

Design and Development of an Autonomous Underwater Vehicle VARUNA 2.0

Aditya Natu, Vipul Garg, Purnyatre Gaur,
Dhruv, Piyush Raj, Shivam Sain
Department of Mechanical Engineering
Delhi Technological University
Delhi, India

Upasana Biswas, Deepanshu Bansal, Nayanika Biswas,
Mihir Ranjan, Megha Goyal, Rahul Gupta
Department of Electrical Engineering
Delhi Technological University
Delhi, India

Shaurya Parashar, Raghav Bansal, Chandan Kumar,
Raman Kumar, Ajeet Kumar
Department of Applied Physics
Delhi Technological University
Delhi, India

Aadhar Tyagi, Aarsh Verdhhan
Department of Computer Engineering
Delhi Technological University
Delhi, India

Abstract— Autonomous Underwater Vehicles (AUVs) have a wide range of applications in marine geoscience and are increasingly being used in the scientific, military, commercial, and policy sectors. This paper entails the design and development of an Autonomous Underwater Vehicle named VARUNA 2.0, developed by the undergraduates of Delhi Technological University. The rationale behind the design is to implement and improvise the maneuverability, robustness, stability, and hydrodynamics of the AUV to be able to perform discrete tasks. The AUV is modeled in SolidWorks followed by finite element analysis in Ansys to ensure the AUV sustain the environment it is exposed to. An automatic battery monitoring system is implemented for power optimization along with controlled power distribution. The software stack is built upon ROS framework along with convolution neural network with image processing being handled by OpenCV 3. Real-time analysis resulting in error correcting codes (ECC) assists in smooth navigation in dynamic environments.

Keywords— AUV, ROS, Image processing, BMS, FEA

I. INTRODUCTION

An autonomous underwater vehicle (AUV) is an underwater vehicle capable of self-propulsion which can perform various operations in both shallow and deep sea environments, such as oceanographic survey, bathymetry calculation, retrieval and inspection of submerged structures, environmental monitoring, and even complicated missions like gathering necessary data that aids in understanding marine life and several parameters affecting it. VARUNA 2.0 is a technical project undertaken by DTU-AUV, a multidisciplinary student research project team formed by students of Delhi Technological University with the aim to design, enhance and fabricate an Autonomous Underwater Vehicle which will serve as a radical change in design that is hydro-dynamically stable, lightweight, and power-optimized and can be used to perform various tasks. VARUNA 2.0 is the result of effective integration of multidisciplinary systems incorporated in the vehicle.

The mechanical model is designed to be stable, both in under and above water conditions. The mechanical structure comprises of the electronic pressure hull, frame, battery canisters, and dropper and torpedo mechanisms. The objective of embedded and the power systems incorporated is majorly to

enhance the reliability, modularity and power distribution. The software stack is designed to run in decentralized multithreaded agent architecture, with individual threads administering the pressure sensor, cameras, control system, IMU, mission planner, all executing input and output operations in continuous loops. Dynamic control is accomplished by PID control algorithms. The vision system is designed to perform tasks like buoy, bin, and path and entry gate detection.

This paper is divided in 3 major sections based on the constructional systems of an AUV. Section 3 briefly discusses the approach and design of the mechanical structure. Power systems, electronic components and the energy flow is explained in Section 4 while the following section briefs the control structure, image processing algorithms developed on ROS framework. A conclusion has been derived in the last section to evaluate the overall performance of Varuna 2.0.

II. LITERATURE REVIEW

In literature, hydrodynamics of torpedo-shaped AUVs had been studied. Khairul Alam, Tapabrata Ray, and Sreenatha G. Anavatti [1] worked on optimization algorithms, nondominated sorting genetic algorithm (NSGA-II) and infeasibility driven evolutionary algorithm (IDEA) to develop an optimized framework for AUV. A population-based optimization algorithm was used [2] for design optimization. Multi-objective constraints are derived from mission requirements, and parameters like minimum drag and internal clash-free assembly have been considered while evaluating various preliminary designs. An iterative procedure for optimization of structural design using power flow method (PFM) has been developed [3] while a fault-tolerant system has been designed to monitor the overall AUV [4], to identify the cause and take necessary measures based on the severity of the failure. This approach ensured the continuation of the assigned task either by accommodating the failure or by reconfiguring the control architecture. Selection of material for pressure vessels and common challenges in the development of vehicle were presented by Tadahiyo Hyakudome in [5]. A brief study and data analysis of operating depth, speed, communication, and navigation systems of 33 AUVs has been discussed in [6]. Time taken by

an AUV to complete the assigned tasks vary according to the operating conditions and the complexity of the tasks. Several attempts have been made over the last decade in designing optimized power systems to minimize the overall power consumption and energy losses. Comparison of different batteries employed in AUVs and charging/discharging circuits to improve on the usage of an AUV in remote areas were explored in [7]. Factors affecting the mass and volume available for energy storage systems of an AVP were discussed in [8].

III. MECHANICAL SYSTEMS

The AUV, Varuna 2.0, has undergone some major mechanical design alterations from its previous model, hence, ensuring that this new underwater vehicle meets future design parameters for all research, development and competitive activities. All the individual components and sub-systems have been designed, fabricated and positioned in a way keeping the desired relative positions of the center of gravity and center of buoyancy in mind. The AUV has been designed and assembled in SolidWorks, following which the simulations were carried out both in SolidWorks and Ansys. The AUV is net positively buoyant, with sufficient restoring torque to orient itself in case of disturbance along with roll motion.

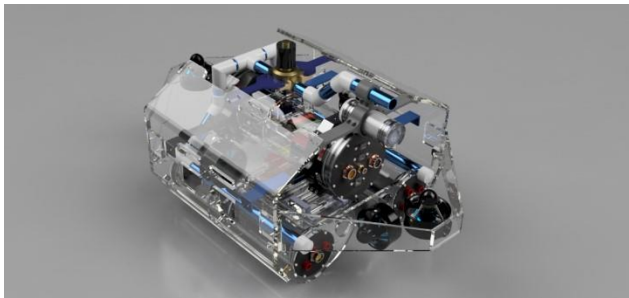


Fig. 1. CAD Model of Varuna 2.0

A. Structural Frame

The frame is uniquely designed by keeping under consideration the static and dynamic stability, structural support, optimized weight and ease of accessibility and disassembly. Rather than using a single material for the frame, clear virgin cast acrylic and ABS are used. It ensures that all components are being fully supported for both static and dynamic loads. The frame has been modeled based on parameters like strength, stiffness, weight, machinability, safety, and aesthetics. Laser Cut process has been used for the manufacturing of the acrylic frame. To achieve more hydrodynamic stability, the frame has been given a higher length to height aspect ratio. To optimize the weight, the parts were topologically optimized while maintaining sufficient strength to meet the requirements. The parts are joined to each other using L brackets or ribs. The frame directly supports all essential systems like electronic hull, battery canisters, pneumatic system, and camera hulls. The parts have been analyzed structurally in SolidWorks.

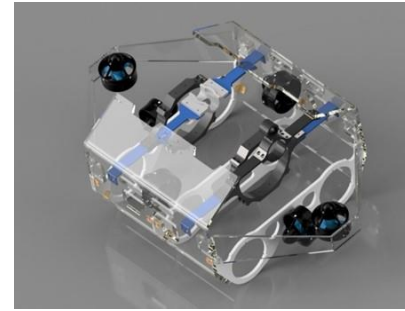


Fig. 2. CAD Model of Frame

B. Pressure Hulls and Flanges

The discreet hulls housed by the AUV are watertight pressure chambers encasing all the sensitive electronic components. Cylindrical acrylic hulls ensure that the stresses are uniformly distributed across the whole body while ensuring maximum utilization of the available volume. These hulls have detachable end caps, attached to flanges. They are capable enough to withstand the pressures and have been designed complying with the principles of fluid statics. Waterproofing is ensured by the use of double silicon radial O-Rings. They have grooves for facial O-rings that are evenly compressed through bolts. Acrylic is being used due to its moderate strength, excellent shock resistance, workability, and comparatively low density.

Hulls have been placed to ensure a lower Centre of Gravity for greater stability of the bot and to avoid the AUV from rolling. There is a total of four hulls, one being the electronic cum software hull, two battery canisters, and front camera hull.

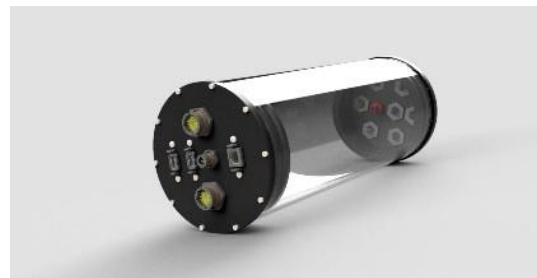


Fig. 3. CAD Model of Electronic Hull

C. Dropper Mechanism

For the purpose of dropping objects from the AUV, a special mechanism has been designed to drop objects at very specific positions in underwater conditions. The location is precisely located by image processing algorithms using the bottom camera which is placed beside the dropper mechanism. Once the AUV is navigated and localized over the region the objects are dropped using a very simple and feasible mechanism. The objects are stored in vertical columns inside a 3D printed part manufactured from ABS and are restricted to fall under gravity force by a plate which can be rotated using a high torque servo motor. A servo motor is specifically used as it can be precisely controlled for inputs in terms of angular values it has to rotate. The columns are located both in clockwise and counterclockwise directions with respect to the mean position of the servo arm. The arm makes the plate rotate until the circular cut-out in the plate coincides with the column profile. As soon it coincides, the object drops and the plate returns to the mean position. After the first drop, the

camera again senses the need to drop another object. If needed, it again follows the same protocol and functions as per the requirements. The mechanism is mounted on the acrylic plate which is further attached to the main frame via brackets.



Fig. 4. CAD Model of Dropper

D. Torpedo System

The AUV employs a simple and cost-effective pneumatic system. The system comprises of PVC tubes, solenoid valve, and a torpedo. The PVC tubes are filled completely with compressed air by a compressor. The air can't release until the solenoid valve opens and directs the air towards the torpedo. Once the target is detected and the AUV is localized about the target, a signal from the microprocessor on-board is sent to the solenoid valve which opens up immediately. The pressurized air is then subjected to the torpedo, which in turn forces the torpedo out with considerable acceleration. The torpedo then traverses a path to reach the target.



Fig. 5. CAD Model of Torpedo System

IV. ELECTRICAL AND EMBEDDED SYSTEMS

The AUV electrical and embedded systems consist of power, control and sensor unit thus interfacing the control subsystem and other sensor devices. The electronic system has been placed on AUV only after thorough static testing on the ground to ensure safety, minimal damage and longevity of the components. The AUV main hull contains electronic hull. Separate hulls are used for housing the batteries, battery monitoring system and battery protection circuitry. Underwater connectors are used for connecting the main hull with other hulls and housing.

A. Electronic Hull

The main hull of VARUNA-2.0 consists of Electronic rack and has been designed keeping in mind the factors of reliability, weight efficiency, faster assembly of components, and isolation of components and ease of handling wires. Erack consists of SBC (Single Board Computer) and AHRS (Attitude Heading Reference System), power distribution board, motor drivers (ESC) and microcontroller. Prevention of electromagnetic interference from the electronic speed

controllers has been ensured by implementing a separate rack for AHRS.

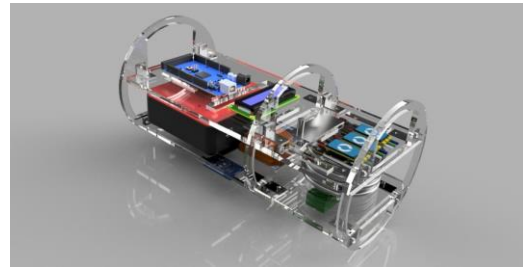


Fig. 6. The electronic rack of VARUNA 2.0

B. Power System and Transmission

The power system of VARUNA-2.0 comprises of Power Source, Battery Monitoring System, and Battery Protection Circuitry. Power Distribution Board and the regulation of power to drive the various systems of the AUV. Two battery hull has been used each consisting of two batteries, protection circuitry, and a BMS has been designed to execute the facility of charging the batteries without interrupting the internal components. Each hull contains two batteries one is used as the main power source and one spare battery is used in case of battery discharge of the main battery.

1) *Power Source:* VARUNA-2.0 is powered by two main 4s, 8,000 mAh. LiPo batteries and two spare 4s 5500 mAh batteries. A spare battery is switched automatically and is used as a power source in case of failure of the main battery. This design provides longer run time to the AUV.

The runtime of the bot has been calculated considering the current requirement of each component. The Power source of the VARUNA-2.0 has been designed to ensure power supply for maximum time to implement longer run time and testing runs. Each of the two batteries is connected to three T100 thrusters through Electronic Speed Controllers. The ideal runtime of the battery can be calculated as follows:

$$\text{Total power being consumed at all time:} \\ (360+66.5+3+0.00625+11.1)W = 440.60625W \quad (1)$$

$$\text{Total nominal capacity of 2 main batteries:} \\ (8+8)*14.8Wh = 236.800Wh \quad (2)$$

$$\text{Ideal operating duration of AUV:} \\ 236.800/440.0625 = 0.53hours = 30.2minutes \quad (3)$$

$$\text{Considering spare batteries then total capacity:} \\ (8+8+5.5+5.5)*14.8Wh = 399.6Wh \quad (4)$$

$$\text{Ideal operating duration of AUV:} \\ (399.6/440.6) = 0.906 hours = 54.41 minutes \quad (5)$$

TABLE I. OPERATING REQUIREMENTS OF MAJOR COMPONENTS OF VARUNA 2.0

S. No.	Component Name	Current Ratings	Operation Voltage	Power Drawn
1	Thruster	5A	12V	60W
2	Intel Nuc I7	3.5A	19V	66.5W
3	Arduino Mega	0.6A	5V	3W
4	Pressure Sensor	1.25mA	5V	6.25mW
5	Servo motor	1A	11.1V	11.1W

2) *Battery Monitoring System:* The BMS of VARUNA 2.0 consists of custom made individual cell voltage monitoring, current monitoring, and capacity monitoring circuitry. It monitors the individual cell as well as the overall voltage of the battery making sure that the battery voltage is between the desired minimum and maximum voltage. Custom made inrush current protection circuitry is used to provide soft start to AUV and limit the inrush current to avoid current spikes. LCD is provided in the battery hull which displays the State of Charge, voltage and runtime of VARUNA 2.0 to enforce dynamic monitoring of battery parameters.

In case the cell voltage is below the threshold, cut off circuitry is provided which monitors individual cell voltage as well as the overall voltage of the battery and cuts the power supply from the battery and at the same time switches it to the spare battery using switching circuit whenever the battery is over-discharged.

MAX17205GEVKIT module is used to monitor the capacity and current drawn dynamically. Arduino Nano is interfaced with MAX module and calculates the run time of the VARUNA-2.0 dynamically. The data from BMS is shared with the microcontroller placed in E-hull through I2C communication.

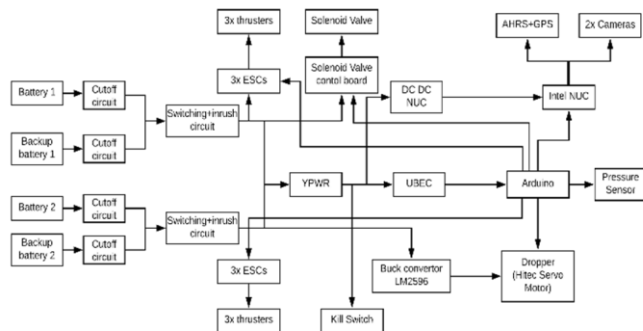


Fig. 7. Hardware Architecture of VARUNA 2.0

3) Battery Protection and Switching Circuitry

a. *Cut-off Circuitry:* There are 2 custom cut-offs circuit used in VARUNA 2.0. It monitors individual cell voltage as well as the overall voltage of the battery. In case the cell voltage is below threshold (3.1 V), it cuts the power supply from the battery. Also, our team has developed a self-made cut-off circuit. It is made up of a Li-Po battery alarm, a capacitor, and a relay. Apart from the cut-off feature, the threshold voltage is adjustable and gives it more flexibility.

b. *Switching and Inrush Circuitry:* The switching circuitry is used for switching the power source to the backup battery in case the main battery gets discharged or fail for some reason. Power relay (TE Connectivity V23076A1001C133), having a current rating of 40A and switching voltage of 12V, is used for switching purposes and is used to switch the power sources. Switching circuit operates in such a way, that the main battery is connected to the NO (normally open) terminal and the backup battery to the NC (normally closed) terminal. The output is taken from the COM (common terminal). The coil is powered by the main battery.

The cut-off circuit will cut the main battery supply when its total voltage drops below 12.4 V or if individual cell voltage drop below 3.1V thus turning the relay from NO state to NC state. Additionally, we have also implemented the inrush protection circuit in the same PCB. For this, we have used MS32 OR540 NTC Inrush Inhibitor by Ametherm. During the powerup sequence, the NTC Thermistor has very high resistance thus dampening the inrush current. As time proceeds, the temperature of the MS32 increases and its resistance drops. In steady state condition, the OR540 has a resistance of 0.01 Ohm. The thermistor is rated for a steady state current of 40 A.

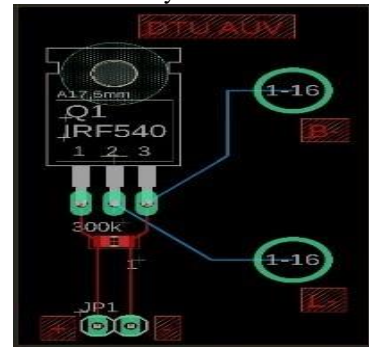


Fig. 8. Solenoid Switching PCB Layout

c. *Kill Switch:* A generic double pole single throw switch is mounted on the vehicle which is fully submersible in water and placed outside the main body of the hull. The switch is used to activate and deactivate the power circuitry of the vehicle. The kill switch has been connected directly to the output of Y-PWR and thus control the power supply of NUC and shuts down the VARUNA 2.0 thrusters, thereby stopping the bot. Along with a common kill switch, spare kill switches are provided on every hull to shut down the components of the corresponding hull only thus ensuring extensive control all over the bot.

d. *Power Distribution Board:* The PDB of VARUNA-2.0 consist of a hot-swappable module, DC-DC isolators and Buck-Boost converters supplying the required power to the equipment.

The DC-DC converters provide different voltage to the electrical systems as per the individual requirements. Two power rails from the battery hull are merged into one rail using Y-PWR (refer Fig. 12) which provides hot-swappable capabilities (load sharing capabilities) so that the battery can be swapped without powering down the SBC and microcontroller. Intel NUC717BNH is selected for processing because of its lightweight and high processing speed (3.5 Gh) and Arduino Mega fulfilled the purpose of the microcontroller as it is very convenient to interface with motor drivers and sensors. The single rail output from Y-PWR is connected to various DCDC isolators. UBEC which has built-in filter capacitors on the input and output is used to supply constant 5V to Arduino mega whereas to power the Hitec Servo motors LM2596 is used. DC-DC Nuc is a dedicated power source which supplies a constant uninterruptible 19V output to Intel NUC. It has many programmable functions that allow it to turn the NUC off when inputs fall below certain thresholds. Additionally, the

input and output conditions can be monitored using its software allowing for easy debugging.

e. Actuator Board: The torpedo system is controlled by a solenoid valve. The valve switches at 12VDC. To activate the torpedo using the microcontroller a switching circuit with an IRF540 MOSFET was implemented. A signal was applied at the gate of the MOSFET to switch it on and allow a larger voltage source across the drain-source. The signal was taken from a digital pin of the Mega, supplying 5V. At $V_{gs} = 5V$, the $R_{ds} = 8\Omega$, which was sufficiently low to turn the solenoid valve on.

V. SOFTWARE SYSTEM

A. Task Planner

This node is responsible for switching between the tasks according to the decision task sequence. The tasks were divided into the primary tasks-buoy detection and localization, gate detection, and localization, bin detection, bin drop, torpedo shooting, resurfacing and the secondary task-line detection and following which was treated as a navigation method.

We utilize SMACH and ActionLib for achieving the same.

The task planner node is written using SMACH which is a ROS-independent python based library used to build hierarchical state machines. It was used to create a structured sequence of performing tasks by calling actions which execute the tasks. Task planning is divided into the following steps:

1) *Task detection:* After exiting the search state, the AUV continues the path till the next task is in the AUV's range of vision followed by deciding the task node to be called according to the sequence. This is made possible with the help of ActionLib library provided in ROS. To accomplish task detection using action, a goal is sent to the action server by the action client. This goal in our case is the detection of a particular object such as path, buoy, gate, etc.

2) *Task Execution:* After deciding the task to be executed, the appropriate task detection and localization node are called to generate the guide data. This guide data is sent to the actuators which act upon these values and move the bot accordingly. In our case, it will be either to move the thrusters with a particular speed or to operate the torpedo or dropper mechanism. Constant feedback is given which provides the Action client about the incremental progress of a goal.

3) *Transition State:* Once the task implementation is completed and the task goal is achieved, a result is sent from the Action Server to the ActionClient. This is sent only once. This is beneficial since it provides the bot with a definite state of completion of a task and the AUV can then enter the transition state. In this state, the line detection node is run which aligns the AUV with the line and navigates to the next task. If the line is not detected in the present field of vision, the AUV searches for the line by

calling the search node which varies the position and orientation of the AUV accordingly.

B. Vision Flow Model

Major challenges that were faced in underwater image processing were image degradation due to dynamic lighting conditions, poor visibility, turbidity and effects such as light absorption, reflection, and scattering by water. The restoration and enhancement of lost contrast and colors are achieved through a contrast limited adaptive histogram equalization known as CLAHE [9]. It is an alternative to adaptive histogram equalization [10] implemented to minimize unwanted amplification of noise present in the image by limiting the enhancement of contrast. The algorithm works by pruning the histogram(pixel intensity vs the number of pixels) at a predefined clip limit before computing the cumulative distribution function. This limits the unwanted contrast amplification in the neighborhood of a given pixel value.

The segmentation of objects in the frames based on color was achieved by applying Otsu's method of adaptive thresholding [11] on the hue-channel of the image represented in HSV (hue, saturation, value) colorspace since the hue channel models the color type. Since our image contains two classes of pixels that can be represented as a bimodal histogram and can be separated as foreground pixels and background pixels, the Otsu's method of thresholding calculates the optimum threshold separating the two classes so that their combined intra-class variance is minimal. Further validations based on geometrical, morphological and spatial properties of the objects are used to accurately extract the contours from the frame.

The localization was carried out by using a modular software package, named VISP [12] that allows fast development of visual serving applications. With our camera providing 2 D features as well as coordinates of feature points in the image, the 3D data i.e. pose estimation of our AUV with respect to the object is provided by this localization algorithm.

For the segmentation task which involves dropping the ball in the bin with a cross, YOLO (You Only Look Once) algorithm [13] was used to classify each image frame as containing a cross or a rectangle and localize the drop coordinates by creating bounding boxes around the detected areas in the real-time video feed. This algorithm uses deep learning and convolutional neural networks for object detection and is one of the fastest object detection algorithms and was therefore chosen for our application of detecting objects in real time. The algorithm which achieves good speed and accuracy are described below:

The image is divided into a grid the size of which can be varied such that each one of the cells will predict all possible bounding boxes and the level of certainty of each one of them. The algorithm proceeds to delete the boxes that are below a certain threshold of minimum probability. The remaining boxes are passed through a non-max suppression that will eliminate possible duplicate objects and thus only leave the most exact of them.

C. PID

We use a proportional-integral-derivative (PID) controller also known as a three-term controller. It uses a feedback mechanism and the fundamental building blocks of control systems called a control loop. A control loop includes the physical components and control functions necessary to adjust the value between a given setpoint (SP) and our process variable (PV). A set point is a final target or the desired value we want to achieve or set our variable at. A process variable is our current variable whose value we are currently monitoring so that we can change it and achieve the desired set point. A PID controller uses the set points and process variable and constantly calculates the difference between the two which is called the error value $e(t)$. It applies a correction based on the proportional, integral and differential value. We use a PID controller as we can tune our pitch, roll, and yaw by applying 3 parallel PID's. It is also easier to implement, robust and highly reliable for the values we want to tune by using it. PID control was selected for yaw control owing to its simplicity, ease of training and precise control as shown in [14] We use a parallel PID loop for configuration.

$$u(t) = K_p * e(t) + K_i * \int_0^t e(t) dt + K_d * \frac{de(t)}{dt} \tag{6}$$

($e(t)$ denotes the error between the current and desired state and $u(t)$ denotes the rpm of the thruster)

The P term is proportional to the current error value and multiplies it with a gain factor K_p . The error value is calculated by subtracting the process variable from the set point and then multiplying this value with K_p . If the controller is only using the P term, then this value defines the output of the controller. If the value of the gain factor is large then it will increase the speed of the control system response. But if it is too large a value then it will overshoot and the process variable will oscillate.

The I term uses the past value of the error and multiplies it with a weighting factor K_i . It sums up the error term over time. This results in an increase of the integral term even with a small error term. The integral response will continually increase over time unless the error is zero, so the effect is to drive the Steady-State error to zero.

The D term predicts the value of the error and multiplies it with a weighting factor K_d . The derivative controller component causes the output to decrease if the process variable is increasing rapidly. It is proportional to the rate of change of the error term.

Effect of each of the controllers K_p , K_i , and K_d are summarised below: We have used three parallel PID's to tune the pitch, yaw, and depth of our bot. The method we have used for tuning these is the Ziegler-Nichols method. This is a heuristic method that produces good values for the following parameters:

- K_p - the controller path gain
- K_i - the controller's integrator time constant
- K_d - the controller's derivative time constant

By varying the three values above, we can directly see the effect of it on our bot. This way we can tune our bot for pitch, yaw, and depth in real time with accurate results. This

method is a bit time consuming but provides accurate results.

D. Stack Structure

The software stack has been developed using Robot Operating System Kinetic and computer vision software was developed using OpenCV Library. The stack was designed as independent, compartmentalized and parallel running nodes each dealing with different functionalities, which work in tandem to achieve a common goal. Use of ROS for stack design helped in enforcing decentralization, process management, adaptability, easier debugging and implemented communication between nodes through custom messages and topics.

The stack was designed as a hybrid topology implemented as ring topology within the topmost nodes, the stack is divided into the modules:

Software Module - It processes the information received from the cameras, makes a decision according to the task selection and forwarding appropriate guide data to the next module.

Navigation Module - Works on the guide data received and calculated the speed and the direction of the thrusters.

Control Module - This module gives appropriate commands to the actuators through the microcontroller and ensures stable movement through PID control.

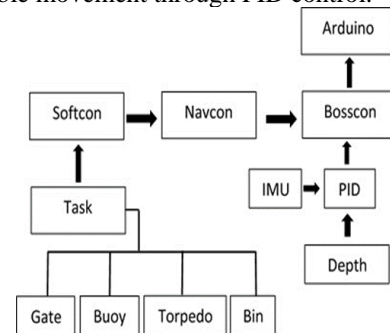


Fig. 9. Software Stack

E. Software Module

It processes the image input through two cameras front and bottom to generate guide data according to the task planner. The package supervises the following nodes:

- Image processing nodes: Pre-processing techniques several image processing algorithms like CLAHE, Gaussian Blur are implemented to overcome issues like poor visibility, turbidity, etc.
- Object Detection and Localization: Various detection and localization algorithms like Canny Edge Detection, Adaptive thresholding, etc. are used along with some Machine Learning classification algorithms like YOLO and CNN.
- Task Planner: This node is responsible for switching between the tasks, guided by SMACH and ActionLib.
- Boss-soft: It sends the generated data to its counterpart at the Navigation module.

F. Navigational Module

It steers the AUV according to the data received from Bosssoft with satisfactory speed and accuracy.

- Convert: A closed loop algorithm implemented through an iterative function which varies the actuators speed and direction according to the difference of the alignment of the AUV and the target given by vision.

- Search: Search mode will be activated, if no tasks are pending, no paths are visible to the AUV. The AUV will start to navigate at predefined paths designed to get the vehicle back on track. This node will also be responsible to navigate the vehicle to the desired location of the next task.

- Boss-Nav: The function of the node is to receive three-dimensional coordinates and communicate with convert node to evaluate precise thruster values.

G. Controls Module

It computes the composite thruster commands and accordingly communicates the speed and direction of the thrusters to the microcontroller.

- Sensor Input Node: Real-time information about the AUV's current position and orientation is acquired through VN-300, an Inertial Measurement Unit (IMU) and Attitude Heading Reference System (AHRS) which combines 3-axis accelerometers, 3-axis gyros and 3-axis magnetometers to give data about the AUV's current yaw, pitch and roll and a pressure sensor which gives data about the depth of the AUV.

- PID Controllers: The control loop feedback algorithm is implemented for yaw depth and pitch control guiding the movement. It takes in the current state data as the parameter and then calculated the speeds and directions of the thrusters as per the Ziegler-Nichols method optimized through graphical visualization and performance analysis.

- Boss-Con: The function of the node is to receive the thruster values to achieve the desired position and orientation as computed by BossNav and the respective values from PID to maintain stability at the desired position.

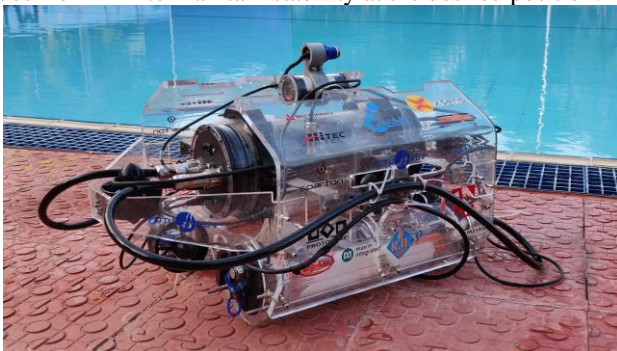


Fig. 10. VARUNA AUV

VI. CONCLUSION

In this paper, the design rationale and the system integration of the VARUNA 2.0 AUV developed, has been discussed in detail. An optimized open frame structure is developed to support the entire system to satisfy the functional objective without compromising the structural strength. Based on the operating velocity of the AUV, hydrostatics of the vehicle has been given due importance for achieving better controllability and maneuverability while for improving the hydrodynamics, the open frame assists in decreasing the drag the vehicle is subjected to by reducing its total frontal area. Emphasis has been laid on the external manipulators to accomplish the required tasks.

Power management and transmission play an important role in determining the endurance of any AUV, and hence strategies have been implemented to make it effective. Failure of AUV due power supply shortage via battery is prevented by switching the power lines to another spare battery placed in each of the battery canisters. Inrush current is also reduced through custom design board. A ROS based software structure is developed to perform multiple tasks simultaneously. Every feature such as object detection for an open gate, path-following capabilities, and the controllers for the stability have been recognized through its separate node. Nodes are classified into sub-systems to reduce the complexity and for the ease of communication among them. While a lot of improvisations have been made in this model as compared to previous ones, there is still a possibility of integrating more advanced systems like acoustics and sonar for better navigation of the vehicle in more adverse conditions.

ACKNOWLEDGMENT

This research and development project was supported by Delhi Technological University. We thank various foundations for assistance with financial support as well as In-Kind sponsorships: VectorNav Technologies, Xsens, SubCtech, Samtec, RJE, Hitec, Sparton, Ion Exchange, Dassault Systmes SOLIDWORKS Corp., Protocase, Maxim Integrated, 3D Paradise, Digi-Key, Blue Robotics, Total Phase. We would like to thank Pacific Sports Complex, Rohini, Delhi for letting us availing the facility of swimming pool for testing the vehicle. We would also like to express our gratitude to our respected Faculty Advisor, Ajeet Kumar, Assistant Professor, DTU, for sharing his pearls of wisdom with us during this research.

REFERENCES

- [1] Alam, Khairul & Ray, Tapabrata & Anavatti, S.G.. (2014). Design and construction of an autonomous underwater vehicle. *Neurocomputing*. 142. 1629. 10.1016/j.neucom.2013.12.055.
- [2] Alam, Khairul & Ray, Tapabrata & Anavatti, S.G.. (2012). An Evolutionary Approach for the Design of Autonomous Underwater Vehicles. 279-290.10.1007/978-3-642-35101-3 24.
- [3] Liu, Guijie & Yang, Tingting & Leng, Dingxin & Tian, Xiaojie & Xie, Yingchun & Mu, Weilei. (2018). Optimization of autonomous underwater vehicle structure shape based on the characteristics of power flow distribution. *Marine Structures*. 62. 77-89. 10.1016/j.marstruc.2018.07.004.
- [4] C. Yang, Keith & Yuh, Junku & Choi, Song. (1999). Fault-tolerant System Design of an Autonomous Underwater Vehicle Odin: an Experimental Study.. *Int. J. Systems Science*. 30. 1011-1019. 10.1080/002077299291877.
- [5] Hyakudome, T. (2011) Design of Autonomous Underwater Vehicle', *International Journal of Advanced Robotic Systems*.
- [6] Aguirre Gmez, Fredy. (2017). State of the Art of Parameters for Mechanical Design of an Autonomous Underwater Vehicle. *International Journal of Oceans and Oceanography*. 11.
- [7] A. M. Bradley, M. D. Feezor, H. Singh and F. Yates Sorrell, "Power systems for autonomous underwater vehicles," in *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 526-538, Oct. 2001.
- [8] Griffiths G, Jamieson J (2004) Energy storage for long endurance AUVs. In: *ATUV international conference*
- [9] G. Yadav, S. Maheshwari and A. Agarwal, "Contrast limited adaptive histogram equalization based enhancement for real time video system," 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), New Delhi, 2014, pp. 2392-2397.

- [10] M. Pizer, Stephen & Philip Amburn, E & D. Austin, John & Cromartie, Robert & Geselowitz, Ari & Greer, Thomas & ter Haar Romeny, Bart & B. Zimmerman, John & Zuiderveld, Karel. (1987). Adaptive Histogram Equalization and Its Variations. *Computer Vision, Graphics, and Image Processing*. 39. 355-368. 10.1016/S0734-189X(87)80186-X.
- [11] Priya, M.S. and Nawaz, D.G.K., 2017. Multilevel Image Thresholding using OTSU's Algorithm in Image Segmentation. *International Journal of Scientific & Engineering Research*, 8(5).
- [12] E. Marchand, F. Spindler and F. Chaumette, "ViSP for visual servoing: a generic software platform with a wide class of robot control skills," in *IEEE Robotics & Automation Magazine*, vol. 12, no. 4, pp. 40-52, Dec. 2005.
- [13] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779788).
- [14] Mohd Aras, Mohd Shahrieel & Teow, B.P. & Teoh, A.K. & Lee, P.K. & Kassim, A.M. & Harun, Mohamad Haniff & Zambri, M.K.M. & Khamis, Alias. (2017). Evaluation and analysis of pid and fuzzy control for auv-yaw control. *Journal of Theoretical and Applied Information Technology*. 95. 4256-4264.