System Design and Implementation of Matsya 2.0, a Technology Demonstrating Autonomous Underwater Vehicle

Prashant Iyengar, Sneh Vaswani, Chintan Raikar, Shivendra Singh, Anaykumar Joshi, Amit Kumar, S Krishna Savant, Mihir Gupta, Satwik Kottur, Hardik Godara, Sant Kumar, Nilesh Kulkarni, Rakesh Kumar, Akash Verma, M. V. Deepak, Dinesh Kumar, Suryapratap Babar, Tushar Sharma, Devyesh Tondon, Anshuman Kumar, Sanidhya Gupta, Kunal Tyagi

Indian Institute of Technology Bombay

Abstract

Matsya 2.0 is the second installation of the Matsya series of Autonomous Underwater Vehicles developed by the AUV-IITB Team to compete at the International Robosub Competition 2013. Based on feedback from visual, inertial, pressure and acoustic sensors, the vehicle is capable of localization and navigation to perform pre defined tasks of identifying objects, shooting targets, dropping markers and robot manipulation. The second iteration by the team has led to significant improvements along verticals of mechanical, electronic and software subgroups. .

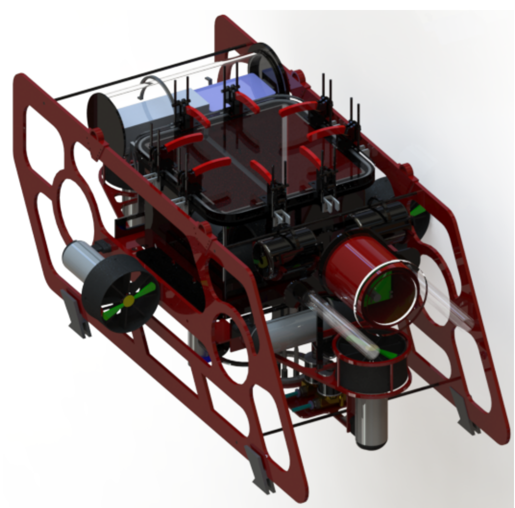

Figure 1: Matsya 2.0

1 Introduction

Water bodies around the globe cover around 70% of Earth's surface area. Majority of this area is still not mapped and is uncertain and this acts a big source of motivation for focussing on underwater robotics. Autonomous Underwater Vehicles (AUV) have opened up a whole new dimension of unmanned applications along the great depth of the oceans. AUVs are currently being used for civilian, defence and commercial applications. These include search and rescue operations, surveillance, detecting faulty pipelines, off shore mining etc. Man has not yet been able to understand the deep waters and what lies beyond; AUVs offer a promise of allowing us to explore.

AUV-IITB is a group of 22 students at IIT Bombay, eager to take on the challenges thrown by the underwater environment. The team works along three frontiers: Mechanical, Electronics and Software, with each of the sub divisions working as a closely knit group. Matsya 2.0 has been designed and developed in a year long process beginning August 2012. The vehicle, weighing just 24 kg, is designed to operate at a maximum depth of upto 40 feet, with an endurance of 1.5 hours.

2 Mechanical

The Mechanical system of Matsya 2.0 is more complete and modular compared to its predecessor with separate enclosures for electronics, battery pod, cameras and also actuators for shooting torpedoes, dropping markers and gripping objects. Newer materials like carbon fiber, ceramic wool, polyurethane rubber have been used to make the vehicle lightweight and robust. While designing the vehicle, a lot of thought has been given to the accessiblity of the different enclosures and attachments. The vehicle has been designed to be dynamically stable along the roll and pitch axes. Weight optimization of the vehicle has been done using rigourous analysis on ANSYS, without compromising on the robustness of the vehicle.

2.1 Hull

Main Hull is a water tight region to host most of the electronic components except the pressure sensor board which is kept in separate enclosure with the pressure sensor. The focus of the design has been robust waterproofing, ease in assembly and disassembly and efficient heat sinking. Main hull of Matsya is cuboidal in shape with dimensions 281 x 276 x 174 mm, fabricated from Aluminium 6061-T6 alloy with acrylic end cap at the top. All the electronic boards are assembled together on a acrylic rack and the wires pass through guides attached to the interior walls of the hull.

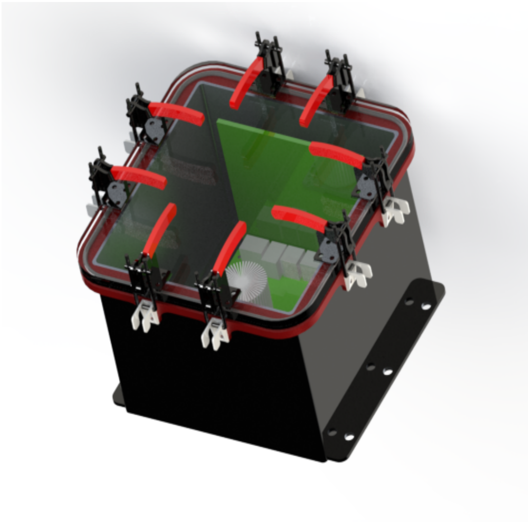

Figure 2: Main Hull

Al 6061-T6 was preferred as the material for the hull body because of its good thermal conductivity, high strength, non-corrosiveness and economic feasibility among other aluminium alloys and materials. Acrylic end cap at the top provides transparent interface for visual detection of water seepage and viewing electronic displays and indicators. The removable end cap becomes the most likely region for leakage. The team experimented with different end cap designs and developed an optimised light weight flange which is welded over the hull and tightened to the hull with an acrylic endcap using pull action latches. Nitrile rubber O-ring is sandwiched in the groove between the flange and the endcap to seize the passage of any liquid into the hull. Round edge of the flange and depth of the groove is designed to keep the O-ring in a relaxed position and ensure optimum compression of the O-ring.

Separate enclosures are made for batteries, pressure sensor board, bottom and front camera, to introduce modularity and flexibility to the system. The team has designed and fabricated the underwater penetrators for routing connections between different waterproof enclosures.

2.1.1 Latches

Pull action toggle latches are fixed over acrylic endcap using threaded inserts to squeeze the O-ring sandwiched between the endcap and the hull body. A lock is designed using E-clip and spring which is mounted on the latch to avoid accidental opening of the latch. The upper bolt is pulled against the spring force to open the lock.

2.2 Frame

The frame of Matsya is responsible for providing a rigid structure to the vehicle. There are many peripherals that the frame needs to house; the positioning and mounting of these peripherals was done strategically to develop a bottom-heavy, open-frame design which exhibits high symmetry, modularity and stability. Since the vehicle operates at low speeds (maximum speed is 0.5m/sec), a closed frame design does not offer a significant advantage over an open frame design. Moreover, an open frame structure ensures easy and fast accessibility and monitoring of any peripheral on the vehicle. To make the vehicle dynamically stable, the position of the peripherals have been choosen so as to align the Center of Buoyancy (COB) and the Center of Mass (COM) vertically some distance apart; with COM lying below COB to obtain an ideal bottom-heavy configuration with natural stability.

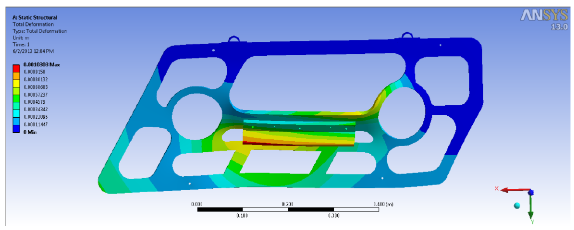

Figure 5: FEA analysis for structural deformation in Delrin Frame

During the development period, different aluminium alloys and several commercially available structural polymers were tried and tested in order to finalize the materials. The design consists of an exterior frame (made of delrin) which supports the interior frame (made of aluminium 6061- T6) and also plays the role of shrouding.

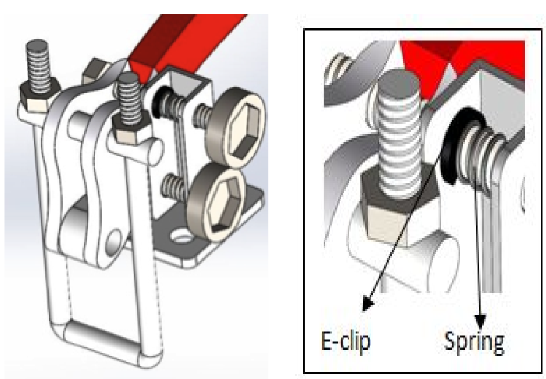

Figure 3: Pull action toggle latches with mounted lock

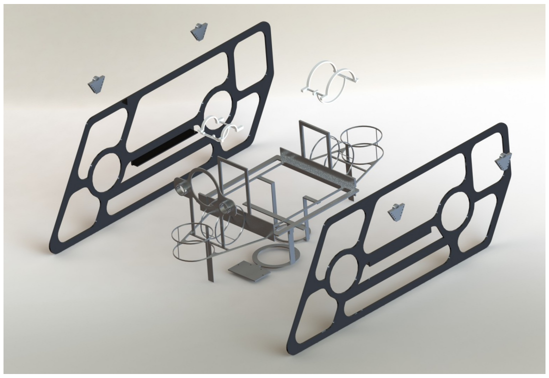

Figure 4: Frame

The parts of the frame were analyzed in ANSYS employing Finite Element Method (FEM) and fabrication was done using CNC machining.

The vehicle has been designed to navigate along 5 degrees of freedom. Six seabotix thrusters were used to allow control over pitch, yaw, surge, heave and also sway (which is a new addition in Matsya 2.0). The dynamic stability of the vehicle also depends on factors such as centre of drag and external forces. The centre of drag determined by the centroids of the effective surface areas of the vehicle, was aligned with the plane of the thrusters to prevent undesirable pitch motion which increase dynamic stability. The surge and sway thrusters are strategically placed to provide optimum yaw control and compactness in vehicle design. Sway thrusters and surge thrusters are placed symmetrically, as close as possible to the center of gravity.

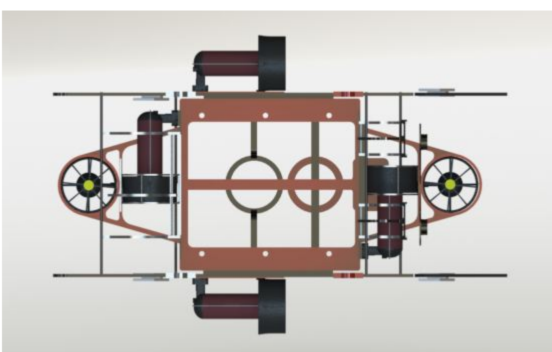

Figure 6: Thruster Positioning

2.3 Actuators

- **Torpedo**: torpedo is made by using ABS plastic rapid prototyping. A small brass rod is inserted axially in the head of torpedo to gain stability and make the torpedo neutrally buoyant. After variuos design iterations, we decided to keep the fins tilted to a 10 degree angle to gain maximum linear traversing stability. Compressed gas at 100 psi is used for its actuation. Body of torpedo consists of a combination of hemispherical front and parabolic cone back. Slenderness ratio has been kept as 5.9. A slot has been provided at the rear end of the torpedo to press fit the air tube with it.

- **Gripper**: gripper arms are machined out of a 3mm aluminium sheet using CNC machining and are actuated by 12 Volts DC solenoids with 15mm stroke length. The shape of the gripper is designed as a hook which can be easily actuated and is normally closed.

- **Marker Dropper**: 12 Volt DC solenoid with 5 mm stroke has been used for dropping glass marbles. The marker dropper system uses a single acrylic tube for holding two markers (marbles). Two solenoids are placed on top of each other and the two markers are placed above and below the upper solenoid.

Figure 7: Torpedo

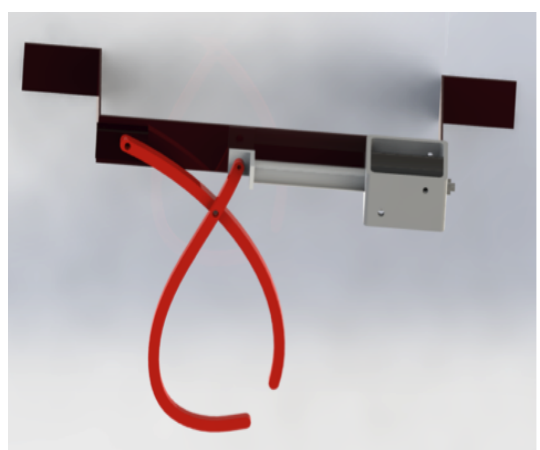

Figure 8: Gripper Assembly

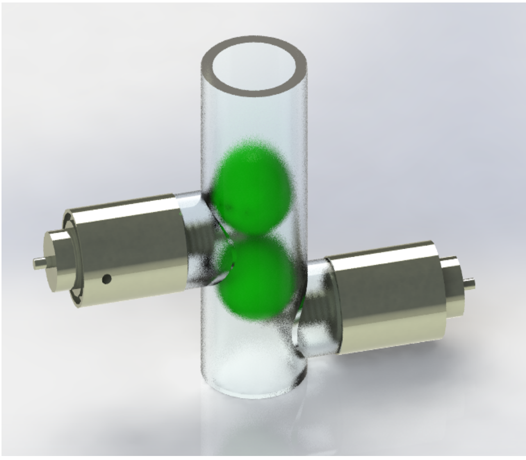

Figure 9: Marker Dropper Assembly

3 Electronics

The electronic system architecture of the vehicle has been designed allowing the software system to achieve optimal control of the vehicle with ease and robustness. Besides scalability in future, this architecture emphasizes on prominent work division while ensuring efficient power distribution. Majority of the boards are designed and populated in-house to achieve the mentioned objectives. All the microcontrollers on the system have been separated out of the main electronics board using microcontroller caps. This approach provides the ease of microcontroller replaceability, off-board microcontroller programming and accumulating the same number of components in much less area. The various processing platforms have been chosen according to the basic needs of sensor data acquisition, controls and power management.

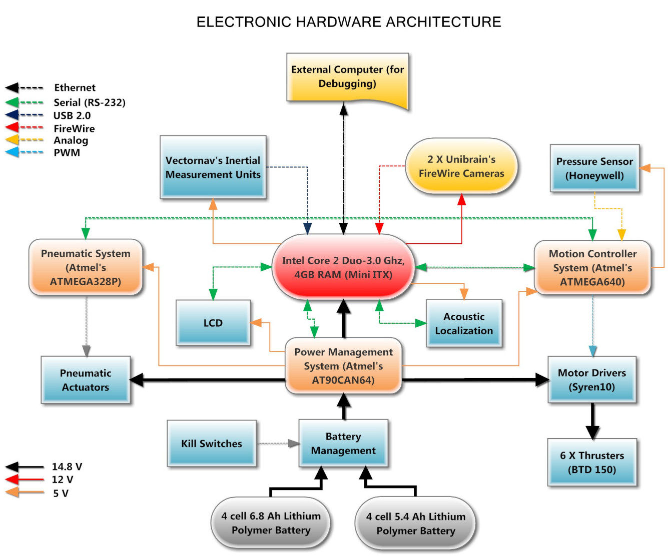

Figure 10: Electronic Harware Architecture

3.1 Electronic Subsystems

3.1.1 Single Board Computer

The vehicle uses Axiomtek's SBC86860 Mini ITX motherboard with an Intel Core 2 Duo Processor clocked at 3.0 GHz and 4GB of RAM. A 32 GB flash drive is used for software storage and data logging. With a compact 6.7" x 6.7" size, this SBC was chosen considering its rich I/O functionality, low power consumption and the new level of performance in image processing. It communicates and commands the motion controller, acoustic localization unit and power management systems serially as per the needs of the vehicle in various tasks.

3.1.2 Motion Controller :

Motion controller system (MCS) dynamic control of the vehicle. As per the setpoints decided by the SBC it executes the closed loop control algorithms providing the desired PWM outputs to the thrusters. The entire system can operate at 5V logic level as well as 3.3V. If a high performance control algorithm is to be implemented, we can conveniently swap it with a high speed low power 3.3V controller without reiterating the design process. Pressure sensor calibration and linearization is an extension of MCS's functions.

Other than running the control algorithms, it also activates or de-activates the pneumatic system as and when required and also provides commands for specific actuations via serial interface to the pneumatic controller. Serial interface for a separate LCD has also been facilitated for debugging the algorithms running on Motion Controller.

Figure 11: Stack of Electronic boards in Matysa 2.0

3.1.3 Pneumatic System :

This system provides the ease of separate maneuverability of the pneumatic actuators as required without affecting other sub-systems. It facilitates a separate path for the large currents to be drawn from the battery when switching the six pneumatic valves separately.

3.1.4 Acoustic Localization Unit :

The hydrophones are arranged in Ultra Short Base Line (USBL) arrangement on the vehicle. Every channel is signal conditioned before been processed to evaluate bearing of the pinger with respect to the vehicle. The signal conditioning involves preamplification followed by bandpass filtering before proceeding towards digitization and bearing calculation. The analog conditioning is on a mixed signal layout which is independent from the digital controller for modularity purposes. The digital controller interfaced to the analog blocks of every channel can conveniently vary the gain of the preamplification and also adjust the centre frequency and quality factor of the switched capacitor bandpass filters. This enables the system to operate at different pinger frequencies. The layout for the analog conditioning of all channels has been done on a 1inch radius PCB (12). Some of the salient features of the design are low noise, offset nulled preamplifiers permitting gains till 600 for the pinger frequencies at the competition. The system provides signals in the dual supply (+5V) range. The system can tone down the signals to 3.3V if the digital controller is 3.3 tolerant. Some work has also been done to denoise the signals for improvement in localization accuracy which will be used in future installations of Matysa.

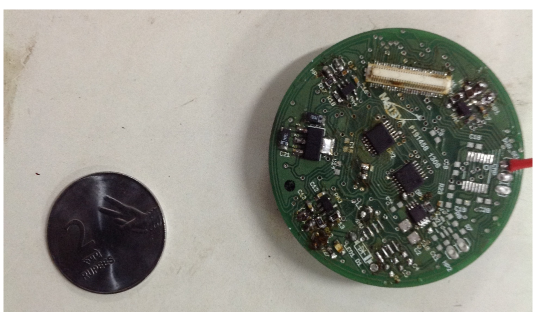

Figure 12: Acoustic Localization Board

3.1.5 Power Management :

The power infrastructure of Matsya 2.0 incorporates various features for smart regulation of numerous loads on the vehicle. Two Thunder Power Lithium Polymer 4 cell batteries having 6.8 Ah and 5.4 Ah capacities are used to provide power for components of the entire vehicle. The higher capacity battery is used to power up the electronics due to the continuous current consumption by the Single Board Computer whereas the latter is used for powering inductive loads such as thrusters and pneumatic actuators. This configuration isolates motor noise from the electronics of the vehicle, at the same time ensuring optimal power for both subsystems. The major power channels operate at +14.8V, +12V, +5V and +3.3V where the lower voltages are generated with the help of appropriate switching regulators. The entire power system is handled by an Atmel's 8 bit AT90CAN64 microcontroller which keeps track of every channel for characterization of sensors via current measurement, data logging to a micro SD card for time stamping of power consumption, detection of any faulty lines and thereby switching off the corresponding channels if necessary and updating critical parameters to the Single Board Computer for diagnostics. Additional features include RGB leds for battery status, extra power lines for scalability and JTAG interface from debugging perspective.

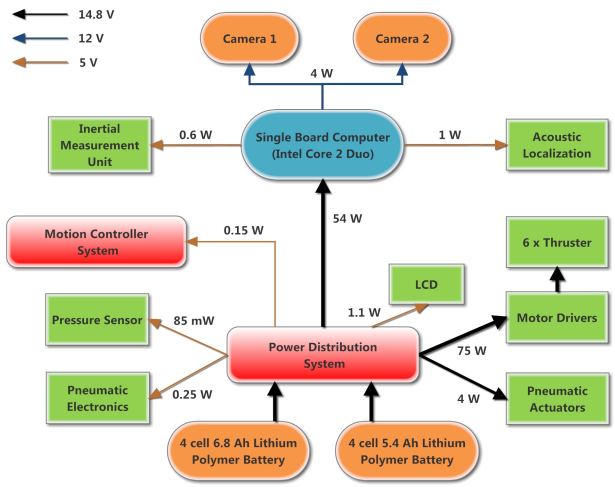

Figure 13: Power Distribution

3.2 Sensors and Actuators :

- Camera: The vision framework takes input from two Unibrain Firewire Cameras (Fire-i™ Digital Board Camera) mounted in the front and bottom of the vehicle.

- IMU: VectorNav's VN 200 is used to estimate the orientation of the vehicle. Certain features like high accuracy measurement over the full operating temperature range, negligible sensitivity to supply voltage variations and temperature dependent hysteresis made it appropriate for the requirements of the vehicle.

- Hydrophone Array: Reson TC 4013 hydrophones are used to estimate the bearing of the vehicle relative to the acoustic pinger. These are miniature hydrophones with very high sensitivity, ideal for measuring sound across a wide range of frequencies.

- Pressure Sensor: US300 Analog Pressure Sensor by MEAS is used to get the absolute pressure value at a given depth. This analog value is read by the Pressure Board, a separate extension of motion controller system, which estimates the depth of the vehicle using a near-linear relation and communicates the depth value to the Motion Controller serially.

- Thrusters, Actuators and Drivers: Six BTD150 thrusters offered by Seabotix have been mounted on the vehicle frame. Each thruster consumes around 80 watts to deliver a thrust force of 12N. SyRen 10 Regenerative Motor Drivers from Dimension Engineering have been used to drive each of these PM DC thrusters. Even 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. A separate motor driver board has been dedicated to facilitate user friendly support of motor drivers giving information on the working of individual motor driver.

4 Software

The software stack has been built on top of the **Robot Operating System** (ROS), developed at Willow Garage. Also, the **Gazebo** simulator has been used to partially test the software stack before deploying the software on the real vehicle. The software system is implemented as a single ROS stack with different packages for managing vision, navigation, hardware abstraction etc. The major design goals of the software stack were extensibility, abstraction and robustness. Also, based on our previous year's experience, the need for a highly flexible debug platform was found to be necessary. ROS helped us to meet our design goals and keep our software modular, with different duties clearly demarcated and distributed into various processes (nodes). The core software however, has been kept generic enough so that it can be easily plugged out of the present framework and plugged into a different robotics framework. The broad layers of the software stack are as follows:

- **Firmware** : The lowermost layer running on the microcontrollers.

- **Middleware** : Responsible for Inter Process Communication and Hardware Abstraction. The middleware helps abstract out the microcontrollers and present them as ordinary processes running on the SBC. Each hardware peripheral connected to the SBC is abstracted out as an individual ROS node. Inter Process Communication is handled entirely by ROS using messages and services. This maintains the modularity of the system and provides a clean API for communication.

- **Processing Layer** : Responsible for processing sensor information (such as videos, and IMU data) and providing meaningful data (such as the center of buoys visible in the video)

- **Application Layer** : Uses data from the processing layer to do useful things. The application layer contains the debug interfaces and the mission planning nodes

4.1 Localization

The objective is to autonomously navigate an AUV using only visual and inertial sensor measurements (the hydrophones have been used only in one task). The localization approach involves solving a local position estimation problem with respect to the environment, given a rough initial state of a relatively static environment. As visual sensing is highly degraded in an underwater environment, localization using visual feedback is a challenging task. In such a scenario, active localization by staying closer to landmarks and updating the position belief is the method we preferred. This helps in compensating for the drift in the measurements inherent to inertial measurement units. The IMU drift has been tackled through dynamic recalibration of the inertial unit using visual feedback.

4.2 Inter Board Communication

The communication stack is responsible for enabling data and command transfer among six different boards on the vehicle. The boards are connected to each other using UART / RS232 links in a tree like structure as shown in Fig:14. All the boards are mutually connected to each other (either directly or indirectly) and this allows data to be transfered between any two boards in the system.

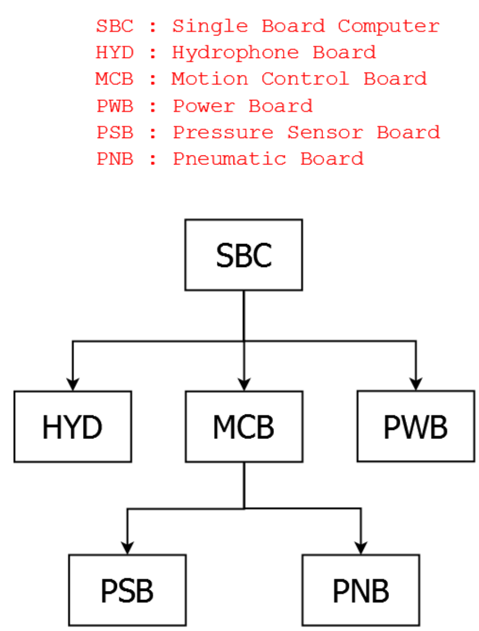

Figure 14: Interconnection Amongst the Electronic Boards

The communication between any two individual boards is based on a "ping and reply" system. A board closer to the root of the tree initiates the communication with boards connected to it. For example, the SBC would initiate the communication with MCB. The communication between any two boards is always dual. If a board A starts communication with B (by sending a stream of fixed amount of bytes), then B would always reply back the same number of bytes containing data relevant to A. As an example, the SBC can get the pressure sensor data from the PSB. To do so, the SBC would ask the MCB to get the pressure sensor data from the PSB and transfer it to the SBC. All the data transfer has been made robust using Cyclic Redundancy Check.

4.3 Mission Planner

The mission planner sits in the application layer of the software stack. The mission planner has been implemented using a finite state machine. The planning system consists of four ROS nodes (basically four processes):

- Planner
- Transition State
- Scan State
- Execution State

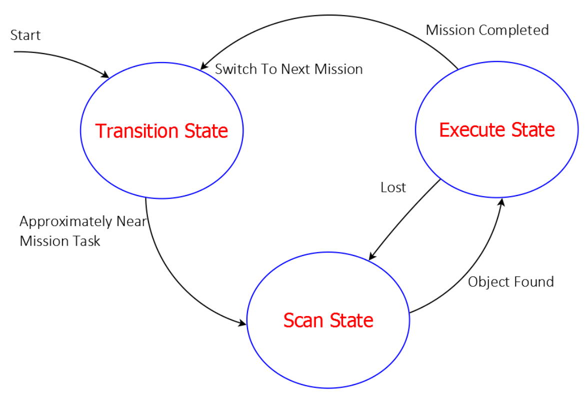

Figure 15: Finite State Mission of the Mission Planning System

The vehicle can be in one of the three states at any instant. To complete a task, the vehicle always starts in the transition state. By knowing some rough information about the location of tasks with respect to each other, the vehicle is manually given a rough map of the arena. While in the transition state, the vehicle simply executes a certain set of control commands to move from one task to another (dead reckoning). Next, the vehicle moves into the scan state to search for relevant objects (buoys, planks etc). In this state, the vehicle wanders around in a small area around its present location, to get an acceptable quality of visual feedback. Once the vehicle finds the object, it moves into the execution state to complete the task (hit a buoy, align wrt to the plank, shoot a torpedo etc). While in the execution state, if suppose the vehicle looses its way and the relevant object goes out of view of the camera, the vehicle again enters the scan state state to again find objects around itself. The planner node manages this Finite State Machine and makes the vehicle move from one state to another. The planner also implements a time-out mechanism for moving into a new task if the present task has not been completed for a long time. On an implementation note, the planner system has been implemented using the ROS actionlib library.

4.4 Debug Interfaces

This year, a lot of emphasis was given on providing a robust debug interface during the vehicle testing period. Three primary debug interfaces are available:

- **Electronic Board Interface** This interface further has three components:
  Motion Control Interface: This provides an interface to view and change relevant parameters on the machine related to the motion dynamics; such as PID parameters, control setpoints etc. The interface also allows the user to view various machine variables, such as the present depth, motor PWM values etc.
  Power Board Interface: This interface helps monitor machine status like its battery levels, the status of the kill switches and thrusters.
  Pneumatic Interface: This interface allows the user to fire torpedos, drop markers and turn on/off the grippers.

- **Vision Interface**: This interface takes care of the various vision related parameters which are required to be tuned depending upon the lighting conditions.

- **Map Interface**: This interface provides a drag and drop interface to create a rough map of the arena. The output of the map is used by the mission planner to help navigate from one task to another.

4.5 Vision

The vision system is probably the most important subsystem of the software stack. Since the vehicle's navigation stack heavily depends on visual feedback, robust image processing algorithms are required to be implemented.

4.5.1 Problems Faced

The objects associated with different tasks are identified using color or shape information. However performing color or shape analysis on raw images is difficult due to various degradations observed in the underwater environment. The main problems faced in underwater image processing is low visibility, blue or green color cast, poor contrast, varying illumination conditions, brightness artifacts, blurring and noise.

Low visibility is because the light is attenuated exponentially as it travels through water. The visibility range for the camera used is around 10-15m in clear water and about 4-5m in turbid water. Under-water images are dominated by blue-green color which leads to low contrast images. The illumination of images is drastically affected by changing ambient lighting conditions and also by varying depth of the vehicle. Brigthness artifacts are often observed near the water surface and near the floor due to interaction of sunlight at the surfaces.

4.5.2 Techniques Used

To handle varying illumination of images, an auto exposure algorithm has been implemented to dynamically change the exposure of the cameras. The image enhancement algorithms helped remove the water color cast and provide images with good contrast and introduce minimal artifacts. We developed a novel contrast stretching algorithm that uses water color and illumination information to process images.

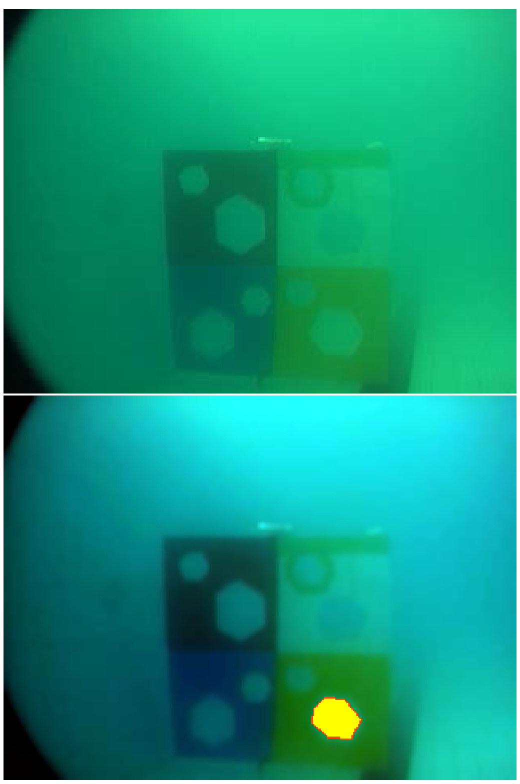

Figure 16: Enhancing Underwater Images

The color detection is performed in HSV Color Space as it provides some robustness to illumination changes. We developed a novel edge based object detection technique to perform edge detection at different edge thresholds and impose loose geometric constraints to identify objects and corresponding edges. Due to poor lighting conditions it was not always possible to detect objects using color analysis.We used connected component analysis to indentify coherent regions in images. Loose geometric and color constraints are imposed on connected regions detected, to detemine the object of interest. Software modules for image processing are built using Intel's OpenCV Library.

5 Acknowledgements

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 SA 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. Leena Vachchani and Prof. Hemendra Arya 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, Oil and Natural Gas Corporation Ltd. (ONGC), Menzel RoboVision, PCB Power, Zephyr Toys, Seabotix, SolidWorks, Unicorn, HNR Johnson.

References

[1] "*ROS: an open-source Robot Operating System*", Morgan Quigley , Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, Andrew Ng

[2] "*Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator*", Nathan Koenig, Andrew Howard

[3] "*A Survey and Comparison of Commercial and Open-Source Robotic Simulator Software*", Aaron Staranowicz, Gian Luca Mariottini

[4] "*Probabilistic Robotics*", Wolfram Burgard, Dieter Fox, Sebastian Thrun

[5] "*Contrast Limited Adaptive Histograph Equalization*", Zuiderveld, Karel

[6] "*Efficient Graph-Based Image Segmentation*", Pedro F. Felzenszwalb, Daniel P. Huttenlocher

[7] "*Simplest Color Balance*", Limare, Nicolas, Jose-Luis Lisani, Jean-Michel Morel, Ana Belén Petro, and Catalina Sbert

[8] "*Shades of Gray and Colour Constancy*", Finlayson, G. D, Trezzi. E

[9] "*Automatic Color Enhancement {(ACE)} and its Fast Implementation*", Pascal Getreuer

[10] "*Enhancing Underwater Images and Videos by Fusion*", Cosmin Ancuti, Codruta Orniana Ancuti,Tom Haber, Philippe Bekaert

[11] "*Design Modelling and Control of an Autonomous underwater vehicle*", Louis Andrew Gonzalez

[12] "*Design Aspects of Underwater Intervention Systems*", Hawley, J. Nuckols, M. Reader, G. Potter,I.

[13] "*Development of an Autonomous Underwater Vehicle in an Interdisciplinary Contex*", Bernhard Gerl,Technische Universität München