Pass through	100	-	-
Gate: Maintain fixed heading	150	-	-
Gate: Coin Flip	300	300	-
Gate: Pass through 60% section	200	-	-
Gate: Pass through 40% section	400	400	-
Gate: Style	+100 (8x max)	-	-
Collect Pickup: Crucifix, Garlic	400/ object	-	-
Follow the "Path" (2 total)	100 / segment	400	-
Slay Vampires: any, called	300, 600	600	-
Drop Garlic: open, closed	700, 1000 /marker (2 + pickup)	700	-
Drop Garlic: Move Arm	400	-	-
Stake through Heart: Open Oval, Cover Oval, Sm Heart	800, 1000, 1200 / torpedo (max 2)	1200	-
Stake through Heart: Move the lever	400	-	-
Stake through Heart: Bonus – Cover Oval, Sm Heart	500	500	-
Expose to Sunlight: Surface in Area	1000	1000	-
Expose to Sunlight: Surface with object	400 / object	-	-
Expose to Sunlight: Open coffin	400	-	-
Expose to Sunlight: Drop Pickup	200 / object (Crucifix only)	-	-
Random Pinger first task	500	500	-
Random Pinger second task	1500	-	-
Inter-vehicle Communication	1000	-	-
Finish the mission with T minutes (whole + factional)	T*100	-	-

Appendix B: Component Specifications

Component	Vendor	Model/Type	Specs	Cost
Buoyancy Control	-	-	-	-
Frame	-	Aluminum Bars	Lightweight	-
Waterproof Housing	-	Acrylic Hull	Lightweight	-
Waterproof Connectors (LAN)	Thingbits	RJ-45	-	\$10
Waterproof Penetrators	Blue Robotics	M10 for 6mm cable	High-pressure seal	\$4
Thrusters	Blue Robotics	T100	5 pound of thrust	\$119
Motor Control	Blue Robotics	Basic ESC	7-26V max current 30A	-
High-Level Control	Student design	PCB	-	-
Actuators	-	Single Shaft DC motor	6V	-
Propellers	Blue Robotics	m-T100	6-16V, max-power 135W	-
Battery	Gens Tattu	Lipo Battery	14.8V 25C 4S 10000mAh	\$149.99
Convertor	-	LM2576 Buck convertor	Input 3-40V,output 1.5-35V	-
Regulator	Texas instruments	7805	In 7-38V, Out 5V	-
CPU	Intel	Intel i7 5 th Generation	-	-
Internal Comm Network	-	USB	-	-
External Comm Interface	-	Lan	-	-
Programming Language 1	-	C++	-	-
Programming Languages 2	-	Python	-	-
Compass				
Inertial Measurement Unit (IMU)	VectorNav	VN-100	3-axis gyros, accelerometers, magnetic sensor	-
Doppler Velocity Log (DVL)	-	-	-	-
Camera(s)	Edmund optics	Point Grey	USB 3.0	-
Hydrophones	Aquarian Audio	H1C Hydrophone	-	\$139
Manipulator	-	-	-	-
Algorithms: Vision	Deep Learning	Yolo with Intel OpenVINO	-	-
Algorithms: acoustics	-	-	-	-
Algorithms: localization and mapping	-	-	-	-
Algorithms: autonomy	-	-	-	-
Open Source software	ROS	Kinetic Kame	-	-
Open Source library	OpenCV	3.4 version	-	-
Team size	24	-	-	-
HW/SW expertise ratio	¾	-	-	-
Testing time: simulator	30h	-	-	-
Testing time: in-water	200h	-	-	-