SubjuGator: A Highly Maneuverable, Autonomous Underwater Vehicle

Jennifer L. Laine, Scott A. Nichols, David K. Novick, Patrick D. O'Malley, Daniel Copeland and Michael C. Nechyba

Machine Intelligence Laboratory
University of Florida
Gainesville, FL 32611-6200

www.mil.ufl.edu

Abstract

This paper describes our on-going development of SubjuGator, an autonomous underwater vehicle designed with an emphasis on mobility and agility. SubjuGator recently competed in the 1999 Autonomous Underwater Vehicle Competition in Panama City, Florida, sponsored jointly by the Office Of Naval Research and AUVSI, and earned second place in that competition. Designed and built by a team of undergraduate and graduate students at the University of Florida, SubjuGator is intended for shallow water operation (30 feet). Its small size (1.2m long x 1m wide x .7m high) and tight turning radius ensures high maneuverability. Two motors oriented horizontally provide forward/backward thrust and differential turning, while two other motors, oriented vertically provide ascent/descent and pitch. Buoyancy is controlled using two solenoids which regulate the amount of ballast in the buoyancy compensator located around the electronics compartment; therefore, we do not require motor propulsion for neutral buoyancy or surfacing. On-board sensors include a phased-array, horizontal-scanning sonar, pressure sensor, digital compass, fluidic inclinometer, and a depth sounder. In this paper, we first describe the mechanical and electronic design of SubjuGator. We then discuss the on-board software architecture. Finally, we report results from the ONR/AUVSI competition, and discuss plans for future work and improvements.

1. Introduction

As our world continues to advance in technology and population, humanity will increasingly begin to look towards our oceans as vital providers of valuable resources. In order to fully harness those resources, however, we must develop reliable and adaptable technologies that will allow us to function and thrive in underwater environments. Our oceans are an alien world that challenges us in ways that our more familiar terrestrial environments do not. Remotely Operated Vehicles (ROVs) have long been the only way for people to work and explore the extreme environments of the deep. Such vehicles are inherently limited in their range and maneuverability by their necessary tethers; moreover, the tasks most often performed with ROVs are repetitious and fatiguing to anyone having to control them over many hours at a time.

Autonomous Underwater Vehicles (AUVs), on the other hand, can perform the very same tasks more efficiently with little or no supervision; thus, in recent years, there has been growing interest by the U.S. Navy and others in the development of intelligent AUVs for ocean mapping, mine detection and salvage operations (see, for example, [1]). To promote research and interest in AUVs, both the ONR and AUVSI society have begun to sponsor annual AUV competitions, beginning in 1998. This year's competition was held in Panama City, Florida in August, 1999. Our entry for the competition, SubjuGator, pictured in Figure 1,

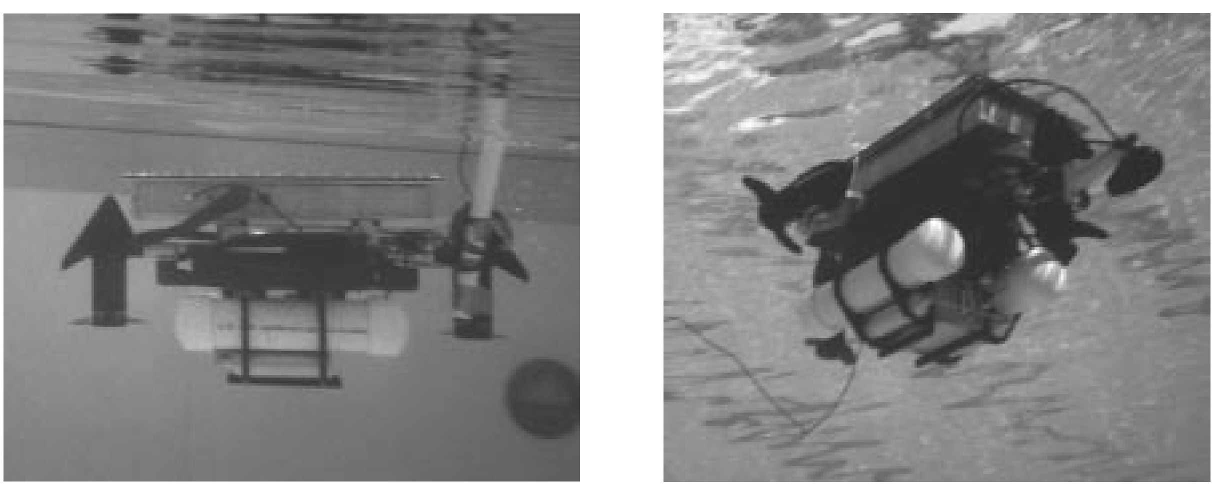

Fig. 1: SubjuGator swims in the University of Florida pool during early testing.

offers an excellent platform for research in underwater sensing and robotics at the University of Florida.

In this paper, we first describe the mechanical makeup of SubjuGator. Next, we describe the electronic and processing hardware, along with the on-board sensing. We then describe the software control of SubjuGator and our strategy for meeting the mission goals of the ONR/AUVSI competition. Finally, we comment on our experiences at the competition and discuss our plans for future work and improvements.

2. Mechanical system

2.1 Body and cage

Figure 2 below diagrams the overall makeup of SubjuGator. The main body of the submarine is composed of a foam core covered in fiberglass with bidirectional and unidirectional carbon fiber for added strength and rigidity. Bolted to the body, the horizontal tail and vertical fins are fabricated from 1/8" aluminum plate and provide directional, roll, and pitch stability. The cage which mounts below the body, is constructed of welded aluminum and Delrin plastic. Within this aluminum frame, the battery is housed, and Delrin tabs are mounted to the outside of the frame. Cutouts in the Delrin tabs support the air cylinders used to control buoyancy. Skids which protect the underside of the submarine are also made of Delrin plastic.

2.2 Buoyancy control

A reservoir of air is stored in two 76,500 cm^3, 3,000 psi spare air tanks (so named because they complement a scuba diver's regular tanks). Both tanks feed air to a regulator, dropping the pressure to 150 psi. From there, the air branches off to two air solenoids. The output of one solenoid feeds directly into the buoyancy compensator, which, when filled with air, increases the buoyancy of the vehicle. The other solenoid attaches to an air valve which vents the air from the compensator, allowing water to fill the chamber, thus decreasing the buoyancy of the submarine.

The air valve is composed of an air-actuated piston and return spring. The 150 psi air enters from the bottom and forces the piston and plunger upward. This allows the air at the top of the buoyancy compensator to exit from the horizontal holes in a stationary aluminum disk and out the plunger. Tolerance between the piston and sleeve allow the air to move around the piston, and the return spring closes the plunger after a short delay. While we have designed the submarine to be neutrally buoyant when the buoyancy compensator is filled completely with water, the buoyancy controller allows the submarine to surface or submerge without motor actuation.

2.3 Electronics container

All electronics are housed in a single compartment (as shown in Figure 3) constructed from 1/4" thick aluminum and sealed with a polycarbonate top. The electrical connections through the container pass through Burton connectors positioned on the side of the box. Bulkhead connections through the box utilize an O-ring seal, and where appropriate, the holes for the connectors are tapped so that the connectors can be screwed in, providing a better seal.

Fig. 2: Exploded view of SubjuGator.

2.4 Propulsion

Four Minn Cota trolling motors propel the submarine. Each motor provides 10.88 kg (24 pounds) of thrust. Two are fixed vertically (fore and aft) and provide pitch stability and ascent/descent motion, while the other two are fixed horizontally (port and starboard) and provide forward and backward thrust.

3. Electrical subsystem

3.1 On-board computing

The main processor for the submarine is an Intel 486SX/33ULP (ultra-low power) evaluation board. Its dimensions are 27.94 cm x 12.7 cm, with an average current consumption of 185 mA. Its following features were important for practical software development: 8Mbytes of DRAM, two serial RS-232 ports, one EPP/ECP parallel port, a PCMCIA socket supporting Type I and Type II cards, a PS/2 keyboard connector, an IDE connector, and a VGA display connector.

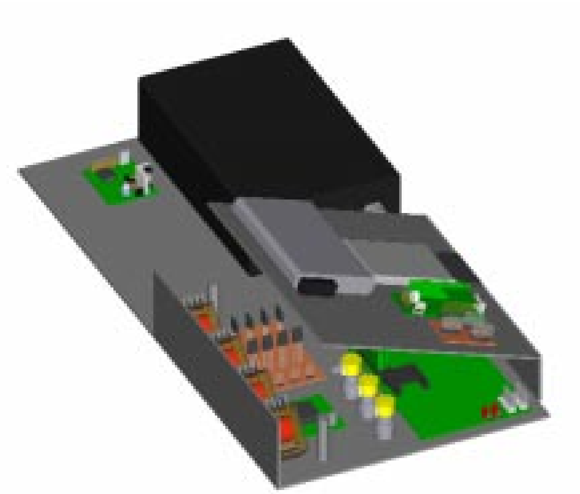

Fig. 3: Electronics box on board SubjuGator.

On board, we run the Red Hat Linux (version 5.2) operating system, an inherently multi-tasking environment. Processes communicate with each other through a shared memory to which each of the processes may attach and share information. Time slicing individual processes guarantees that data is up-to-date or at least periodic, thus minimizing delays and ensuring maximally efficient operation. Using an industry standard processor with a commonly available operating system has significantly reduced our development and testing time, since code can be written, compiled and tested on a faster computer running the same operating system.

The Intel 486 board is interfaced to the motors and sensors through the Motorolla 68HC11 microcontroller. All the electronics are powered by a nickel metal hydride intelligent Energizer battery with a push-button power indicator and a serial interface to read charge information. Its capacity of 3.5 amp-hours provides ample battery life for long periods of continuous testing. SubjuGator's propellers are powered by a 12 volt EXIDE Gel Master deep-cycle battery normally found on electric wheelchairs. This battery provides 40 amp-hours of capacity and has been specifically designed to be drained and recharged many times.

3.2 Wireless ethernet

The primary communications interface to the processor aboard the submarine is accomplished through a wireless ethernet. Two Harris-manufactured wireless LAN, PCMCIA cards, based on the Harris PRISM chipset, communicate at 1.2Mb/s between the submarine and a base station, which can be any laptop running either the Linux or Windows operating system. The communications protocol is TCP/IP, over which telnet sessions may be run to control and monitor all aspects of the on-board electronics without having to physically touch the processor inside the electronics container or extract the submarine from the water.

3.3 Digital compass

The Precision Navigation TCM2 digital compass is a high-performance, low-power electronic compass sensor that outputs compass heading, pitch, and roll readings via an electronic interface to the central processor. Since the compass is based upon a proprietary triaxial magnetometer system and a biaxial electrolytic inclinometer, it contains no moving parts.

3.4 Depth sensor

SubjuGator uses an MSP-300 depth sensor by Measurement Specialties Inc., rated to 100 psi, that outputs an analog DC voltage between 1 and 5 volts. For each 10 feet of water, the sensor voltage changes by 0.225 volts. To maximize sensitivity, we built an amplifier which produces a 5 volt swing for a depth change of 20 feet. Based on the resolution of the analog to digital port on the 68HC11 microcontroller unit, we have an approximate sensor resolution of two inches of water.

3.5 Phased-array sonar

SubjuGator carries a Sea Scout phased-array sonar transducer manufactured by Interphase. This sonar consists of eight individual transducers positioned as a forward-looking array and one transducer positioned downward. Both the forward and downward looking transducers have a 12° conical beam modulated at 200 kHz. The forward looking array can sweep across 90° (–45° to +45°). Since the Sea Scout sonar achieves beam-shaping through a phased array of sonar transducers, it does not have any moving parts.

We chose sonar as our primary sensor for several reasons. First, unlike a camera, it is unaffected by ambient light and can therefore generate useful sonar images through murky waters. Second, it has a flexible range — from very close to very distant. Third, the nature of the returned data lends itself to analysis through common image processing techniques, as discussed in Section 4.2 below.

The sonar communicates to the main processor via a standard PC parallel port. One horizontal scan is divided into 91 beams corresponding to -45° to +45° from the centerline of the transducer. Each beam has a separate gain or intensity which determines how much energy is emitted and thus the effective range of that beam and the size of the object (if any) being imaged. The strength of the return signals reflected off the objects in the water are quantized into a three-bit number, and the time-of-flight calculation of the signal determines the distance. Using the angle of the beam, the distance and the strength of the return, a two-dimensional view of a 90° cone in front of the submarine can be constructed. Figure 5 shows some sample images gathered with the sonar.

Our sonar performs one other important task — namely, the height measurement of the submarine above the floor of the pool. A separate, non-steerable, sonar element is housed in the transducer pod for this purpose. As before, we use time-of-flight of the sonar signal to arrive at an approximate height for the submarine.

4. Software structure and navigation

While our hardware design of SubjuGator is flexible enough to allow for a variety of underwater missions, we have written some of our software to meet the specific mission objectives of the ONR/AUVSI AUV competition. Below therefore, we first briefly describe the mission challenges in the AUV competition. Subsequently, we outline how we have designed our software to meet those challenges.

4.1 Competition description

The second annual ONR/AUVSI AUV competition took place in August, 1999 and was hosted by the U.S. Navy's Coastal Systems Station in Panama City, Florida. An approximately oval-shaped test pond at the Coastal Systems Station, 110m by 70m in size with a maximum depth of 12m, served as the arena for the competition. In the pond, six 3m-wide gates, composed of 4" PVC pipe, are placed along an approximate isobath near the periphery of the pond. During the competition, submarine entrees must start at a designated point in the pond, and autonomously navigate underneath the crossbar of all six gates. At the completion of the run, the submarine is then required to surface.

We have designed our software to be capable of completing this mission. Figure 4 below illustrates the overall software flow and architecture on-board SubjuGator. Five different main processes control the behavior of SubjuGator. They interact with each other and with lower level processes via shared memory. The low-level processes handle sensor readings and transfer values from shared memory to the microcontroller directing the motors and the solenoids.

A process manager, *procman*, initializes the shared memory, and spawns all of the other processes. The two main processes, *maintain-height* and *gate-detection/obstacle-avoidance* determine a suggested heading and speed based on sensor readings, and use the errors between the current and desired sensor values to smoothly direct the submarine along a target path. An *arbiter* process determines which of the suggested headings and speeds will be used, while the *pilot* process translates the error between current and desired heading, pitch, speed, and depth into motor commands. Below, we describe each of the main processes in turn.

4.2 Gate detection/ obstacle avoidance

Gate detection and obstacle avoidance both exclusively use the phased-array sonar to accomplish their respective tasks. For that reason, they are combined into one process. Obstacle avoidance, the simpler of the two algorithms, steers the submarine away from any large concentrations of high sonar returns, such as the walls of the pond. Gate detection, on the other hand, is a significantly more complicated function. This process has to detect and track a gate over a relatively large distance, and clearly distinguish it from random noise in the sonar images. Consider for example, Figure 5, which shows nine sonar images in sequence as the submarine is approaching a sample gate (approximately 12 feet in front of a vertical pool wall). For this sequence of images, the sonar is set to look 40 feet ahead, and the submarine is approaching the gate head-on; in other words, in the sonar images, the gate is located at $0°$.

From Figure 5, we observe that the pool wall behind the gate is quite apparent as a massive concentration of high-return signals. We also note, however, that there is quite a bit of noise in the sonar images, most likely the result of sonar echoes in the confined pool area. In fact, in some of the early sonar images of the sequence, the gate in front of the submarine is difficult to distinguish from the surrounding false echoes. Thus, it is apparent that analysis of an isolated sonar image will most likely not yield accurate gate-detection results with a high degree of confidence. Therefore, our sonar analysis does not look at sonar images in isolation, but rather as a coherent sequence, where stationary objects will track in a speed-dependent manner.

In our analysis, we first preprocess the sonar images by equalizing all nonzero returns. Although this reduces the strength-of-signal resolution from four levels to two, we have found that the additional two levels of resolution are not particularly reliable and therefore may safely be discarded.

Next, we use a simple region-growing algorithm to detect contiguous regions of high sonar returns. Nearby regions are merged into a single larger region if the distance between the two regions is below some threshold. For each of the regions, we then create a parametric description, which includes the height, width and location of that region. Those regions which do not conform to the expected size and shape of a gate (our target object) within some tolerance are eliminated from further consideration.

Next, for each of the remaining potential gates, we search the previous sonar image for similarly shaped objects in the approximate vicinity of those gate-like objects found in the current sonar image. If a current region does not match the approximate location of a similar region in the previous sonar image, we assume that it is a new object that has not yet been detected. We assign a relatively low confidence value to such a region, since it may only represent some temporary false echoes. On the other hand, if we do find a match in the previous image, we increase the confidence value that that region represents a gate. This tends to strengthen the confidence value for actual gates over a sequence of sonar images, while at the same time suppressing false positives.

In Figure 5, we have plotted those regions with nonzero confidence value for the nine sonar image sequence. Note how the region corresponding to the gate grows stronger over time, while

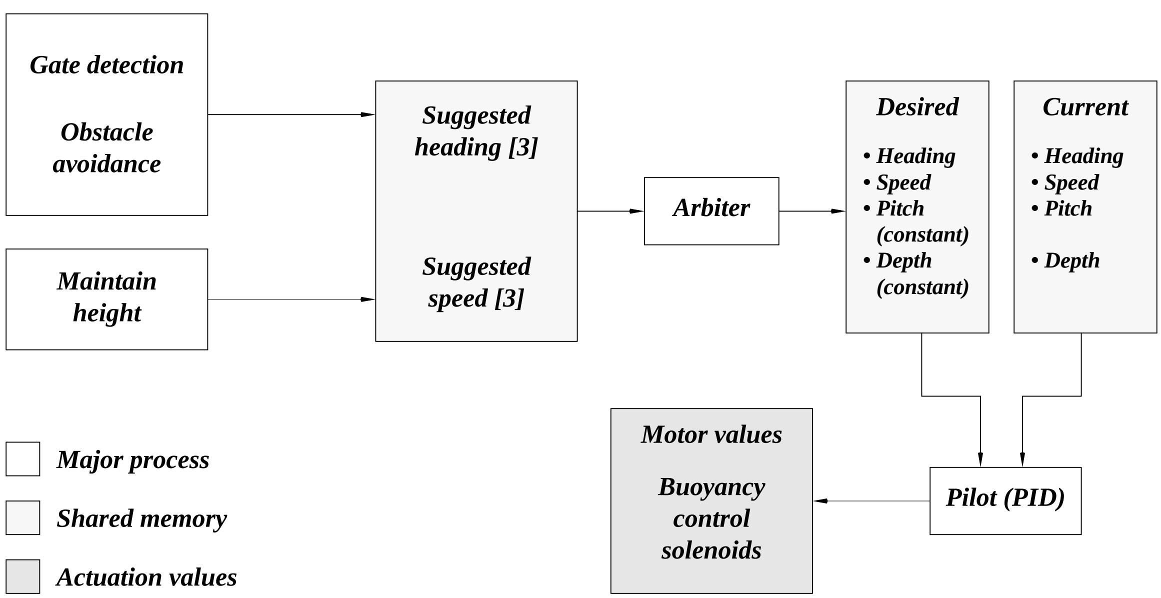

Fig. 4: Overall software architecture for SubjuGator.

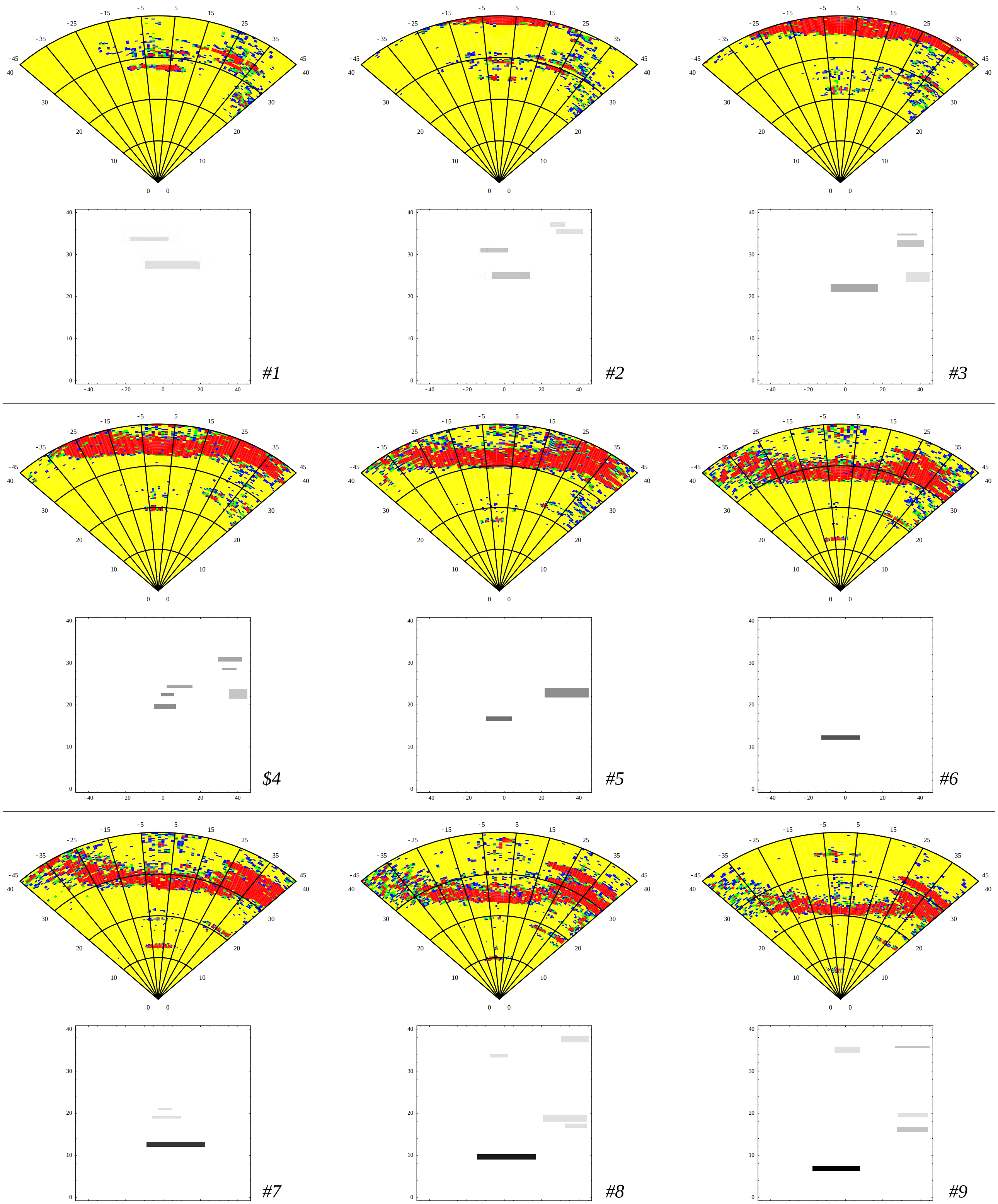

Fig. 5: As SubjuGator approaches a gate (in front of a wall), the gate-detection algorithm becomes increasingly confident in the existence and location of the gate. Each rectangular plot indicates, through color intensity, probable locations of a gate from the corresponding sonar image above it. All distances are in feet.

background noise remains at relatively low confidence levels. By the seventh sonar image in the sequence, at a distance of approximately 12ft, the confidence in the gate region is sufficiently high to alert the arbiter that a gate has been detected at a specific distance and heading. Thus, the gate detection/obstacle avoidance process does not recommend a course of action for the submarine until either a gate is detected or a wall (or other large obstacle) is encountered, at which time it will tell the arbiter either to steer away from large obstacles or move towards an identified gate.

4.3 Maintaining a constant height

Following the isobath that the gates are approximately located on can be accomplished by diving to and staying at a specified depth, and then maintaining a fixed height off the bottom. As shown in Figure 6, the submarine can change its height by varying its horizontal position right or left.

If a deviation from the desired height is detected, *maintain-height* will calculate a new heading request based on the height/horizontal displacement, the distance ahead the submarine is "looking," and the effectiveness of the last correction request. A height error is translated into a horizontal displacement, and a course heading is calculated to put SubjuGator back on course within *D* feet. *D* can vary from small to large to accommodate quick responses in tight turns, and smooth corrections in long straight runs.

4.4 Arbiter

The heading is selected by the *arbiter* on a priority basis. *Gate-detection/obstacle-avoidance* has higher priority than *maintain-height*. The process *gate-detection/obstacle avoidance* gives a suggested heading only when it perceives that a gate is present under the correct conditions. Otherwise it asserts a -1 value for heading. *Maintain-height* uses the height above the bottom to determine its lateral position and make appropriate heading suggestions. It continuously presents a suggested heading and speed to the arbiter.

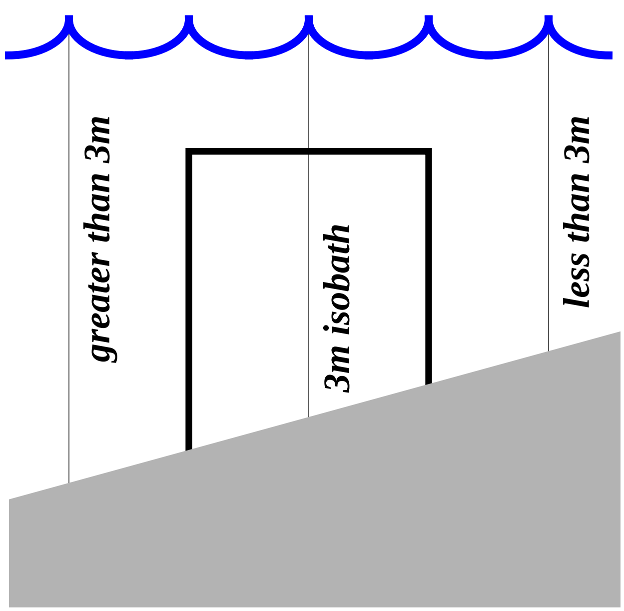

Fig. 6: SubjuGator will follow a 3m isobath along the pool perimeter.

Based on this priority scheme, the *arbiter* first checks to see if the *gate-detection/obstacle-avoidance* process has suggested a heading (greater that -1) and speed. If so, these values are passed on to the desired heading and speed memory blocks. Otherwise the suggested heading and speed from *maintain-height* is chosen and transferred. The main goal of this methodology is to maximize the use of the most reliable sensors (compass and depth sensor) as well as error check the less reliable sensor (sonar) with the more reliable ones.

4.5 Pilot

As the submarine moves through the water, errors between the desired and current values of heading, speed, pitch, and depth will be controlled through a standard PID controller. The determination of the motor actuation values is based on the submarine's position and orientation divergence according to,

$$m(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (1)$$

where *m(t)* is the motor value and *e(t)* represents the error at time step *t*. The continuous equation is converted to its discrete equivalent and the errors are calculated from the difference between the current and desired heading, pitch and depth.

Manually tuning the gains in equation (1) above can involve much trial-and-error. In order to short-circuit this process, we have implemented *Q*-learning [2] to automatically tune the gains to achieve the best response over time. We allow the submarine to explore different gain combinations and reward those with desirable control properties such as small overshoot and fast response time. This is not only a painless alternative to manual fine tuning, but also offers an automated procedure for adjusting the gains, if the mechanical parameters of the submarine are changed or redesigned.

5. Results and future work

During allotted practice time in the test pond at the Coastal Systems Station, SubjuGator successfully navigated all six gates on the course. Unfortunately, weather conditions the day of the competition deteriorated with periodic rain and gusts of wind that caused relatively strong currents in the test pond. Therefore, our simple PID controller, whose gains had been tuned in calmer waters, generated significantly more overshoot and oscillations during our competition run. While we successfully steered through the first four gates, we missed each of the last two gates by no more than one or two feet.

Less than a month removed from this year's competition, we are already looking forward to next year's AUV competition, and plan a number of improvements. We are currently redesigning the body to be sleeker and more hydrodynamic than our current, admittedly bulky design, Figure 7 depicts this new design in its earliest stage. Since the vertical propellers in our current design were rather underused, we have oriented all four propellers along the forward motion of the submarine. While this design loses some up/down maneuverability, it significantly enhances maneuverability in the forward and backward directions.

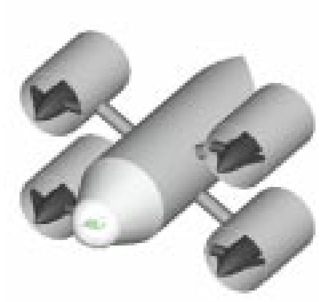

Fig. 7: New design for SubjuGator currently being built.

We also plan to add some additional on-board sensors, including a camera (for up-close sensing), possibly to be mounted in the nose of the new body, and an inertial measurement unit. Finally, we expect to improve the robustness of our controller, so that the submarine is better able to handle those disturbances that forced our submarine partially off-track in August.

Acknowledgments

We gratefully acknowledge the following corporations for their generous donations to the SubjuGator project: Burton Electrical Engineering, Exide Batteries, Energizer Power Systems, Intel, Interphase, Harris Semiconductors and Precision Navigation. We also thank the University of Florida College of Engineering for their financial support, and the University of Florida Division of Recreational Sports for allowing us access to the University's pool facilities. Finally, we thank AUVSI, ONR and the Coastal Systems Station for their sponsorship and hosting of the AUV competition.

References

[1] K. P. Valavanis, ed., "Special Issue on Design and Navigation of Autonomous Underwater Vehicles," *IEEE Robotics and Automation Magazine*, vol. 6, no. 2, 1999.

[2] C. J. Watkins, *Learning from Delayed Rewards*, Ph.D. thesis, King's College, Cambridge, UK, 1989.