Technical Design Report: Tarang

Devendra Kharolia, Kamlesh Kalasariya, Jaskeerat Singh, Prakhar Maheshwari,
Shashank Katiyar, Neelabh Singhania, Shubham Korde, Naveen Chandra R,
Ayush Gupta, Priank Prasad, Inshu Namdev
Faculty Advisor: Indranil Saha, Department of Computer Science, IIT Kanpur

***Abstract*—Tarang is the third Autonomous Underwater Vehicle developed by team AUV-IITK to participate in the 24th Robosub competition organized by AUVSI. The vehicle has significant improvements over the previous AUVs in terms of mechanical design, safety measures, custom PCB boards, SLAM, and vision algorithms. The team has designed a single carbon fiber hull for the vehicle with major changes in the marker dropper, pneumatic system and grabber. The power management system and electrical boards have significant improvements to cater to the specific needs of the competition. Considerable advancement in the software stack have also been made, including real-time object detection using YOLOv3, cascaded PIDs for precise actuation, and Simultaneous Localization and Mapping (SLAM) system for navigation.**

Competition Strategy

The team debuted in the 22nd Robosub competition in 2019 and gained valuable insights. This year the team's primary focus will be on enhancing the reliability of the vehicle in performing tasks. The primary objective will be to perform the most confident tasks before moving on to the other complicated ones. The vehicle is equipped to perform as many as tasks as possible. The weight of the vehicle is significantly reduced, thus giving increased agility and performance. The propulsion system is simplified, and noteworthy improvements were implemented in the controls and vision systems.

The team is confident that the vehicle will perform the gate task with a random starting position and pass through the 'G-Man' side of the gate with an 8X style multiplier. It will then attempt to touch the buoy after following the orange path marker. After completing the buoy task, the vehicle will move towards the marker dropper task. The marker dropper task has improved accuracy thanks to the new mechanical marker dropper design and improvements in object detection algorithms. Thus we are confident that the vehicle will perform the marker dropper task even if it fails to remove the lid from the bins (the vehicle will try to remove the lid for at most three times before moving on to the next task).

The team has been successful in setting up the hydrophone system for acoustic localization over the last year. Due to the pandemic, testing of hydrophones in water or simulation was not possible, but the unit tests for the acoustic system have been performed. The team plan to use the acoustic system with vision data as a backup. The team has experimented with the pneumatic torpedo shooting mechanism in the simulation. It is hoped that the vehicle will shoot the torpedo through the larger hole from the correct side of the prop. Finally, the vehicle will try to find the location of the octagon and rise to the surface within the octagon without attempting to move the bottles.

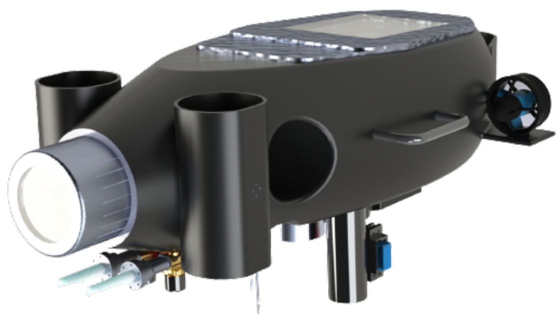


Fig. 1: Tarang

Design Creativity

A. Mechanical Subsystem

Naveen Chandra et al. [1] proposed the design of the previous AUV: Anahita, which was widely acknowledged. However, from the physical testing point of view, it still had some drawbacksthe current AUV tries to address some of these flaws.

Anahita had seven watertight casings to increase the modularity of various systems, but this led to high susceptibility to leakage, often damaging the electronic components during pool testing. In addition, these multiple casing caused much wiring to and fro from the casings. Also, due to the compact design and lack of foreseen planning, some of the sensors/components were not properly accessible.

Some of these design flaws were addressed in Tarang, which will be discussed over the following sub-sections.

Fig. 2: The central hull of Tarang

1) Main Hull: A significant improvement over the previous AUV is the design of a single main hull. The vehicle hull is made of carbon fiber, making it easier to mold into the desired shape and reducing weight. The weight reduction is advantageous as it significantly minimizes the thrust to move or stop the vehicle. In addition, a single hull allows an increase in the simplicity of the design, reducing penetrators' requirements in and out of the hull and improving accessibility. The penetrators were often the primary cause for the water leakages, thus reducing them led to a decrease in the risk of water leakage.

This year, the team installed a pressure sensor in the main hull, which assists in water leakage detection. Once internal components are fixed, the pressure inside the hull is slightly decreased. If a substantial increase in pressure is observed at any instance, kill switch action is triggered. This allows the team to test for water leakages without keeping the vehicle inside any water source.

2) Propulsion System: Due to clever thruster placement, Tarang uses 6 T200 Blue Robotics thruster compared to the previous AUV- Anahita, which used eight thrusters of the same kind. All the 6 degrees of freedom are possible with Tarang, verified using an open-source simulator Gazebo. Motion simulations in the gazebo ensured no excess load (RPM) is put on thrusters due to their lower number, even at comparable speeds as the last vehicle. The reduced number helps bring down the vehicle's cost without compromising on the performance of the vehicle.

Fig. 3: Thruster placement in Tarang

3) Marker Dropper: The previous marker dropper design had some disadvantages like high precision 3-D printing requirements during manufacturing and a tricky reloading procedure. It was also difficult to mount it on the vehicle. This year's marker dropper is much simpler, more accurate, and more reliable. It will have two markers which will fall straight down once they are released. The marker dropper is located near the bottom camera to minimize the errors due to coordinate transformations. It consists of a star-shaped obstructor preventing the marker from falling. Once a trigger signal is received, the servo actuated star-shaped obstructor rotates and allows the markers to fall.

4) Grabber: The grabbing mechanism employs a four-finger mechanism. A single servo actuates the mechanism. The rotation motion provided by the servo is converted into a linear actuation using a slider-crank mechanism. As a result, the grabber achieves a maximum diagonal

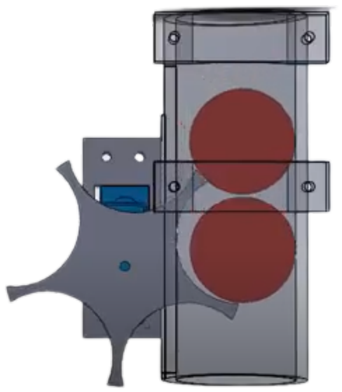
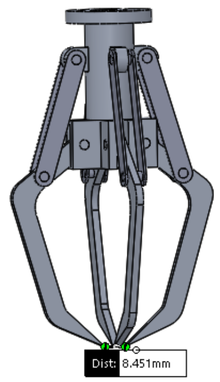
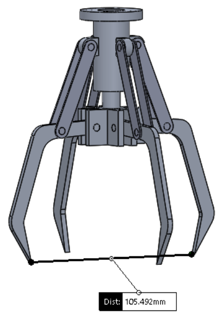


Fig. 4: New mechanism for marker dropper

(finger-to-finger) extension of 105mm with a 1cm actuation.



(a) minimum extension



(b) maximum extension

Fig. 5: Grabber

5) Torpedo: The torpedo is 3D printed using PLA. Its average density is slightly less than that of water. The torpedo, being positively buoyant, can be easily recovered even after it has been fired underwater. Its streamlined design reduces drag. The center of gravity and center of buoyancy does not create torques as they coincide with the center of mass of the torpedo, resulting in a highly stable design. When the torpedo is perturbed from its straight trajectory, the fins make an angle with the flow. Due to this, the fins generate a restoring torque and moves the torpedo to its initial position.

The torpedo assembly uses two IP68 rated solenoid valves (one for each torpedo). Due to the waterproof solenoid valve, the whole torpedo assembly can be installed as a single unit outside the vehicle and mount at the base. Thus, it removes the need for unnecessary air tubes.

B. Electrical Subsystem

Tarang's electrical system includes power sources, sensors, actuators, and all the computational resources required to complete autonomous underwater tasks. New PCBs for multiple purposes like voltage conversion, power routing are designed. There are two layers of stacks inside the hull, which will be used for mounting different electronic devices. This year, a dedicated power board is designed for power monitoring and distribution to all components. The power system includes a custom buck and boost converter designed to our specifications and requirements with the flexibility of placement and connections. With the new microcontroller board for Tarang, we saved a lot of PCB space and money over an Arduino shield used previously by reducing the unused GPIO pins. The new architecture uses 2 ESC boards with 4 ESCs on each board, of which one ESC on each board is a backup in case of failure.

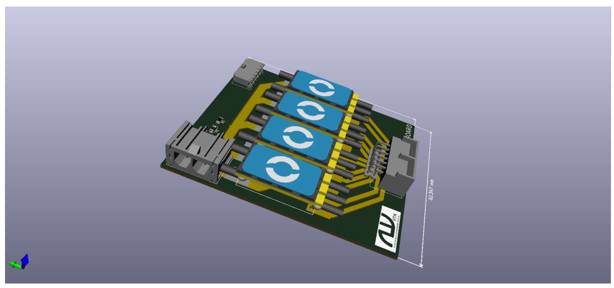


Fig. 6: ESC board

1) Power Management and Distribution System: The vehicle uses two 14.8V 18A-h batteries to power the complete system. One battery is wholly dedicated to the thrusters, which have a high power consumption. The other battery powers the rest of the electronics (micro-controller, servos, solenoid salve, DVL, GPU, preamps for hydrophones) by generating 12V and 19V using high-efficiency buck and boost converters.

- **Custom made Boost Converter :** The custom boost converter powers the onboard

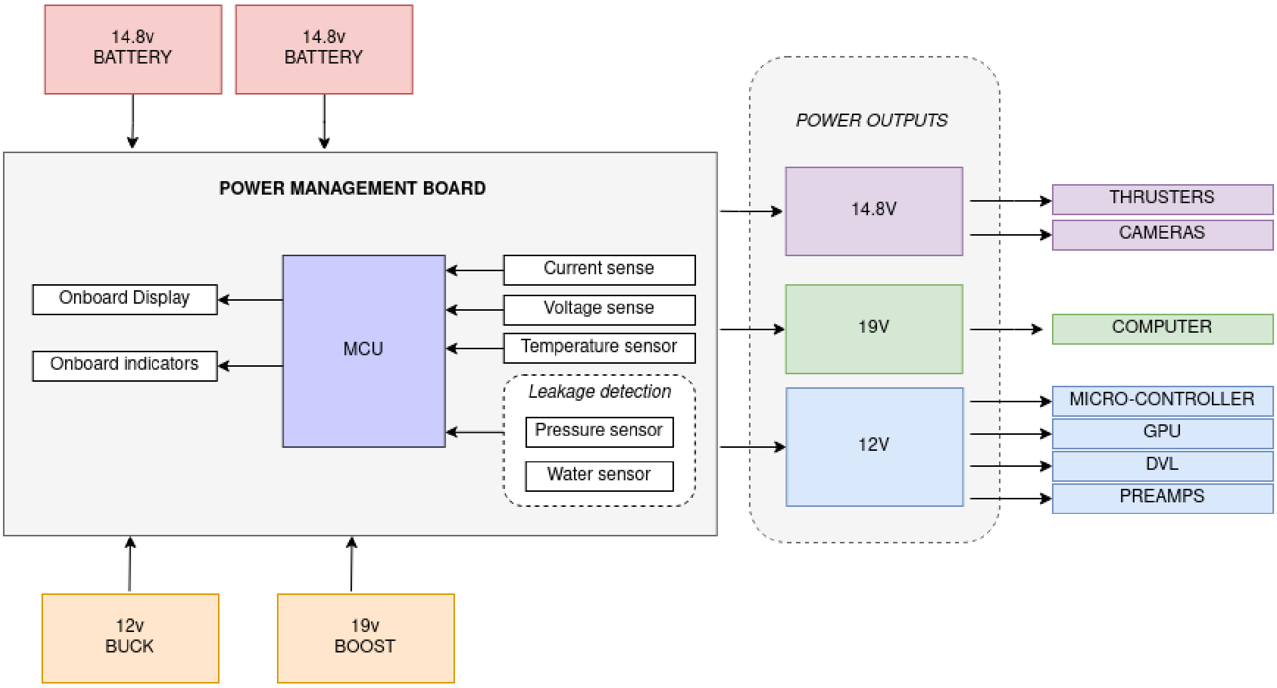


Fig. 7: Power Management in Tarang

computer, which operates at 19V. As the power rating for the onboard computer is high (57W), the boost converter had to be very efficient as the onboard computer will be powered on for the complete duration of the mission.

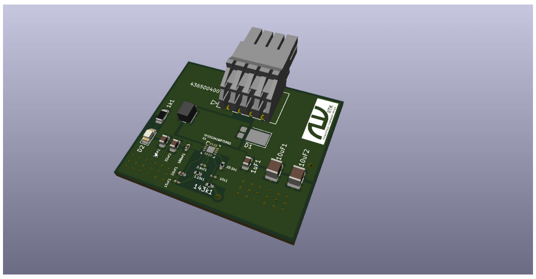


Fig. 8: Boost converter

- **Custom made Buck converter :** Most of the low-power electronics are powered through the custom-designed buck converter, which outputs 12V. The team ensured high efficiency for the buck to minimize the power losses. The power through the buck converter will be controlled using micro-controller GPIO, this allows us to turn off buck's complete power supply and save power.
- **5V Power supply for servos :** The 5V power supply required for driving the servos is created through a regulator with the 12V input from the buck. The power

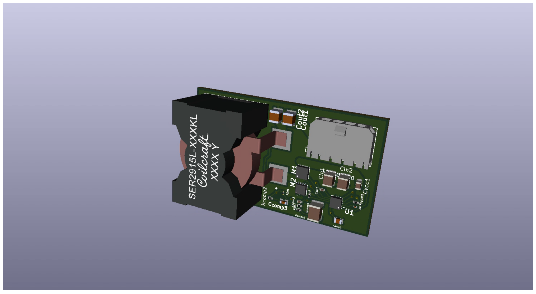


Fig. 9: Buck converter

losses of the regulator are insignificant as the regulator will be turned on only for the duration of servo usage. This saves a lot of space and cost that goes into the making of another buck converter.

The power-board uses an RP2040 micro-controller module to build a robust and small solution while getting sufficient GPIOs for sensors and other peripherals.

2) Safety Measures: The power management board in the vehicle takes care of the under-voltage and overcurrent faults. This was done using a hall effect current sensor(ACS-712) to measure the current flowing through each battery and a simple resistive voltage divider for battery voltage measurement. The microcontroller on the powerboard features a display and multiple LED indicators for battery monitoring and threat alarming. The Kill Switch mechanism has also

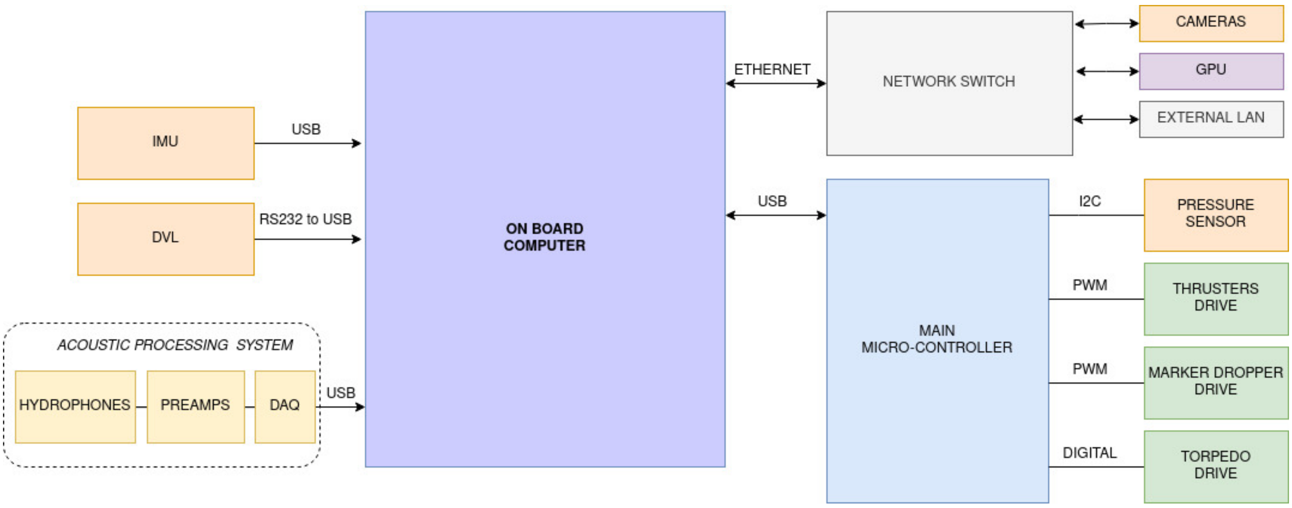


Fig. 10: Electrical Architecture of Tarang

been upgraded using a PMOS while toggling the gate voltage through the reed switch. It also provides a penetrator/connector-free interface for the Kill switch, ensuring better waterproofing.

The internal pressure sensor(BMP388) in the vehicle is used to test for leakage before the vehicle is deployed underwater by measuring whether the hull sustains the applied relative drop in pressure. In addition, temperature reading (provided by the same BMP388) can be used for safely shutting down the ICs if they do not have default thermal shutdown.

3) Sensor Integration: Integration of industrial sensors and interfacing directly with onboard computers enables robust and real-time state estimation. This year, the camera is upgraded to iDS uEye for better color quality and enhanced focus.

Through the help of a newly introduced network switch, the camera feed can now directly be transferred to GPU for object detection and recognition. The external LAN and the onboard computer also have direct control over cameras and GPU. In addition, the new microcontroller board is designed to significantly reduce its size (using only necessary GPIO pins) and organizes connectors for the actuators, manipulators, and several other peripherals.

4) Onboard Computer: The onboard computer is powered by an Intel Core i7 processor. It is powerful enough for image processing and real-time computing. It acts as the primary interface between all the sensors and actuators directly or via some other micro-controller. The new camera is now interfaced to the CPU via Ethernet, which earlier was done using USB.

5) Actuators and Manipulators: The servo-actuated marker dropper and the solenoid valve-controlled torpedoes are all driven through the main micro-controller connected to the CPU via USB. The new ESC breakout board is built on a two-layer PCB with traces exposed to air allowing more current tolerance. The main micro-controller also provides the signal to ESCs for driving thrusters.

6) Connectors: New Molex micro-fit connector series with three configurations (board-board, wire-wire, wire-board) is used on all the new boards for more placement flexibility, making the boards modular.

C. Software Architecture

The software architecture is improved to make the code modular making it easier to test, debug and integrate. In addition, significant advancements in Simultaneous Localization and Mapping strategy (SLAM), tuning of the controller, and vision algorithms were made. The software stack of Tarang consists of dedicated layers for hardware integration, controls, navigation, motion planning, and acoustic localization. The software stack uses the Robot Operating System (ROS noetic) framework by Willow Garage, which works on Ubuntu 20.04 OS that acts as communication middleware between all the processes running on the robot. The code was migrated from Python 2 (which has been deprecated) to Python 3 to use all the latest functionalities. In addition to it, the team has also updated the code to use the latest versions of the

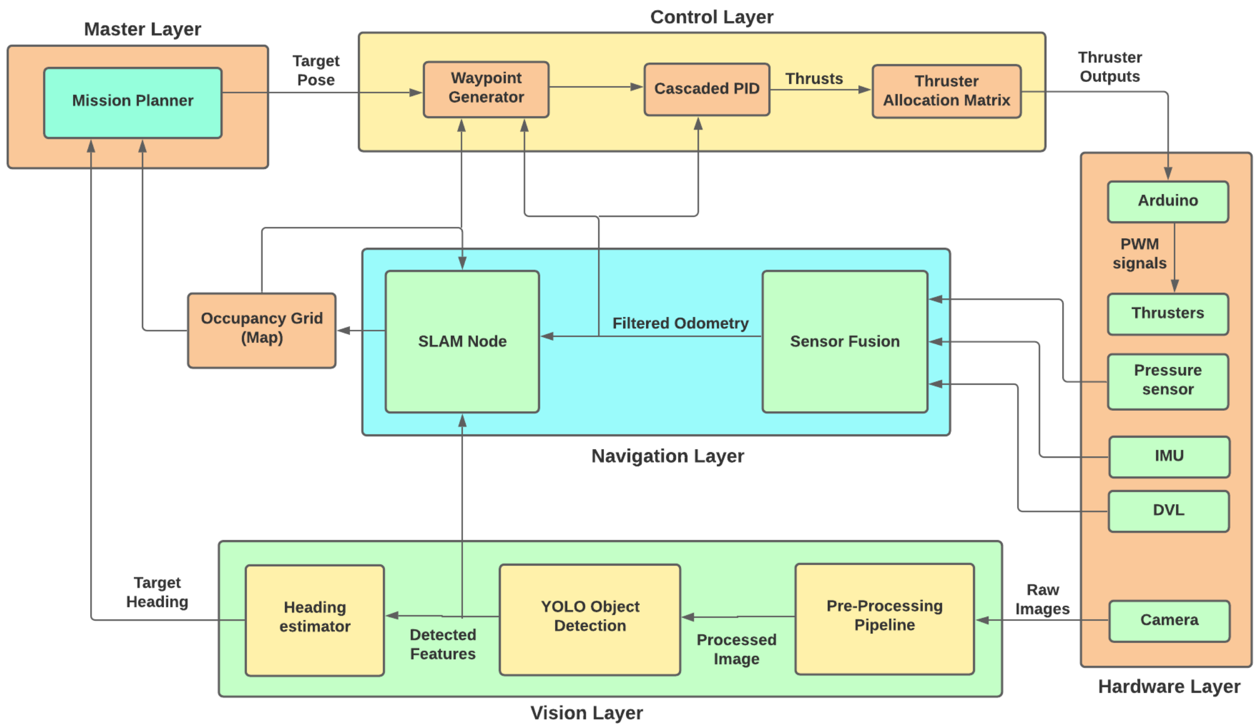


Fig. 11: Software Architecture of Tarang

third-party libraries like OpenCV, YOLOv3, and other ROS packages. The software stack consists of the following layers:

- **Master Layer:** It controls and coordinates the actions of all other layers to perform the tasks autonomously. All the decision making and strategy gets coded in the master layer, which commands the nodes in the other layers to perform different functions. The master layer contains the task-specific code. The signals and instructions for completing all the tasks originate from the master layer.
- **Control Layer:** It contains the implementation of the cascaded PID controller the vehicle uses. The control layer calculates the thrust for each of the thrusters to manoeuvre the vehicle as desired. It also generates the trajectory and waypoints to perform the wanted task.
- **Navigation Layer:** It contains the code for the Simultaneous Localization and Mapping (SLAM) algorithm. It performs sensors fusion, estimates the vehicle's current position in the world, and generates the world map based upon the filtered sensor information.
- **Vision Layer:** It contains the code for all the image processing and vision-related tasks. The vision layer receives the feed directly from the cameras, performs computation on the received data for preprocessing, object detection or visual odometry and sends the processed output to other nodes which require it.
- **Hardware Layer:** It is responsible for integrating sensors with the software stack. It collects the sensors-specific plugins and utilities to receive information from the sensors and publishes it on the topics for the other nodes to use.

Advantages of such a software architecture are:

1) It makes the development easier as different layers can be developed independently and tested asynchronously.
2) It enables easy debugging and troubleshooting.
3) It ensures that the code is scalable and maintainable and provides a straightforward way to integrate external libraries and expand the codebase.

1) Controls: The control system in our new vehicle has been improved by performing fine thruster calibrations and using a cascaded PID controller for precise movements. Tarang is fully actuated with six thrusters providing six degrees of freedom to the vehicle. Each thruster is calibrated to map the thrust vs PWM input pulse. These mappings are used to generate a thruster allocation matrix to distribute the thrusts generated by the PID controller to the thrusters. Since each thruster provides thrust only in a particular degree of freedom, it gives a highly decoupled system that allows the vehicle to perform aggressive manoeuvres. Furthermore, decoupled thrusters with the independent position and velocity controller provide a way to tune the position and orientation controller independently. Hence, the PID systems can be tuned easily.

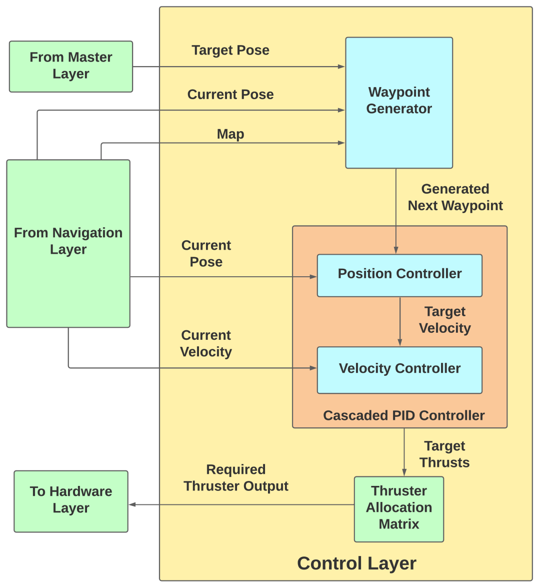


Fig. 12: Architecture of the Control Layer

The software stack has a new implementation of cascaded PID controller for better motion tracking, which considers the error in velocity as well as the error in position to calculate thrusts. It allows a faster compensation with the velocity controller providing a mechanism to prevent overshoot. In simulation testing, the cascaded PID controller provided better results with more robustness to change in environmental conditions. Since the cascaded PID has two separate controllers for velocity and position, respectively, it gives finer control on the system's response by the set of gains of both the controllers. Since the vehicle's weight is less, it can provide faster response, but it also prone to large overshoots and oscillations, so parameters are tuned to provide damping and slow down the response. The motion tracking of Tarang is better than our last vehicle Anahita in terms of lower settling time, almost zero overshoot and ability to perform aggressive manoeuvres.

2) Navigation: The team has set up a sensor fusion pipeline to combine the readings from different sensors and build a better assessment of the measurement using the Kalman Filtering algorithm. The *robot localization* package available in ROS is used for the purpose of sensor fusion, which contains inbuilt implementations of the filtering algorithms like EKF (Extended Kalman Filter) and UKF (Unscented Kalman Filter). It allows for the integration of all basic types of sensor messages like Odometry, DVL and IMU. In addition to these messages, the team included pressure sensor readings to increase depth estimation accuracy. The output state is represented as a 15-dimensional vector. The usage of sensor fusion enabled compensation for the errors in IMU reading due to magnetic interference and position offset.

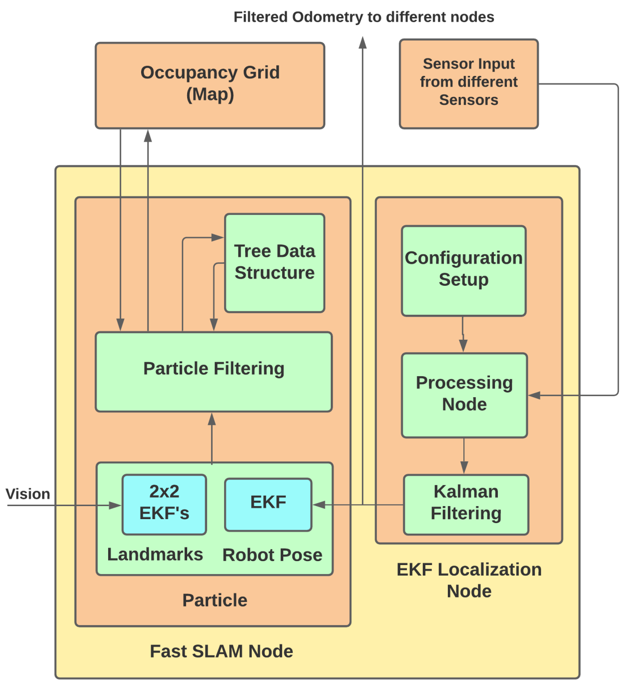


Fig. 13: Architecture of the Navigation Layer

To solve the Simulataneous Localisation and Mapping problem, an implementation of the SLAM algorithm known as FastSLAM [2] has been added. Fast SLAM provides a factored and more efficient way to solve the SLAM problem and provides a way to solve it with a complexity that scales logarithmically with the number of landmarks observed.

In Tarang's software stack, the locations where the tasks are performed are represented as landmarks in the environment using 2X2 Extended Kalman Filters. A set of particles is also stored, each of which holds the state of all the individual landmark EKFs. The final state of the world is represented by a weighted sum of each of these particles. The new observation received from the vision layer about a landmark is used to update the estimate of the world by using the particle filtering algorithm in which the particles with low importance weight are filtered out in the next generation. The navigation layer publishes a world map estimate using a 2.5-dimensional occupancy grid. The occupancy grid stores the estimates of the current state of the robot, global locations of the landmarks and previously traversed locations on the map. The global map helps us in planning and changing our strategy dynamically.

3) Mission Planner: The mission planner contains the strategy to perform all the other tasks. The master layer has the mission planner node, which gives all the different lower layers instruction to accomplish the tasks as per the defined strategy using service-client calls. The mission planner switches on the vision layer to detect the target and switches on the desired task node to execute a task. The task node can also perform the motions such as surge, sway, heave or yaw independently, enabling the vehicle to go from one location to another. A combination of these movements, which can be set in the master layer by the user, achieves the desired motion. The master layer also contains the switches for all the basic motions, the competition's main tasks, and the vision layer. Such a switch system gives easy control over vehicles motion and enables making changes in mission planner effortless.

4) Vision: The vision layer contains the implementation of various algorithms which enable the vehicle to perform a task. We have used the open-source computer vision library OpenCV4 to implement multiple algorithms. OpenCV is an extremely powerful library with inbuilt implementations of various image processing and object detection algorithms. The various stages in the image processing are:

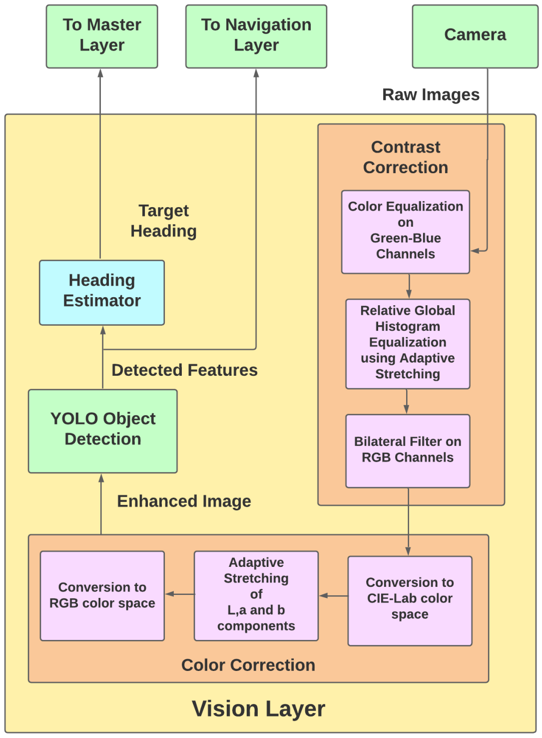


Fig. 14: Architecture of the Vision Layer

Preprocessing: The images captured underwater have low visibility due to attenuation of the propagated light and scattering of light by the absorbed particles. The attenuation of light increases exponentially with distance from the object, due to which distant objects almost disappear or are at least tough to identify. Scattering also affects the color balance and makes the images appear more bluish. Thus, it is challenging to observe the actual colors underwater. To solve these problems, preprocessing of the video feed is necessary by applying multiple filters before we can extract any information from it. Since the images are used to estimate the location of various objects and the vehicle itself, the lengths represented in the images must be true. The camera distorts the features in the image changing their shape and length, so the images are undistorted in the preprocessing pipeline. To undistort images, distortion coefficients of the camera are required. The distortion coefficients of the camera are obtained by its calibration using images of objects of known size and shape. In our case, a checkerboard pattern was used so that the distortion is known beforehand.

The preprocessing pipeline in the vision layer uses the Relative Global Histogram Stretching method to improve image quality by applying

(a) Before Preprocessing

(b) After Preprocessing

Fig. 15: Before and After preprocessing

contrast correction and color correction to the camera output. The contrast correction pipeline applies colour equalization on the green-blue(G-B) channels of the image, followed by relative global histogram stretching. A bilateral filter reduces the noise by using a non-linear smoothing filter to the image. The contrast-corrected image is then passed to the color correction phase, which converts the image to CIE-Lab color space and stretches L, a and b components followed by CIE-Lab to RGB conversion.

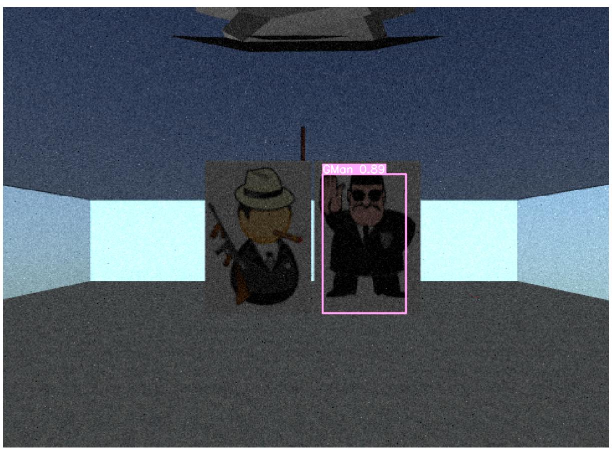


Fig. 16: Object Detection using YOLOv3

Object Detection: For detecting various objects like buoys, gates during the tasks, the YOLOv3 object detection algorithm is used in contrast to classical computer vision algorithms used in our last vehicle Anahita. YOLOv3 provides better results than classical algorithms as it generates the bounding box in a single pass of the input image as compared to multiple passes in classical methods. YOLO is preferred over classical computer vision algorithms as the classical algorithms are highly dependent on the parameters, which may fail in the competitive environment due to changes in lighting conditions. Machine learning-based algorithms like YOLO are robust to change in environmental conditions as they are trained on an augmented dataset to cover all possible scenarios.

To train the YOLOv3 network, rosbags of camera feed were generated by running the vehicle in simulator and recording the camera output and then augmented (rotation, scaling, color variation, occlusion) the frames obtained from these rosbags to generate an extensive dataset.

Experimental Results

Testing is an essential aspect of developing any autonomous system. Unfortunately, this year's pandemic adversely affected our team and disrupted all the AUV's design and testing timelines. However, it gave the team plenty of time to plan, design, and simulate all the subsystems and debug various issues.

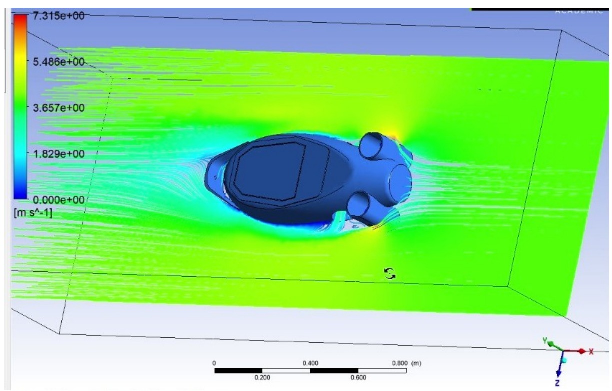


Fig. 17: Drag analysis of hull in ANSYS fluent

After designing the vehicle, next step was to simulate it across various simulation software to get flow visualization and stress analysis. After a set of iterative design and simulation, we finalized the configuration. The vehicle's total drag is 4.019 N, corresponding to 0.6 m/s. The pressure profile across length is shown in figure

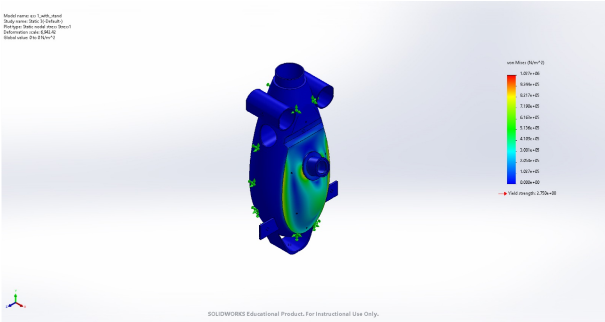


Fig. 18: Stress generated on base due to weight

19 and the stress profile across the length is shown in figure 18.

The yield stress and elastic limit of aluminum are 275 MPa, while the maximum stress of the AUV is 1.027 MPa. Topological optimization is used by removing non-essential weight (area containing less stress). The marker dropper, torpedo, and grabber were simulated in Solidworks' motion study.

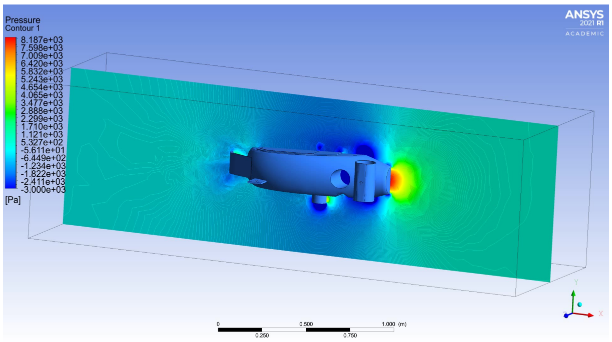


Fig. 19: Pressure Contour corresponding to 0.6 m/s

A structural analysis of the vehicle's base was performed to simulate deformation in the base due to the internal components to finalize the thickness of the Aluminium base. The entire side of the hull was fixed, and a load of 100 N was applied to the base. The maximum deformation was found to be .015mm.

Structural analysis of vehicle stand was performed, on which it will rest when not in water. The thickness of the stand was kept to be 1.5cm, and the analysis showed a maximum deformation of just over .02mm, which was satisfactory.

To test the software stack, extensive Testing was performed in the gazebo using the simulated robot model of our vehicle. The open-source simulation tool *UUV-simulator* was used to simulate an underwater environment. The controller was tested in the simulation and fine-tuned the parameters. The maximum overshoot was limited to less than 5 percent and achieved a settling time of 10 seconds and negligible steady-state error for a unit step signal.

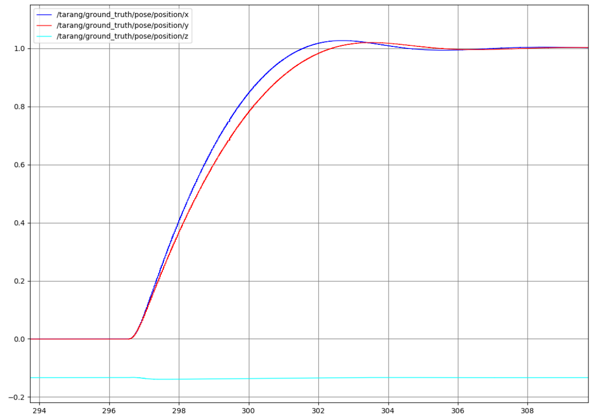


Fig. 20: Step Response

The ability of the vehicle to perform tasks was tested. A simulated gazebo world was made using exact models of the props used in competitions, and the tasks were performed autonomously in that simulation. The plugins from the UUV-simulator library were used to simulate the physics of the world. The gate task, path follower task, buoy task, marker dropper task, and octagon task were performed successfully.

Fig. 21: Gazebo Simulation

ACKNOWLEDGMENT

We want to thank DoRD, IIT Kanpur for funding our project. We would also like to thank the following sponsors for making the design and testing of our vehicle possible: ANSYS, iDS, Solidworks, Sparton, xSense, and Mathworks.

REFERENCES

[1] A. Jain, N. R. Chandra and M. Kumar, "Design and Development of an Open-frame AUV: ANAHITA," 2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV), 2018, pp. 1-5, doi: 10.1109/AUV.2018.8729807.

[2] Michael Montemerlo and Sebastian Thrun et al., ”Fast-SLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem”, 2002 Proceedings of the AAAI National Conference on Artificial Intelligence.
[3] Dongmei Huang and Yan Wang et al., ”Shallow-water Image Enhancement Using Relative Global Histogram Stretching Based on Adaptive Parameter Acquisition,” 2018 24th International Conference on Multimedia Modeling, Bangkok, Thailand.

Appendix

A. Component Specifications

Component	Vendor	Model/Type	Specs	Cost	Status
Buoyancy Control	Blue Robotics	Buoyancy foam	Density:192kg/m3	119$	Purchased
Frame	-	Carbon Fibre	-	-	Selected
Waterproof Connectors	Fisher connectors	Bulkhead connectors	-	3000$	Purchased
Thrusters	Blue Robotics	T200		1690$	Purchased
Motor Control	Blue Robotics	Basic ESCs		25$	Purchased
High Level Control	Microchip	ATmega2560		6$	Selected
Actuators	Hitec	HS-5086WP	Servos	200$	Purchased
Battery	Blue Robotics	BATTERY-LI-4S-18AH-R3-RP	4S, 18Ah, Li-ion	600$	Purchased
Regulator	TI	Boost Converter	19v, 57W	20$	Selected
CPU	Intel	NUC8i7BEH	i7 Processor, 8GB memory, 240GB storage	570$	Purchased
Internal Comm Network	Open Source Robotics foundation	ROS			
External Comm Interface	Telda	Ethernet	5 Port Network Switch	5$	Purchased
Inerial Measurement Unit	Xsense	MTi-300	IP67, 520mW	Sponsored	Received
Doppler Velocity Log	Teledyne Marine	Pathfinder DVL	Bottom-Track velocity	Sponsored	Received
Vision	IDS	UI-5260SE Rev. 4	1936 x 1216,2.35 MPix, 47fps	Sponsored	Received
Acoustics	Aquarian	AS-1 Hydrophones (x4)	STFT	1500$	Purchased
Manipulators	Self Designed	Grabber, Marker Dropper	4 finger Grabber, servo marker Dropper		Build complete
Algorithms:Vision (Open source)	open Source	OpenCV, Darknet			
Algorithms: Acoustics		cross-correlation	sampled at 300kHz		
Algorithms: localization mapping	FastSLAM	External Kalman filter	15 variable state estimator		
Algorithms: autonomy		State machine	ROS Smach		
Open Source Software		IMU, DVL drivers	Serial Protocols		
Team Size	31				
Testing Time: Simulation	100 hrs				
Testing Time: in-water	0				
Inter-vehicle communication	null				
Programming Languages	C, C++, Python				

B. Outreach Activities

The team conducts various workshops and exhibitions throughout the year. Every year during the orientation of the new batch, our team conducts an exhibition to introduce the novel and exciting field of underwater robotics to the incoming freshers. It serves the purpose of introducing robotics to the entire student community on campus. The team also collaborates with its sponsors to hold an educational session for students. Last year, we collaborated with our sponsor ANSYS Inc. to conduct a campus-wide workshop on ANSYS Fluent, demonstrating various fluid simulations. The event witnessed notable participation by the campus students.

During our annual technical fest, Techkriti, many students from across the country visit our institution for the event. The team conducts special workshops during the fest to educate the visitors about marine robotics and portray our work and achievements.

On the occasion of the Diamond Jubilee of our institution, an Open House event was organised for about 4000 high school student. Our team put up an exhibit of our vehicle Anahita and interacted with enthusiastic and inquisitive children from different schools for the entire day. We regularly post educational content and resources using our social media handles on Facebook and Instagram.

Fig. 22: Ansys Workshop

Fig. 23: Educational workshop for students

Fig. 24: Exhibition during Techkriti

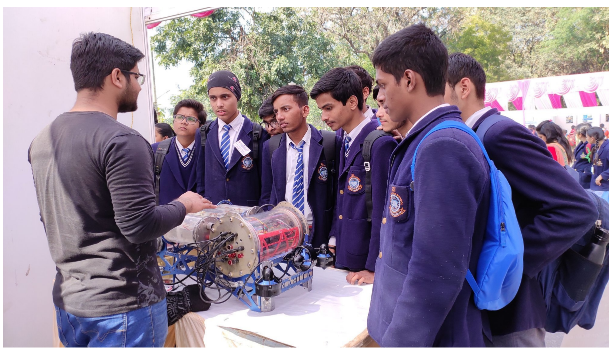


Fig. 25: Open House Exhibition