The Stingray AUV: A Small And Cost-Effective Solution for Ecological Monitoring

Christopher Barngrover, Thomas Denewiler, Greg Mills and Ryan Kastner
Department of Computer Science and Engineering
University of California San Diego
La Jolla, California 92093-0404

***Abstract*—Underwater vehicles have recently become more useful in ecological monitoring, largely in part to advanced processing capabilities enabled by modern computers. Most underwater vehicles are torpedo shaped and non-holonomically controlled, which makes them efficient, but they lack precise maneuverability. Some cube-shaped vehicles are used when more exact navigation is necessary; however they cannot take advantage of gliding motions and hydrodynamic lift as their vehicles have a large amount of drag. The Stingray Autonomous Underwater Vehicle (AUV) is a compact, lightweight AUV with a unique design implementation. The hull of the Stingray is a carbon fiber shell with a biomimetic design reminiscent of its ocean-dwelling namesake. This streamlined profile provides very low drag and allows the vehicle to glide through the water. The Stingray also uses a unique propulsion system, combining three vertical thrusters on the wings and tail for roll and pitch with two Voith-Schneider propellers mounted underneath for yaw and surge. In addition, these two propellers provide the ability to strafe, allowing the vehicle to move with six degrees of freedom. This enables the Stingray to easily maneuver at slow speeds and hover in a similar fashion to a helicopter, while also being able to take advantage of the lift generated by its wings to glide like a fixed-wing aircraft.**

I. Introduction

Underwater vehicles have many applications in both remotely operated and autonomous implementations. Most such vehicles are torpedo shaped, while others are cube shaped [1]. The torpedo shape is efficient in surge, but lacks precise maneuverability. The cube shaped vehicles allow for more exact navigation, but they do not move efficiently due to their shape, causing a large amount of drag [2]. Furthermore, most underwater vehicles in use are quite large and heavy, making deployment and recovery difficult for small teams. Finally, the norm for underwater vehicles is an extremely high price, making it unavailable for many research efforts.

The Stingray autonomous underwater vehicle (AUV) was designed with the above difficulties in mind. The vehicle has a carbon fiber shell with a biomimetic design, allowing for efficient movement through the water. The vehicle uses a unique propulsion system that includes two Voith-Schneider propellers for controlling yaw and surge, as well as the ability to strafe. With this propulsion system the Stingray has precise maneuverability at slow speeds and the capability to hover. Since the Stingray currently assumes an environment with man-made cues for vision based navigation, it is readily operational with low cost cameras and comparatively cheap to other vehicles.

This paper is organized as follows. The background on the Stingray is given in Section II. Section III describes the hardware solution in the subsections of hull, thrusters, electronics, and sensors. Similarly, Section IV discusses the software solution in terms of the responsibilities of the four main modules, including controls, user interface, vision, and planning. The future plans for Stingray 2.0 are described in Section V, which has subsections for the intended hardware and software improvements. The conclusion is then presented in Section VI.

II. The Stingray

Over the past several years, the San Diego iBotics group, which consists of students from several San Diego area community colleges and universities, have built an AUV to compete in AUVSI & ONR's Annual International AUV Competition. The requirements of the competition are to develop a vehicle capable of completing a number of underwater challenges that focus on navigation, vision, sonar and environment manipulation. The approach that iBotics takes to this competition is not only to compete as a student organization, but also to shape the future of underwater vehicles by working to create the next generation of AUV technology. The result of these efforts is the Stingray AUV as shown in Figure 1.

(a) Solid model view. (b) Stingray in water with diver.

Fig. 1. The Stingray AUV.

III. Hardware Solutions

The hardware solutions for the Stingray vehicle were chosen to minimize weight and cost. The first step was the design of the hull, which should be lightweight, strong, and hydro-dynamically efficient. The hull was then populated with a propulsion system, watertight electronics enclosures, and external sensors. This configuration provides the necessary feedback for environment perception and vehicle control.

A. Hull Design

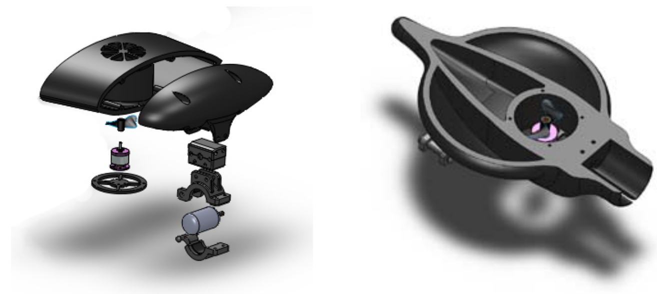
The hull of the Stingray is a small, lightweight, carbon fiber shell with a biomimetic design intended to mimic the shape of the cartilaginous fish. This hull was designed and handcrafted as a mold, which was used to lay and set carbon fiber for the creation of resulting hull. The hull has since affected almost every design decision on the vehicle by serving as both a rigid size constraint and an inspiring standard of quality and craftsmanship. Once the hull was fabricated, a seamless access hatch was installed in the top for mounting electronics enclosures, sensors, and thrusters. Figure 2 shows the original hull and a finished version with the lid off.

(a) Post carbon fiber fabrication. (b) Finished design with lid open.

Fig. 2. The Stingray Hull.

The streamlined profile of the hull provides very low drag and allows the Stingray to glide almost effortlessly through the water. The hull of the Stingray is unsealed, allowing the main cabin to flood when submerged. This keeps the volume of the pressure vessels to a minimum and eliminates the need for extra ballast, which would increase the vehicle's dry weight. This means that the maximum operational depth of the Stingray is solely dictated by the pressure housing of its electronics.

The addition of the propulsion system required the carbon fiber shell to be modified. Modular wing and tail caps were designed with two requirements in mind. Firstly to complete the body hydrofoil and secondly to vertically mount the wing and tail thrusters. The caps were also filled with a closed cell polyurethane foam to increase the buoyancy and balance the vehicle. The wings were created with a 3D printer and coated in epoxy to prevent water logging. This allows different caps to be rapidly created and gives the Stingray the adaptability to trim buoyancy, install new thrusters and experiment with different flow surfaces. The buoyancy is trimmed to make the vehicle slightly positively buoyant and to naturally level pitch and roll. This allows the vehicle to surface if power is lost to the thrusters and to minimize the amount of continuous corrective action required by the vertical thrusters to stabilize vehicle orientation.



(a) Exploded view of the wing. (b) Cross section of the tail.

Fig. 3. 3D printed wing and tail.

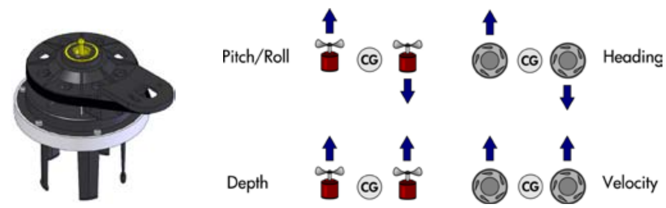
The improved buoyancy results in a more statically stable vehicle that stays level without actuation and is slightly buoyant so that it rises to the surface if motor power is lost. The wings and tail were each designed using 3D design software and then printed using a 3D printer. The resulting plastic was then painted in epoxy so that the plastic does not retain water. Each piece is printed with mounting brackets for the transducers, which are used for triangulating the location of an acoustic pinger for navigation purposes. The 3D designs of the wings and tail are shown in Figure 3.

B. Propulsion System

The Stingray uses two types of thrusters for propulsion. Two Voith-Schneider [3] propellers are mounted underneath the vehicle and are responsible for controlling yaw and surge. Three simple thrusters are inset into the wings and tail in a vertical orientation. The wing thrusters control roll and depth while the tail thruster controls pitch.

The Voith-Schneider propellers, as shown in Figure 4 (a), are a novel approach to underwater propulsion. By utilizing two of the Graupner 2358 Voith-Schneider assemblies in a counter rotating configuration, the Stingray is able to move in any relative direction in the horizontal plane while simultaneously maintaining any arbitrary heading. The actuation of the Voith-Schneider propellers can be seen in Figure 4 (b). This means that with only two of these propellers three degrees of freedom are achieved; the Stingray can move forward or backward, strafe left or right, turn on a dime or any combination thereof. It is this unique propulsion system that enables the Stingray to hover near objects of interest or even rotate around an object at a fixed radius while keeping the object in front of the vehicle. The ability to strafe allows the Stingray to keep any of its cameras pointed in an arbitrary direction while the vehicle is in motion.

The vertical thrusters are used by the Stingray to control pitch, roll, and depth. Each thruster contains a Scorpion 22mm brushless motor with an attached 2-blade RC boat propeller. Brushless DC motors allow us to conformally coat



(a) Voith-Schneider propeller. (b) Thruster Actuation.

Fig. 4. Propulsion Mechanisms.

the windings for saltwater hardening and offer high torque density and high efficiency. The high torque density lets the thrusters occupy a smaller size and weight for the same amount of available thrust. The actuation of these thrusters can be seen in Figure 4 (b).

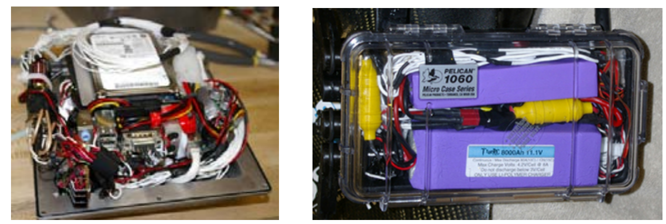
The propulsion system on the Stingray allows for six degrees of freedom in movement. This enables the Stingray to easily maneuver at slow speeds and hover in a similar fashion to a helicopter, while also being able to take advantage of the lift generated by the wings to glide like a fixed-wing aircraft. The Stingray uses considerably less energy than a vehicle without flight dynamics when operating at speeds of 1-2 m/s. The propulsion system allows for fine movements giving the Stingray the capability of moving into areas that other underwater vehicles could not navigate, e.g., shallow water coral reef lagoons.

Due to its freedom of motion, the Stingray is very effective at discovering and tracking objects. Most underwater vehicles are torpedo shaped and non-holonomically controlled, potentially requiring multiple passes of a region to adequately identify an object of interest. In contrast, the Stingray can easily perch-and-stare, hover and/or strafe near an area/object of interest, allowing long observation times. This allows for data communications with a fixed wireless transmitter (e.g., to download data from an underwater wireless sensor) as well as monitoring and mapping underwater geology and ecosystems. It has the potential to get close to objects of interest while maintaining a safe range so as to not harm those objects or the vehicle itself.

C. Enclosures and Electronics

There are two pressure vessels on the Stingray: the computer box and the battery box. These house the electronics and batteries that control and power the vehicle. These pressure cases solely dictate the potential depths of the Stingray as the hull is fully flooded.

The computer box is a resealable aluminum housing design as shown in Figure 5 (a). The main computer is a PC104-based Kontron ETX-CD Intel Core 2 Duo with 1.2GHz, 2GB of DDR2 RAM, 100Mbps Ethernet port, and VGA support. Ubuntu Linux serves as the operating system. The Labjack U3-HV5 A/D data acquisition card is used to monitor and collect data from a number of the onboard sensors. A Ocean Server OS-3000 compass is used to determine the attitude and angular acceleration of the vehicle. As the computer box contains the Stingray's most expensive and essential electrical components, we designed and built a custom moisture sensor that indicates if the amount of moisture in the computer box ever reaches a potentially dangerous level.



(a) Computer box without lid. (b) Closed battery box.

Fig. 5. Electronics enclosures.

The second pressure vessel is used to house the power components as shown in Figure 5 (b). The battery box is constructed from a modified Pelican case. A 4 Amp-hour 14.8 V lithium polymer battery is used to power the main computer, and its peripherals including the data acquisition card and sensor array. A second 8 Amp-hour 11.1 V lithium polymer battery is used to power the propulsion systems. A Scorpion Commander 45 Amp electronic speed controller is used to manage power to each of the motors on the Voith-Schneider propellers. Each of the vertical thrusters is driven by a Castle Sidewinder 25 Amp electronic speed controller. The battery box is designed such that it can be easily opened and closed multiple times, a requirement driven by the amount of charge each battery can hold. This also reduces the amount of maintenance required in the field.

D. Control and Perception Sensors

The Stingray also includes a number of sensors for control and perception. The control sensors include a compass for orientation as well as a pressure sensor for depth. The perception sensors are cameras for detecting man-made objects and passive sonars for locating an acoustic pinger. Both of these perception sensors provide self-localization for navigation.

An Ocean Server OS-3000 3-axis solid state compass, shown in Figure 6, is used to determine the attitude of the vehicle. The compass uses a combination of magnetometers and accelerometers to determine the angles of pitch, roll, and yaw where the latter is used as a compass heading. These values are output from the compass as an ASCII stream via USB.

An SSI Technologies P51-15-S-B-I36-4.5V-R pressure sensor, as shown in Figure 7, is used to measure the depth of the vehicle relative to the surface of the water. The 0.5 - 4.5 VDC output voltage range supplied by the pressure sensor is measured using the Labjack data acquisition card. The pressure sensor output voltage corresponds to a pressure in the range 14.7 - 29.5 psi. This provides a resolution of 2.44mm for the first 10m below the water surface.

Fig. 6. Ocean Server OS-3000 compass.

Fig. 7. SSI Technologies pressure sensor for depth.

A Remote Ocean Systems CE-X-18 underwater camera is used to view objects in front of the vehicle. The vehicle also uses an Inuktun FireEYE underwater camera that faces downwards from the vehicle. The two different cameras are shown in Figure 8. The choice to use two different cameras is based on the desire to test multiple camera options. These cameras provide an analog signal to the frame grabber in the computer box, which pulls images from the camera feed and provides them to the computer.

Fig. 8. ROS and Inuktun underwater cameras.

The sonar system relies on three channels of recorded audio using Aquarian Audio H1a Hydrophones, as shown in Figure 9. The hydrophones are placed on each of the extremities of the Stingray: one under each of the wings and one under the tail. The geometry of the hydrophones forms roughly an isosceles triangle. For the nominal orientation of the sub, the plane formed by the hydrophones is tilted approximately 10° downward from the horizontal plane. This angle of difference between the two planes is neither a strategic choice, nor a disadvantage to the system. The vehicle is currently configured to detect an acoustic pinger through a custom acoustic processing board for localization based on the data received by the three hydrophones.

Fig. 9. The Aquarian Audio Hydrophones.

E. Summary

The goal of the hardware solution for the Stingray was to keep the cost and weight low. The cost was contained by using simple sensors and thrusters and designing a custom hull and electronics enclosure. The Stingray has a dry weight of approximately 10 kg and can easily be carried by one person. This is extremely light compared to other comparable vehicles which could weigh 40kg or more. This weight was kept low by designing the hull to be small and casting it in carbon fiber.

IV. Software Solutions

The software solution for the Stingray was designed entirely from the ground up. The architecture of the system uses individual modules responsible for individual tasks. There are four main modules and then a series of specialized modules. The first main module implemented was the *Nav* module, which is in charge of the controls. Next we implemented a *GUI* module in order to communicate with the vehicle and change settings or gains. The other two modules are the *Vision* module and the *Planner* module. The communication between the modules is a proprietary messaging system using TCP/IP, which allows flexibility but requires extra effort for maintenance of the protocol. A block diagram of the architecture, including main modules and secondary modules, is shown in Figure 10.

A. Controls

The *Nav* module is in charge of the controls for the system. It connects to the two control sensors, the compass and the pressure sensor, in order to estimate a state for the vehicle. There are actually two secondary modules that communicate with the compass directly and the pressure sensor via the Labjack data acquisition card respectively. The *Nav* module communicates over the TCP/IP protocol with the compass module to receive current pitch, roll and yaw estimations. It also receives the pressure level from the Labjack module. This is enough information for the *Nav* module to estimate the orientation and depth of the Stingray.

The *Nav* module can receive a manual target command from the *GUI* module or an autonomous target command from the *Planner* module. The *Planner* module generates the target commands based on perception input from cameras via the *Vision* module or the sonars via the custom sound board.

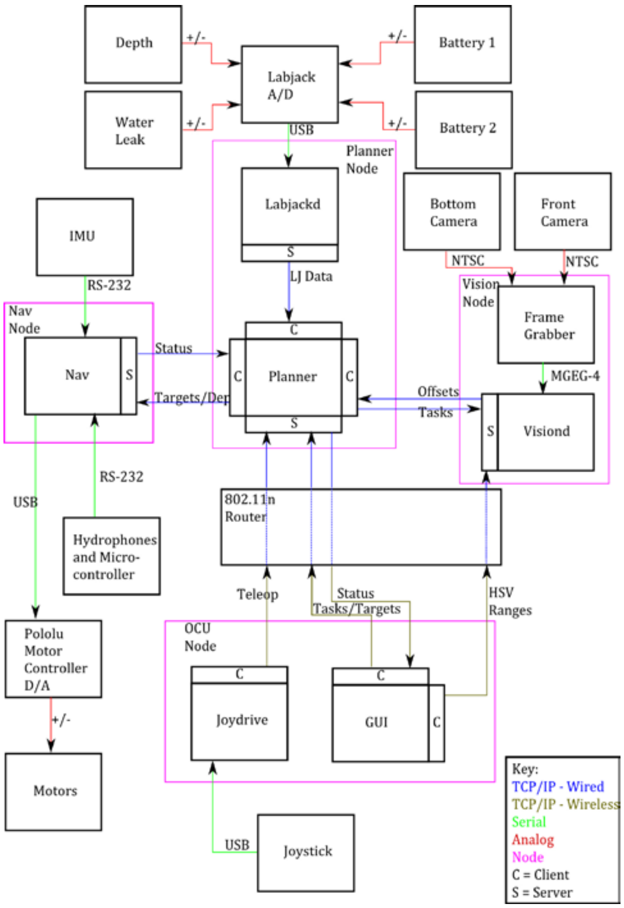


Fig. 10. Block diagram of the architecture for the Stingray.

In order to maintain target values for pitch, roll, yaw and depth, the Stingray implements a number of proportional-integral-derivative (PID) loops. These loops compare the sensor data to given target information in order to calculate an error and apply the appropriate thrust. After a thrust is applied, the loops will continue and update their thrust command based on the new state.

B. Graphical User Interface

To aid in controls development, we created a very simple GTK-based GUI application for use while tethered to the vehicle. The control station works in a client/server configuration with the Stingray, and all communication is done over standard TCP/IP connections, using our proprietary messaging format. The Stingray control station shows the current status of the vehicle. This section includes the compass output of pitch, roll and yaw. The status section also shows Labjack data acquisition outputs, which include the raw pressure data, the water sensor output and the two battery voltages. The control station GUI allows you to alter the current gains for the thrusters for quick tuning of the propulsion system. It also allows you to tune the vision system and manually put the *Vision* module into various object search modes. The GUI can be used to start the *Planner* module in a certain state for testing different stages of the planning capability. Finally, it allows you to manually set a target, turn on or off data logging, and turn on or off the camera displays. Figure 11 shows a screen shot of the shore side GUI.

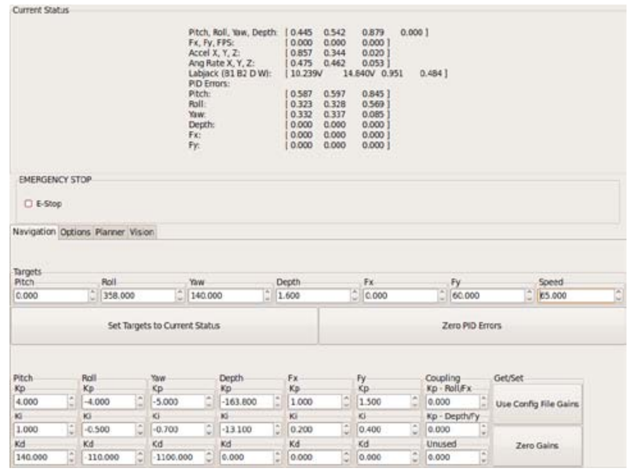


Fig. 11. Screen capture of the GUI for configuration work when tethered.

C. Vision

The Stingray vision system provides relative position information from underwater camera images. The frame grabber captures data from the cameras mounted on the front and bottom of the hull, and those images are then processed using object detection and recognition algorithms.

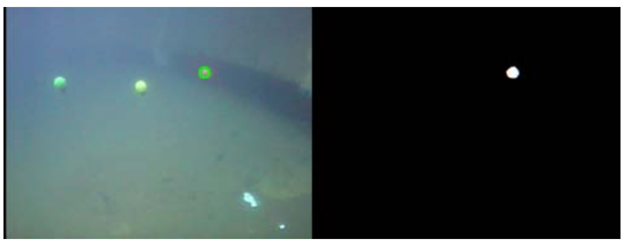
The object detection and recognition algorithm used in the Stingrays vision system uses the OpenCV [4] computer vision libraries and JBoost [5] for optimizing object classifiers. The algorithm is trained *a priori* by labeling example images for positive and negative pixels. The labeled images are decomposed into hue, saturation and value (HSV) and these values are inputs into the JBoost software. The boosting algorithm creates an optimized decision tree via machine learning, which outputs a score when given a single HSV pixel.

In practice, the *Planner* module sends commands to the *Vision* module, instructing it to search for certain trained objects, which can be used as a destination or could act as a path for the vehicle to follow [6]. The captured images from the frame grabber are again decomposed into (HSV) components so that a single coordinate represents all color information independent of lighting fluctuations [7]. Each pixel's HSV is input into the optimized decision tree created by JBoost. The output of the decision tree is a score representing the likelihood of being part of the target object. This score is thresholded to label the given pixel as positive or negative. After all pixels are considered, the result is a binary image as shown in the right side of the images in Figure 12, which shows the detection of three different color buoys based on the given vision task.

To reduce the unwanted effects of image noise, the binary images go through a post processing algorithm, where morphological operations are performed. The first technique applied to

(a) Vision task to find the green buoy.

(b) Vision task to find the yellow buoy.



(c) Vision task to find the red buoy.

Fig. 12. Detection of the location of specific colors of buoy. The right side of the image is the binary result from the algorithm. The left side shows a green circle around the center of the detected object.

the image is opening, the order operations of erosion followed by dilation, which is used to remove noise in the binary image. The opposite technique of closing is used in two iterations to fill in binary objects containing gaps. The image is then smoothed by Median blur with a 7x7 kernel to create smooth edges of the binary objects. Finally, the convex hull algorithm approximates the shape of the binary object with only convex corners, providing more complete binary objects.

Determining the orientation of objects, such as pipes, requires additional calculations. To find the orientation of pipes relative to the vessel, the edges of the object are found from the binary blob using the Canny edge detector. Next, the Hough Transform is used to find straight lines in the binary image. We use the Probabilistic Hough Transform (PHT) due to its ability to combine similar lines with a gap between them [8]. From this, the long edges of the pipe can be found from all possible straight lines. The long edges of the pipe are averaged to create a relative yaw angle, which can be provided to the *Nav* module as a target. An example of the detection of two pipes in an image is shown in Figure 13.

To use the vision data for navigation control, the objects must be localized within the image once they are detected. To determine the relative position of an object, the centroid of the binary blob with the largest area is calculated. This

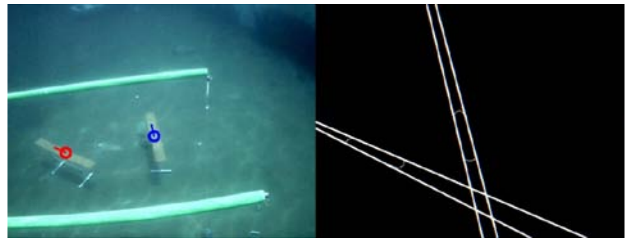


Fig. 13. Detection of location and orientation of two pipes. The right image shows the outline of the pipes from the binary image with the estimated long edges. The left image shows the labeled centers of the pipes and the estimated orientation in red and blue respectively.

calculation can be seen in Figure 12 as green circles and in Figure 13 as red and blue circles. For tasks that require the vehicle to maintain a fixed position relative to the target, the difference between the center of the images and the center of the detected object is used to drive the PID loops.

D. Planner

The *Planner* module is designed to make decisions for the Stingray during autonomous operations. The module manages the *Nav* and *Vision* modules through messages over the TCP/IP connection or the sound board over serial. The *Planner* decides what task is next on the mission and commands the *Vision* module to search for the appropriate object or the sound board to listen for a pinger. The module then manages the responses in order to send target commands to the *Nav* module. In operation, this module requires *a prior* knowledge of the environment in order to understand the mission and the series of tasks.

E. Summary

The block diagram in Figure 10 shows the four main modules of *Nav*, *GUI*, *Vision* and *Planner* as well as the secondary modules, such as *Labjack* or *Joydrive*. The modules in the system communicate with each other via a proprietary messaging system over TCP/IP. The custom messages allow for flexibility, but require extra effort for design and maintenance.

V. Stingray 2.0

A. Hardware Changes

The next generation of the Stingray AUV will maintain its small form and agility but utilize a more adaptable hull. Rather than custom enclosures that fit within the Stingray's slim carbon fiber body, the Stingray 2.0 will boast a reconfigurable chassis capable of accepting different payloads. The ease of taking on different modules will make the Stingray AUV and ideal oceanographic research platform. The hydrodynamic hull will be fabricated as a cover for the modular chassis. This allows us to remain efficient in the water while at the same time making the electronics, sensors, and wiring on the chassis easily accessible. Figure 14 shows a 3D drawing of the concept for the Stingray 2.0 chassis.

Fig. 14. A 3D drawing of the concept for Stingray 2.0. Shown are the chassis, five thrusters, two computer enclosures, a cylindrical battery enclosure on a small junction enclosure, and three camera enclosures.

Stronger commercial thrusters, proven to withstand the corrosive sea environment, will outfit the new vehicle. There will still be one tail thruster for controlling pitch and two wing thrusters for controlling roll and depth. The change will be using two of the same thrusters on the wings to control yaw and surge. The five thrusters are shown on the tail, wings and next two the camera mount in Figure 14.

Wetmateable underwater connectors and cabling provide an efficient means of distributing power and communications throughout the vehicle as well as accommodating new hardware. There will be two main enclosures, one for the controls computer and one for the vision computer. There will also be a junction enclosure as a breakout box for power, including relay switches in order to kill power to the thrusters. All the electronics enclosures will be made from anodized 6061-T6 aluminum to provide both the necessary corrosion resistance and strength. The two computer enclosures can be seen as the wings of the vehicle in the drawing of Figure 14.The junction enclosure is a small box under the cylindrical battery enclosure in the center of the vehicle.

The new Stingray will have upgraded sensors, but will keep costs low by continuing to use simple sensors. The compass will be an upgraded version of the Ocean Server compass and the pressure sensor will remain the same. The cameras will be the main improvement, since the vision processing is are primary source of environment input. The new cameras will have gigabit ethernet connection to provide a digital video stream. Stingray 2.0 will have a mount on the nose with a stereo pair of cameras facing forward and third camera facing down. In an effort to reduce costs and simplify the design both mechanically and for software, identical features will be used whenever possible. For example, all three chosen cameras will be the same as will their enclosures and connectors. The camera mount is shown on the nose of the vehicle in Figure 14.

B. Software Changes

The software will also be overhauled for the change to Stingray 2.0. The main change will be to remove our own proprietary module and TCP/IP messaging scheme for a tested and reliable open source architecture designed for robotic applications. The software suite is called the Robot Operating System or ROS [9], which is mainly developed by Willow Garage. ROS uses the concept of nodes that communicate using TCP/IP just like the current Stingray software, but the ROS implementation is more robust and offers access to software written by many other groups that have been standardized on the ROS architecture. At this time there are well over 100 different repositories from companies and universities in addition to the main ROS repositories. By porting the Stingray software to ROS we can very easily use nodes written by Willow Garage or other groups. Another benefit of using ROS is the ability to use their visualization and debugging tools to monitor the state of the system in real time.

A diagram of the current system is shown in Figure 10. Using the *rxgraph* ROS tool a graph can be generated during runtime that shows all of the nodes that are running as well as the topics where they are publishing their messages. Figure 15 shows what our software system will look like after we port to ROS all of the modules described in Section IV. The difference with the ROS architecture from our current implementation is that each entity in the system is a node, regardless of its importance. The graph in Figure 15 shows that even the compass and thrusters will have their own nodes.

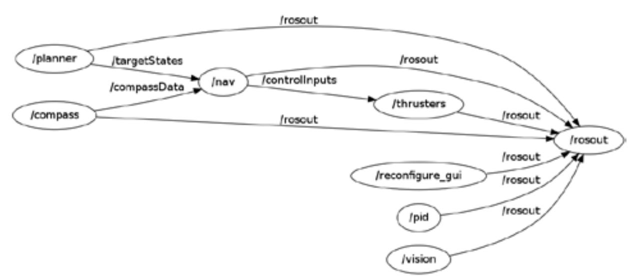


Fig. 15. Graph of Stingray ROS nodes and topics.

VI. Conclusion

This paper outlines the design and implementation of the Stingray AUV, which is a small, lightweight, and cost effective platform. We described the hydrodynamic hull design and how this was the basis for future decisions on the system. A major capability of the Stingray described is the unique propulsion system, which allows the vehicle to move in six degrees of freedom, including the ability to strafe. The Stingray uses a few simple sensors for controls and for perceiving the environment. We also described the proprietary module software system, which uses PID loops for control in the *Nav* module, a *Planner* and a *GUI* module for for providing targets, and a *Vision* module for perception and navigation input based on detection of man-made objects.

This paper also outlines the future plans for Stingray 2.0, which will build on the two major shortcomings of this Stingray platform. One is the tight and barely accessible interior to the hull, which will be improved by making the modular chassis as a frame for the hydrodynamic shell. The other is the messaging system between modules, which will be improved by using a supported framework from the ROS architecture. The Stingray vehicle is already a small and cost effective AUV and the next generation will build on this precedent.

ACKNOWLEDGMENT

The authors would like to thank the SD iBotics Team from past and current years for their dedication to the design and implementation of the Stingray AUV.

REFERENCES

[1] M. MacIver, E. Fontaine, and J. Burdick, "Designing future underwater vehicles: principles and mechanisms of the weakly electric fish," *Oceanic Engineering, IEEE Journal of*, vol. 29, no. 3, pp. 651 – 659, july 2004.

[2] P. Stevenson, M. Furlong, and D. Dormer, "Auv shapes - combining the practical and hydrodynamic considerations," in *OCEANS 2007 - Europe*, june 2007, pp. 1 –6.

[3] B. Jrgens and W. Fork, *The fascination of the Voith-Schneider-Propeller: History and engineering.* Hamburg: Koehler, 2002.

[4] G. Bradski and A. Kaehler, *Learning OpenCV*. O'Reilly Media, September 2008.

[5] Y. Freund and R. E. Shapire, "A short introduction to boosting," *Journal of Japanese Society for Artificial Intelligence*, vol. 14, pp. 771–780, 1999.

[6] B. Balasuriya, M. Takai, W. Lam, T. Ura, and Y. Kuroda, "Vision based autonomous underwater vehicle navigation: underwater cable tracking," *OCEANS '97. MTS/IEEE Conference Proceedings*, vol. 2, pp. 1418–1424, 1997.

[7] H. D. Cheng, X. H. Jiang, Y. Sun, and J. L. Wang, "Color image segmentation: Advances and prospects," *Pattern Recognition*, vol. 34, pp. 2259–2281, 2001.

[8] N. Kiryati, Y. Eldar, and A. M. Bruckstein, "A probabilistic hough transform," *Pattern Recogn.*, vol. 24, pp. 303–316, February 1991.

[9] M. Quiqley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System," in *International Conference on Robotics and Automation*, 2009.