System Design and Implementation of Autonomous Underwater Vehicle "Matsya"

Sneh R. Vaswani, Chintan Raikar, Biswajit Parida, Anaykumar Joshi, Amit Kumar, Ranjeet S. Banthiya, Shivendra Singh, Naman Kothari, Pallav Dhoble, Mihir Gupta, Satwik Kottur, Mohit Beswal

Indian Institute of Technology Bombay, India

Abstract: "Matsya" is an autonomous underwater vehicle (AUV) developed by a team of students at the Indian Institute of Technology Bombay (IITB). Developed over a design cycle of seven months, Matsya is capable of localizing itself in an underwater environment and complete some predefined real life tasks for the Robosub 2012 competition. Majority of the subsystems have been developed and manufactured in-house. To facilitate navigation, the vehicle takes feedback from visual, inertial and pressure sensors. It is well equipped with all the necessary hardware support to process the same and control the actuators. The current version has 5 degrees of freedom and is a platform for integration of all basic systems of mechanical (hull and frame), electronics (power management, motion controller, SBC and sensors) and software (Image processing, debugging platform and control systems). With several innovations in design and flexibility in user experience, the vehicle is a test bench for development and testing of different AUV motion planning algorithms.

1. INTRODUCTION

With the advent of progress in underwater robotics, the need for unmanned autonomous underwater robots is rising in different sectors of the industry. Since they can be conveniently deployed in challenging environments, they find diverse applications; from Oil and Gas Industry, to monitoring power lines, to defence applications for surveillance, reconnaissance etc., besides academic for research observations and teaching.

AUV-IITB is a group of multidisciplinary students studying at IITB in Mechanical, Aerospace, Computer Science, Electrical and Metallurgical Engineering and Material Sciences departments. The goal of the group is to develop an efficient, robust and state of the art AUV that can compete at the annual Autonomous Unmanned Vehicle Systems International (AUVSI) and the Office of Naval Research (ONR) Robosub competition. The 15th Robosub competition will be held at San Diego, California at the Space and Naval Warfare Systems Command's (SPAWAR), Transducer Evaluation Centre (TRANSDEC) facility from July 17th to July 22nd, 2012. The student built AUVs are subjected to completing real life tasks and they are expected to accomplish them without any human intervention.

This is the first attempt of AUV-IITB at building an AUV and hence the objective has been to get the basic systems of mechanical, electronic and software tightly integrated before advancing towards further complexities. Based on the requirements, the team has been divided into three divisions namely: Electronics, Mechanical and Software.

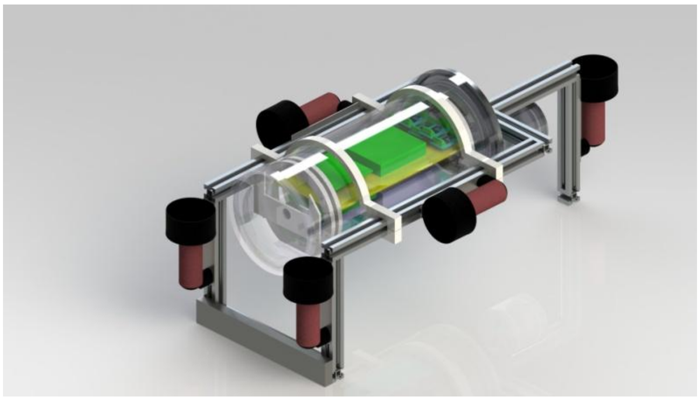


Figure 1. CAD design of Matsya

2. DESIGN OVERVIEW

The design of the vehicle has been done with utmost emphasis on modularity, ease of control, consistency in output and simplicity besides the design constraints set forth by the competition.

The five thrusters give the holonomic vehicle 5 degrees of freedom, namely heave, surge, yaw, pitch and roll. The vehicle can operate at a maximum velocity of 0.6m/s. Weighing 20kgs, the vehicle measures 1.00m in length, 0.53m in breadth and 0.32m in height. The on-board lithium polymer batteries can sustain the vehicle in operation for almost an hour. The vehicle takes feedback from on-board Inertial Measurement Unit (IMU) which functions as an Attitude and Heading Reference System (AHRS), cameras and pressure sensors. In-house designed electronics maintains the modularity in power management and motion control. The software stack has been built keeping in mind portability, user-friendliness, reusability and efficiency.

3. MECHANICAL DIVISION

The mechanical division of the team focuses on the design, prototyping and manufacturing of the pressure vessel (hull), frame, actuators and underwater connectors. The electronics and batteries are housed in the hull with cables penetrating out to sensors/actuators. The actuators are mounted on optimized positions of the frame for complete ease in navigation of the vehicle. The vehicle is imparted a minor positive offset from neutral buoyancy.

3.1 Hull of Matsya

As described, the hull is the water tight region of the vehicle. The focus of the design[1] has primarily been in areas of:

i) Ease in assembly and disassembly

ii) Robust waterproofing

iii) Efficient heat sinking: The electronics in the hull need an outlet for the continuous dissipation of heat generated by them.

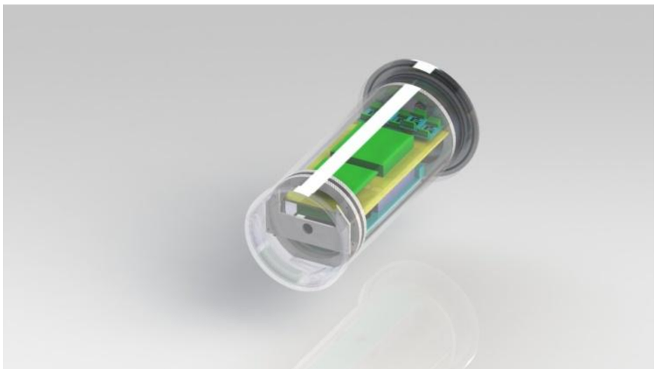


Figure 2. SolidWorks CAD model of the hull

To minimize the drag forces [2], a cylindrical hull design was incepted (figure 2). The lower surface is made flat to ensure the camera captures undistorted images. Acrylic is chosen for the casing of the hull and for the fixed front end cap. Aluminium is chosen for the rear end cap to vent the heat through conduction. The following considerations are made for selection of the materials:

i) Transparent: The hull is kept transparent for visual detection of water seepage, viewing electronic displays/indicators and cameras field of view.

ii) Light weight.

iii) Non corrosive.

iv) Availability and ease in machining.

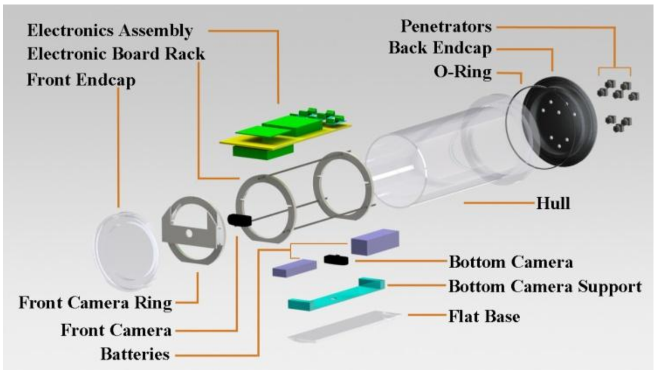


Figure 3. Exploded view of the hull

Figure 4. Rear end cap with O-ring groove

Waterproofing: Waterproofing is the most crucial aspect of hull design. The team experimented with different end cap designs and water proofing techniques. The removable end cap is the most anticipated region for leakage. The team has used rubber O-rings with grooves (figure 4) for the same. The O-ring is mechanically squeezed between two surfaces to seize the passage of any liquid into the hull.

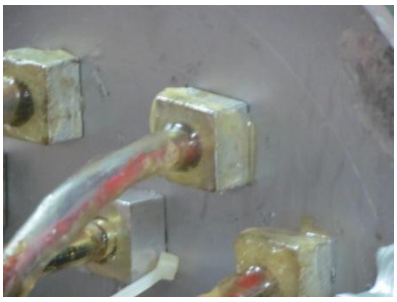


Figure 5. Penetrators on end cap

Penetrators: The team has designed and manufactured the underwater penetrators mounted on the aluminium end cap (figure 5). The cables for the thrusters and batteries are routed out of the hull through these penetrators.

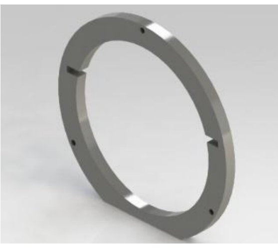


Figure 6. Support rings for electronics stack

Electronics Stack: The electronic boards are stacked inside the hull on either sides of an acrylic sheet. The plate is held on support rings (figure 6), giving the user the flexibility to slide the boards inside out.

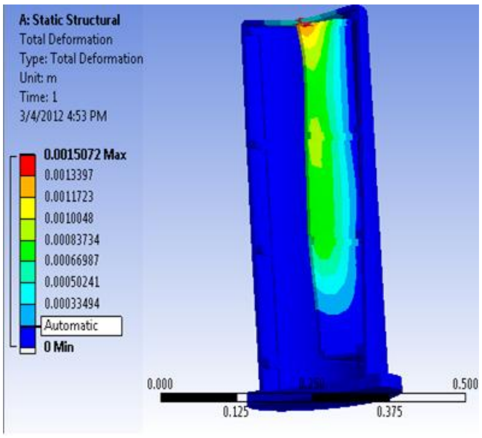


Figure 7. Total Deformation distribution of hull

Structural analysis: Static structural analysis of the frame is done using ANSYS CAE finite element analysis (FEA) software package (figure 7). To withstand the pressure of water at 50 feet, the optimum thickness of the hull is evaluated and the obtained results have been tested experimentally.

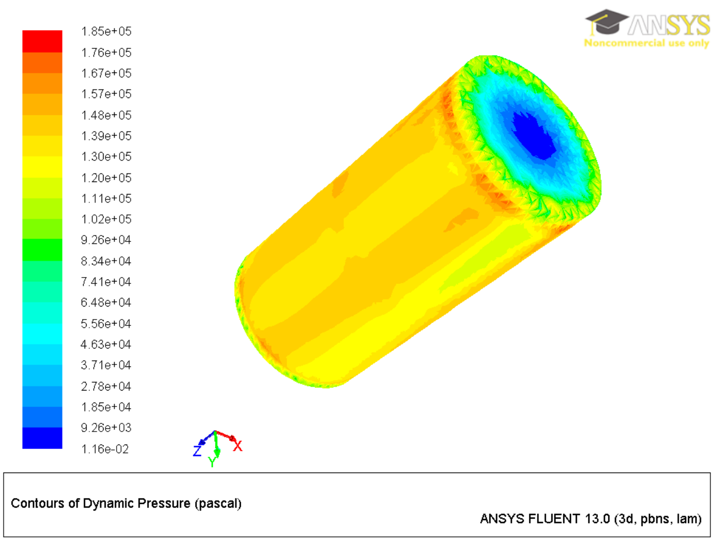


Figure 8. Fluent CFD Analysis of Hull

The drag co-efficient and hence the drag forces have been simulated using Computational Fluid Dynamics (CFD) modelling on ANSYS FLUENT. The objective of the analysis is to give an estimate of the force required to steer the vehicle at constant speed and provide some insight in judging the practical thrust requirements. At a constant speed of 0.5 m/s (assumed), the estimated drag co-efficient of the vehicle is 0.817 with a thrust force requirement of 15.235N. The maximum attainable speed of Matsya using the given thrusters is 0.65 m/s. The CFD calculated values of drag forces agree fairly well with the experimental results (figure 8) [3].

3.2 Frame of Matsya

Figure 9. Skeletal structure of Matsya's frame

Major emphasis is laid on modularity, static and dynamic performance besides robustness of the frame. Aluminium 8020 sections are used to construct Matsya's frame as it serves the primary purpose of strength and in-plane alignment of all the thrusters. This offers an innate flexibility in placement and adjustment of thruster positions to alter the dynamic behaviour of the vehicle. All the sections are anodized for better corrosion resistance.

Matsya's frame supports the hull, five thrusters, ballast weights besides the enclosure for Ethernet cable and battery charging ports. Custom fasteners and holders are designed and manufactured for assembling the frame. It facilitates adjustment of thruster positions and provides modularity in construction. Studs are provided for the protection of heave thrusters and to support the vehicle when placed on ground.

Design of the frame and position of the mountings ensure maximum separation along the vertical line between centre of gravity and centre of buoyancy. This provides natural stability in roll and pitch motions [4]. Special emphasis is given to symmetric mounting of components; both on frame and inside the hull. This provides better yaw control by reducing the amount of unbalanced torque during motion. Critical parts of the frame and the vehicle were analyzed using ANSYS. Besides prediction of failures, the results also helped to optimize the weight to strength ratio.

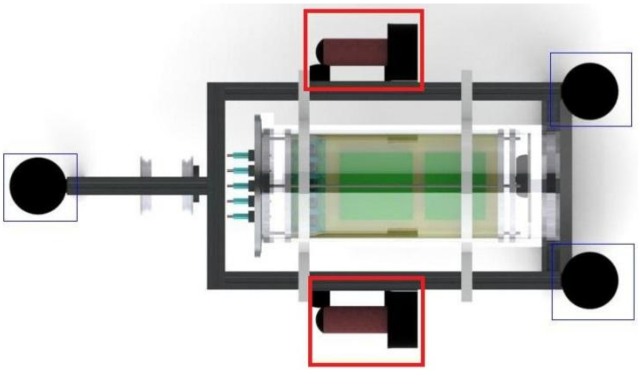


Figure 10. Orientation of thrusters

Thruster mountings: Five thrusters have been mounted on Matsya to provide five degrees of freedom except sway. The configuration shown in figure 10 suffices the requirement besides reducing overall weight and expenditure. Thrusters that actuate yaw and surge motion (marked in red squares) are strategically mounted to counter the torque by drag force on centre of gravity. This naturally controls the pitch of the vehicle under acceleration/deceleration. The thrusters marked in blue squares actuate motion along heave, roll and pitch axes.

4. ELECTRONICS DIVISION

The electronic systems in the vehicle act as platform for software systems to be executed. The processing platforms are chosen based on the basic needs of vision processing, controls and power management (figure 11). The hardware architecture is designed with emphasis on modularity and scalability in the future. Majority of the boards are designed and populated in-house.

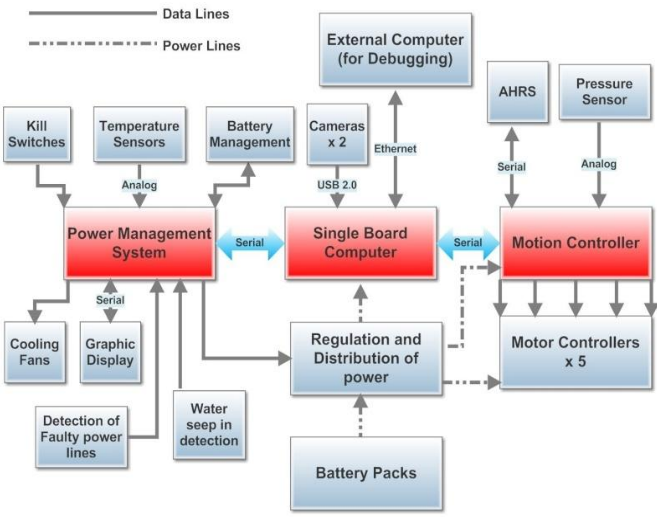


Figure 11. Electronic Hardware Architecture

4.1 Power Management

The vehicle runs on power supply from two lithium polymer battery packs: 14.8V and 11.1V. The supply line for the thrusters is kept separate from that of the electronics to avoid cross talk of motor noise into the electronic circuits. An 8 bit microcontroller (Atmega 2560) ensures the power management of the vehicle.

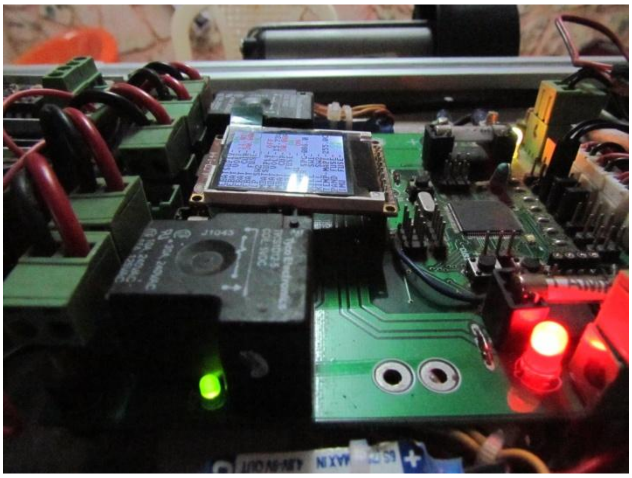


Figure 12. Power board of Matsya

Battery Management: It monitors the power levels of the battery besides communicating the power consumed to the SBC with time stamping for characterization purposes. Hall Effect current sensors mounted on the power board continuously measure the current consumed from both the supplies. Switching regulators ensure regulated power supply to different electronic boards mounted in the hull.

Temperature Control: On-board point contact temperature sensors keep a check on the temperature in the hull and control the speed of the cooling fans. If the temperature of the hull at any point exceeds a certain threshold then the power board kills the supply lines to the vehicle.

Water seep in Detection and Kill switches: If water accidentally seeps into the hull then the power board detects the same and disconnects the supply lines to the electronic boards. Magnetic kill switches on the vehicle are interfaced to the power board to enable/disable the supply lines as desired.

Power Distribution: The board ensures clean power distribution besides visual and electronic detection of faulty lines. A graphic LCD displays the status of the supply, power lines, temperature etc. The system has different visual indicators for displaying different parameters like power levels, emergency situations etc. The transparent hull allows the user to observe them from a distance (figure 12).

4.2 Single Board Computer

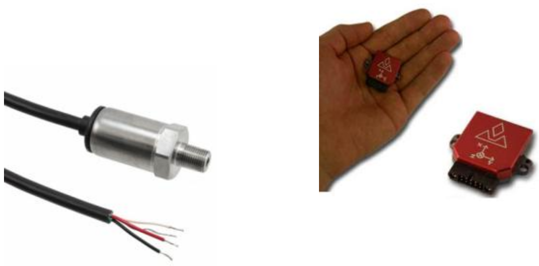
Figure 13. PandaBoard ES

PandaBoard ES is used for vision processing and communication to on-board/off-board processors. It is a single board computer development platform based on TI's OMAP4430 system on a chip. The SBC features a dual core 1.2GHz Arm cortex-A9 CPU, 384 MHz GPU and 1GB RAM. The primary reason for choosing the PandaBoard ES (figure 13) is its small form factor (100 × 110 mm) and low power requirements (18W) as compared to its computational capacity. The board takes feedback from two USB 2.0 cameras. It also communicates the entire status of the vehicle to an off-board processor through Ethernet. The communication with on-board processors is done serially.

4.3 Motion Controller

The motion controller of the vehicle takes feedback from the inertial sensors, the pressure sensor, the SBC and controls the thrusters. It executes the control loops based on directives and set points from the SBC. An 8 bit microcontroller clocked at 16MHz and 8KB SRAM (Atmel's Atmega 2560) is chosen for the same.

4.4 Sensors and Actuators



(a)Pressure Sensor (b) AHRS

Figure 14

Pressure Sensor: The pressure sensor (figure 14a) is used to estimate the depth of the vehicle underwater. An absolute pressure sensor from SSI technologies P 51 series is mounted at the aluminium end cap. It gives an analog feedback to the motion controller.

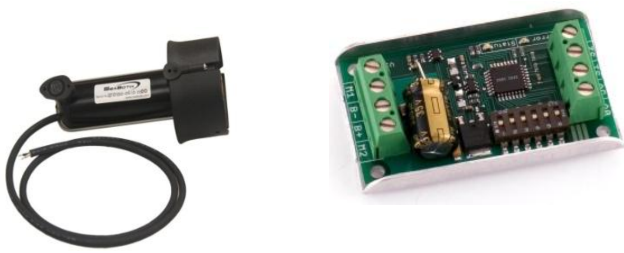
Attitude Heading Reference System (AHRS): Vectornav's VN 100 (figure 14b) is used to estimate the heading of the vehicle. It has a 3-axis accelerometer, 3-axis gyro sensor and a 3-axis magnetic sensor besides a 32-bit processor which serially communicates with the motion controller at 20 Hz after fusing the outputs from the inertial sensors.

(a) Camera (b) Current Sensor

Figure 15

Cameras: Images are captured using two Logitech C310 HD web cameras (figure 15a) placed at the front and bottom of the hull. These cameras are USB 2.0 compatible and are interfaced to the SBC.

Current Sensors: These are Hall Effect current sensors(ACS 709) that are used by the power board to get a feedback of current been consumed from the batteries(figure 15b).



(a) Thrusters (b) Motor Drivers

Figure 16

Actuators and drivers: The thrusters mounted on the vehicle consume 80 watts to deliver a thrust force of 12N. These are the BTD 150 offered by Seabotix. The Syren 10 from Dimension Engineering is used to drive these PM DC motors (figure 16). With their small form factor, they can deliver up to 180 watts continuously. The drivers are operated in lock anti phase drive mode for motor control.

5. SOFTWARE DIVISION

Software and firmware development are needed on different platforms. Firmware requirements for the power management, motion controller and the software stack on SBC are the major needs. For the ease of testing and quality user experience, a debugging platform is also developed.

5.1 Software stack on SBC

The software stack is built on top of the Ubuntu Core GNU/linux distribution, which is a minimal ubuntu distro. The entire software stack has been written in C/C++. The software stack is responsible for the mission planning, image processing, and handling the communication between the hardware modules, power board and motion controller board.

Middleware: The objective of the middleware in the software stack is to provide a neat and clean interface for applications to communicate with each other. This maintains the modularity in the software system. It provides an abstract layer on top of the operating system API.

Our custom middleware (figure 17) is based on the philosophy of the "Internet". Processes are imagined as nodes in a network, and a central process is imagined as a router. Every process can send "mails" to every other process using "addresses". The router process is responsible for updating the "address vs. process table"

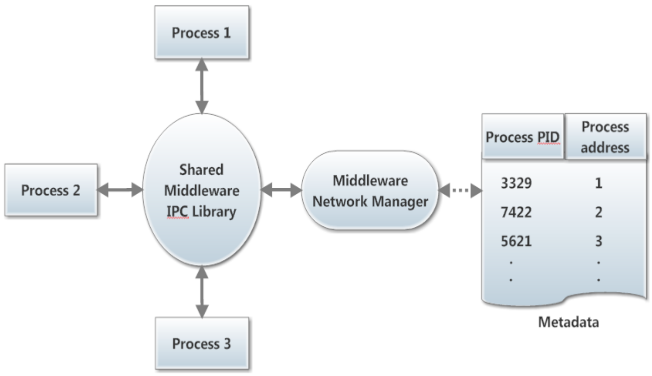


Figure 17. Block diagram of the middleware

and broadcasting it to every process. The middleware is implemented using pipes and signals. The device I/O for the serial ports and the camera are encapsulated and presented to the applications as I/O objects.

Vision: Software modules for image processing are built on Intel's Open CV Library, which provides optimized image processing functions and flexibility to code in C. The front-facing camera is used for localization while the bottom camera is used primarily for orienting the vehicle along the planks placed at the bottom. Subroutines from the FFTW library are used to compute Fourier transforms due to their novel code generation and run time self-optimization techniques.

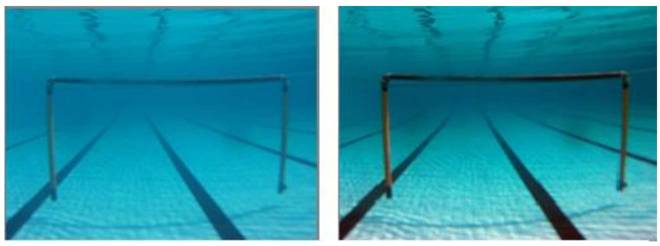
Images tend to get degraded as the vehicle goes underwater (figure 18). As depth increases, the amount of light on objects decreases.

(a)Image in air **(b) Image in water**

Figure 18

Refraction effects are observed due to presence of camera in a different medium as compared to the environment (figure 18) [5].

For this reason, the team has worked upon implementation of different algorithms for underwater image enhancement [6]. Primarily to remove non uniform illumination and enhance contrasts in images, the frames were filtered with a homomorphic filter. Wavelet denoising was also done before the images were used to track objects. But these algorithms turned out to be computationally expensive on the SBC.



(a) Original Image **(b)Enhanced Image**

Figure 19

Color Contrast Stretching: Using the principle of histogram normalization, the obtained underwater images are stretched to obtain the full possible contrast. This enhances the color quality and removes the effect of differential scattering of light by water (figure 19) [7].

Gate Pass Detection: The detection of validation gate is done by detecting canny edges (figure 20) of the enhanced underwater images. The algorithm ensures correct set of edges are chosen and estimates the position of the gate for navigation.

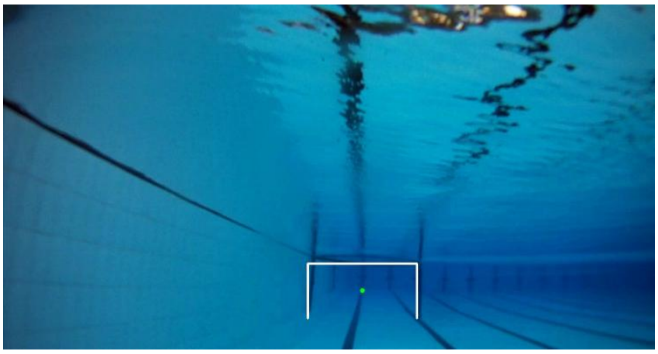


Figure 20. Gate Pass detection using Canny Edge Filter

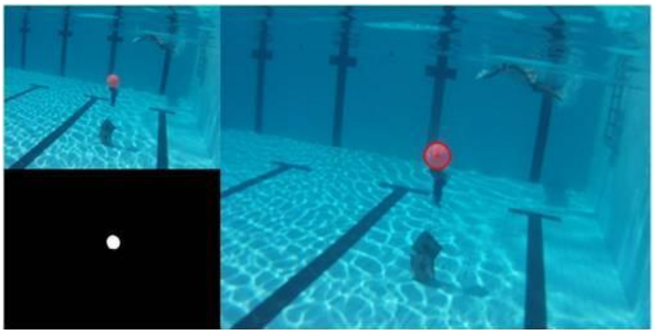


Figure 21. Buoy detection

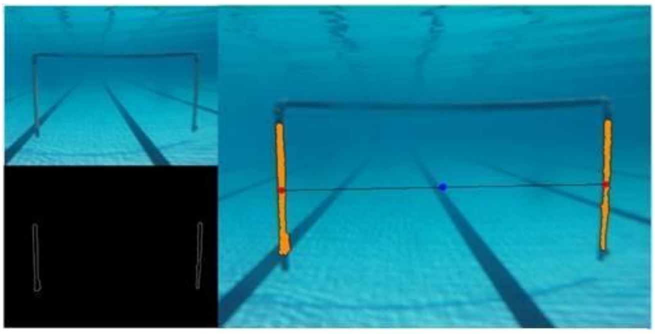


Figure 22. Gate Pass detection using Colour Identification

Colour identification is done by thresholding; using colour masks and the obtained threshold image is used for further processing. Contour detection, Hough transforms, Image erosion/ dilation are used based on the respective tasks (figure 21 and figure 22).

b) Debugging Platform

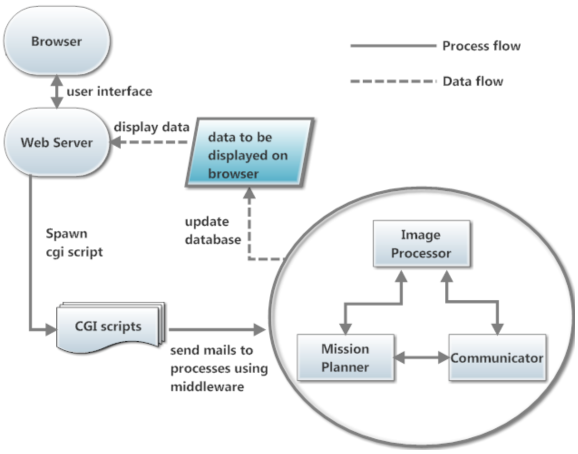


Figure 23. Block Diagram of Debugging Platform

Before the vehicle is deployed to complete tasks autonomously, every software module needs to be perfected. To ease the process and minimize time consumption, a robust debugging platform is needed. The requirements that it needs to support are:

i) Observe the current status of the vehicle. The user should be given the flexibility to see the images captured by the camera, status of mission planner, detect of emergency alarms etc.

ii) Ability to reprogram any controller on the vehicle from an external processor.

iii) The user should be able to update any control/ image processing parameter from an off-board computer. This may be done dynamically, when the vehicle is in operation to visually observe the effect of the changes made.

The debugging platform (figure 23) is implemented using C, JavaScript, AJAX. The vehicle is connected to an off-board computer through Ethernet. It presents itself as a web based interface. The PandaBoard hosts an apache web server which is spawned as soon as the Operating system on the SBC boots up. The raw images from the on-board cameras as well as the processed images from the image processing task are updated on the browser at 2 fps. The image processing parameters can be tuned using the web interface through user friendly task bars.

The control equation parameters for the stable navigation of the machine can also be updated using the web interface.

Corresponding to any change in parameter on the web interface, CGI scripts are spawned by the web server on the SBC. These CGI scripts use the middleware to send "mails" to relevant processes. While tuning the control parameters for the control equations, the CGI scripts mail to the “Communicator” task, which ensures the efficient transfer of data (parameter values) from the SBC to the motion controller board. Similarly, while tuning the image processing parameters, the CGI scripts mail the “Image Processing” task which updates its parameters accordingly.

The debugging platform also provides the flexibility of programming the firmware on the microcontroller boards in the vehicle. This feature is very useful when one is iteratively optimizing the code with regular test runs. Any on-board microcontroller can be programmed by uploading the .hex file on the server (using the traditional "browse and upload" feature provided by HTML).

c) Firmware

The firmware for the microcontrollers on power board and motion controller boards is developed in C. The primary reason for choosing C is its portability and flexibility in optimizing it much more than higher level languages. Of all the languages, C is the closest to assembly and unlike assembly is very succinct.

6. CONTROLS

PID control loops for each of the axes generate the respective corrections for the desired set points. Each loop for each axis has different sensors and hence different refresh rates [8, 9]. Despite varied refresh rates generating corrections for the same thrusters, the vehicle can accurately maneuver along desired axis due to well tuned PID controllers. The set points to be achieved are serially updated by the SBC. The SBC also directs the motion controller for the sensor to be used depending on current task.

The image processing modules compute the spatial co-ordinates of interest and communicate the same to the motion controller. These act as the set points for the control loops which are dynamically updated as the tasks proceed towards completion. The SBC also directs the sensor to be used for feedback.

7. STRATEGY

a) Navigation through Gate Pass:

As the vehicle is immersed in water, it settles at a pre defined depth and searches for the gate pass.

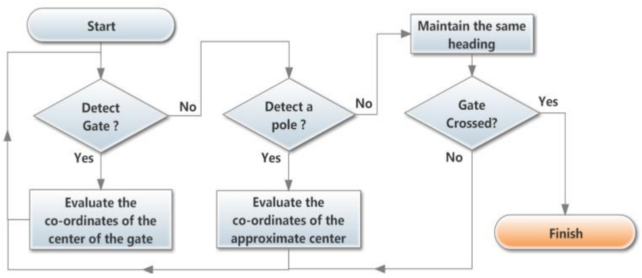


Figure 24. Flowchart for "Gate Pass" task

On detecting the gate, the spatial co-ordinates of its centre are sent to the motion controller which act as error for yaw controller. The vehicle navigates towards the gate pass and aligns its trajectory towards the centre of the goal post. The length of the detected gate act as error for the surge controller. The depth is maintained constant for the entire task. As the gate is crossed and is out of sight of the vehicle, it starts searching for the path (figure 24).

b) Orient along the path:

After crossing the gate, the vehicle detects the "path" and orients along the same before heading towards the next task.

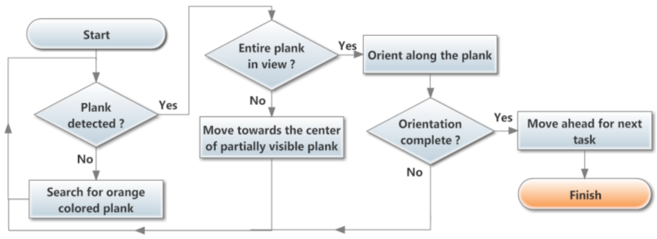


Figure 25. Flowchart for "path" task

As the bottom-facing camera sights the path, it proceeds towards the centre of the visible region of the plank. This is done to ensure the centre of the plank continues to remain in the field of view of the vehicle whenever it begins to orient along the path. As the threshold of area increases, the vehicle begins orienting along the path. Once the vehicle orients within a certain tolerance, it proceeds towards the next task (figure 25).

c) Training:

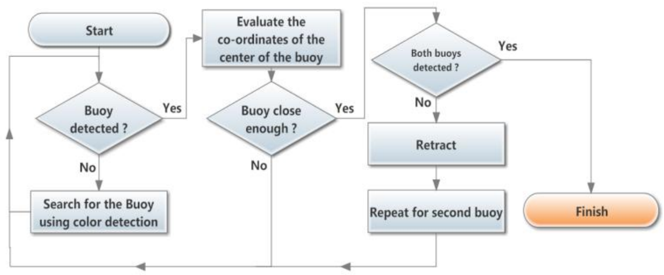
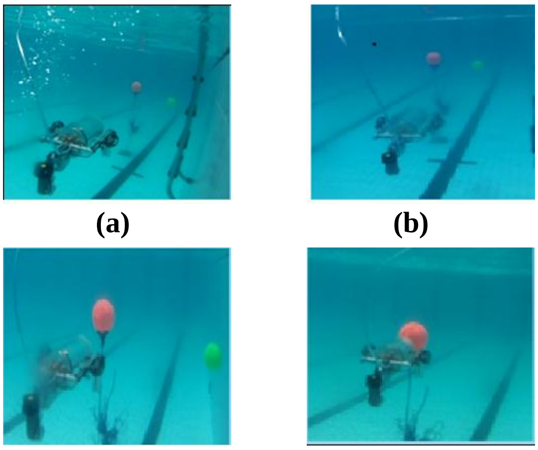


Figure 26. Flowchart for "training" task

The AUV searches for the buoy in the respective priority order. On detection of the same, the the spatial co-ordinates of the centroid of the buoy act as errors for yaw and heave axes controllers. The area of the buoy is used for correction along surge axis (figure 26).

8. APPROACH AND RESULTS

Since the team is working on underwater vehicles for the first time, it necessitated the perfection of basic systems of each division. Till the hardware was designed, prototyped and fabricated, the software team started building the framework for the same. The image processing modules were developed on land mobile robots and the motion controller firmware was developed on a small test bench. The design of the hull was revised 7 times and prototyped 4 times before the final hull design was fabricated. The vehicle has been rigorously tested and iteratively improvised (figure 27).



(a) (b) (c) (d)

Figure 27. Matsya detecting buoys at the IITB swimming pool

10. ACKNOWLEDGEMENT

AUV-IITB team would like to thank every individual/organisation who has supported the team in developing “Matsya”. The team thanks the Dean R&D and the Dean ACR of IIT Bombay for their financial support. It also thanks the Aerospace Department of IIT Bombay for providing lab workspace to the team. AUV-IITB would especially like to thank the faculty advisors for the project; Prof. Hemendra Arya and Prof. Leena Vachchani for their guidance at every step.

The team would like to thank its corporate sponsors for their kind support. This journey would not have been possible without their presence: Vectornav, SolidWorks and Seabotix.

11. REFERENCES

[1] Gonzalez L. A., Design Modelling and Control of an Autonomous Underwater Vehicle, Honours Thesis (2004)

[2] Jones, D. A., Clarke, D. B., Brayshaw I. B., Bacrillon, J. L. and Anderson B., 2002, The calculation of hydrodynamics coefficients for underwater vehicles, Defence Science and Technology, Platforms Sciences Laboratory.

[3] Toncu G., Stancu V. and Toncu C. D., A Simplified Model for Evaluation of an Underwater Vehicle Drag, Proceedings of the 2nd International Conference on Maritime and Naval Science and Engineering, Brasov, Romania, (2009)

[4]Naomi Ehrich Leonard, Jerrold E. Marsden, Stability and drift of underwater vehicle dynamics: Mechanical systems with rigid motion symmetry, Physica D: Nonlinear Phenomena, Volume 105, Issues 1–3, Pages 130-162, ISSN 0167-2789, 10.1016/S0167-2789(97)83390-8, (1997)

[5] Atsushi Yamashita, Megumi Fujii, Toru Kaneko, “Color Registration of Underwater Images for Underwater Sensing with Consideration of Light Attenuation”.

[6] St´ephane Bazeille, et. al., “Automatic Underwater Image Pre-Processing”, (2006)

[7] Kashif Iqbal, et. al., “Underwater Image Enhancement Using an Integrated Colour Model”, IAENG International Journal of Computer Science, (2007)

[8] Wettergreen D., et. al., Autonomous Guidance and Control for an Underwater Robotic Vehicle, Proceedings of the International Conference on Field and Service Robotics (1999)

[9] Graver J., Underwater Gliders: Dynamics, Control and Design, PhD. Dissertation, Mechanical Engineering Dept., Princeton University, (2005)