SubjuGator: A Highly Maneuverable, Intelligent Underwater Vehicle

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

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

http://www.mil.ufl.edu/subjugator

Abstract

Undergraduate and graduate students at the University of Florida have redesigned, modified and tested an autonomous submarine, SubjuGator, to compete in the 1999 ONR/AUVSI Underwater Vehicle Competition. A modified version of last years entry, SubjuGator is designed for shallow operation (30 feet), with emphasis on mobility and agility. SubjuGator retains its small size (1.2m long x 1m wide x .7m high) and tight turning radius, ensuring 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.

SubjuGator is controlled through an embedded 486/33MHz processor running the Linux operating system. It is interfaced to a number of sensors, including a phased-array, horizontal-scanning sonar, a pressure sensor, a digital compass, a fluidic inclinometer, and a depth sounder. Two separate power supplies drive the motors and electronics, respectively. The motors are powered by a 40 amp-hour sealed lead-acid gel cell battery, while a 3.5 amp-hour nickel metal hydride battery powers the electronics. Most of the components on the vehicle have been either donated from companies or designed and built in our laboratory. The electronics and electronics container are designed to be modular so that they may be removed from the current submarine body and used in a future design without major effort.

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; that is why both the ONR and AUVSI organizations are interested in their development and have sponsored the annual AUV competition in Panama City, Florida.

Our entry for the competition — SubjuGator — offers an excellent platform for research in underwater sensing and robotics. Not only are students able to implement algorithms and theories learned in class on a practical, real-world platform, but SubjuGator can also support the development and research of new techniques in underwater sensing and navigation.

In this paper, we first describe the mechanical makeup of SubjuGator. Next, we describe the electrical and processing hardware, along with the on-board sensing. Finally, we describe the software control of SubjuGator and our strategy for completing the competition objectives successfully.

2. Mechanical system

2.1 Body and cage

Figure 1 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. The horizontal tail and vertical fins are fabricated from 1/8" aluminum plate and are bolted to the body. The tail structure provides directional, roll, and pitch stability.

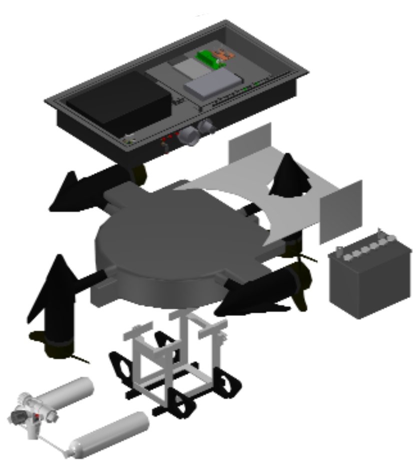

Fig. 1: Exploded view of SubjuGator.

The cage which mounts below the body, is constructed of welded aluminum and Delrin plastic. The battery is housed within the aluminum frame, and Delrin tabs are mounted to the outside of the frame. Cutouts in the Delrin tabs support the air cylinders used by the buoyancy controller. 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. The air fills the compensator and 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 and thus decreasing the buoyancy of the sub.

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.

The submarine is set 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 2) constructed from 1/4" thick aluminum and sealed with a polycarbonate top. All electrical connections through the container utilize Burton connectors positioned on the side of the box. All 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.

2.4 Propulsion

Four Minn Cota trolling motors are used for propulsion. 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. The other two are fixed horizontally (port and starboard) and provide forward/backward thrust. A 254 cm diameter aluminum shroud prevents direct access to the propellers, and the back of each shroud is further protected with crossbars.

3. Electrical subsystem

3.1 Motor control

The motor driver for the vehicle was designed and built in our laboratory using engineering design automation software and a computer controlled board-milling machine. We employ high power 12V coil Schrack relays to change motor direction and MOSFETs, rated at 75 amps, to amplify the power of the PWM signal and the direction signal to each of the four motors. These signals originate as TTL levels from a 68HC11 microcontroller, and optoisolators raise the voltage of the digital signals to the 12 volt maximum desired by the motors and the relays. External protection diodes prevent the FETs from experiencing high voltage swings caused by constant charging and discharging of the motors. Consequently, the motor driver is robust and cost efficient to repair. The total cost of the board and parts, which are all readily available, is $16, and the circuit utilized is relatively static insensitive compared to other available motor driver circuits. The software controlling the motors on the microcontroller allows 100 discrete speeds in both the forward and reverse direction for each motor.

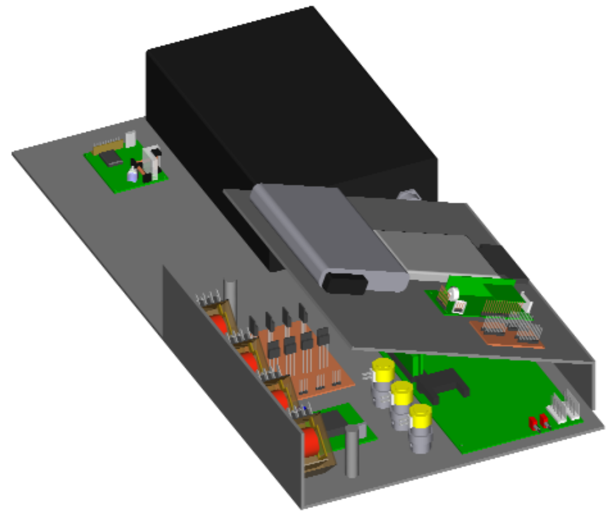

Fig. 2: Electronics box on board SubjuGator.

3.2 Power supply

The power for SubjuGator's motors is supplied from a 12 volt EXIDE Gel Master deep-cycle battery normally used for electric wheelchairs. The dimensions of the battery are 19.685 cm x 13.17625 cm x 18.57375 cm. It weighs 24 pounds and has a capacity of 40 amp-hours. We chose a deep-cycle battery since it is designed to be drained and recharged many times.

The battery powering the electronics is a nickel metal hydride intelligent Energizer battery with a push-button power indicator and a serial interface to read charge information. It outputs 12 volts, has a capacity of 3.5 amp-hours, and a weight of 5 pounds. A commonly available DC-to-DC converter is used to step down and regulate the voltage to 5 volts at 3 amps to power the electronics. The long battery life of the nickel metal hydride battery makes it well suited for our application.

3.3 Main processor

The main processor for the sub is an Intel 486SX/33ULP (ultra low power) evaluation board. Its dimensions are 27.94 cm x 12.7 cm, and has an average current consumption of 185 mA. The 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.

On board, we run the Red Hat Linux (version 5.2) operating system, an inherently multi-tasking environment. Interprocessor communication is achieved through shared memory. A block of memory is first created which contains program structures or specialized variables. The different software processes then attach to this shared memory and use these structures to communicate between one another.

Time slicing the routines 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 reduces development and testing time. The code can be written, compiled and tested on a faster computer running the same operating system. Moreover, multiple researchers can test modules in parallel.

3.4 Wireless ethernet

The primary communications interface to the processor aboard the sub is accomplished through wireless ethernet (IEEE802.11). Two Harris-manufactured wireless LAN, PCMCIA cards based on the Harris PRISM chipset communicate at 1.2Mb/s between the sub and a base station, which can be any laptop running Linux or Windows95. The communications protocol is TCP/IP. Over the TCP/IP link 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 sub from the water.

3.5 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.

3.6 Depth sensor

SubjuGator uses an MSP-300 depth sensor by Measurement Specialties Inc. rated to 100 psi and outputting 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 for this competition, 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 microcontroller unit (the same chip used for motor control), we have an approximate sensor resolution of 2 inches of water.

3.7 Phased-array sonar

SubjuGator carries a Sea Scout phased-array sonar transducer manufactured by Interphase. The 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. Finally, the nature of the returned data lends itself to analysis by common image processing algorithms.

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 sub is constructed. Figure 3 shows several sample images gathered with the sonar.

We analyze the data returned from the sonar using common image processing techniques. First, we clean and smooth the image using a dilation filter (to reduce the effect of operating system-induced noise) and an averaging filter. Due to the rough quantization of the data and frequent saturation of the transducer return signals, we ignore all but the largest values. Next, we remove noise and isolate the features of the image that could potentially be a gate or a wall. This is done through a “blob” analysis algorithm wherein each “blob” or feature consisting of a localized area of high-strength returns is examined for its centroid and size. All areas of high-strength returns under a certain size threshold are discarded as noise.

The data from the sonar image analysis is then used to guide the sub through the gate when approaching a gate. It is not intended to actively find the gate initially. Instead, it asserts itself when a gate is present at a prescribed distance and heading. The same data can also be used to prevent collisions with the wall of the pool.

Finally, the sonar performs one other important task — namely, the height measurement of the sub 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 sub.

4. Software structure and navigation

The overall software flow in Subjugator is illustrated in Figure 4 below. 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*, creates and initializes shared memory and spawns all of the other processes. Three of the main processes, *maintain-height*, *gate-detection/obstacle-avoidance* and *target-zone*, 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 each three 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.

4.1 Gate detection/ obstacle avoidance

Gate detection and obstacle avoidance both exclusively use sonar to accomplish their task. For that reason, they are combined into one process. Obstacle avoidance, the simpler of the two algorithms, steers the sub away from any large concentration of high sonar returns (or “blobs”). Gate detection, on the other hand, is a complicated function. The process has to detect a gate when it is both far away, and looks like noise, and when it is near, so that the two uprights are resolved into two blobs. It does this by using a sequence of sonar images to analyze the data over time. For each image, a confidence measure is computed for each blob. If a blob is consistent across multiple images, then it is probably a gate (or other stationary obstacle). When the confidence is high enough, the process issues a recommendation to the arbiter. That is to say, this process does not recommend a course of action for the sub until either a gate is detected or a wall (or other large

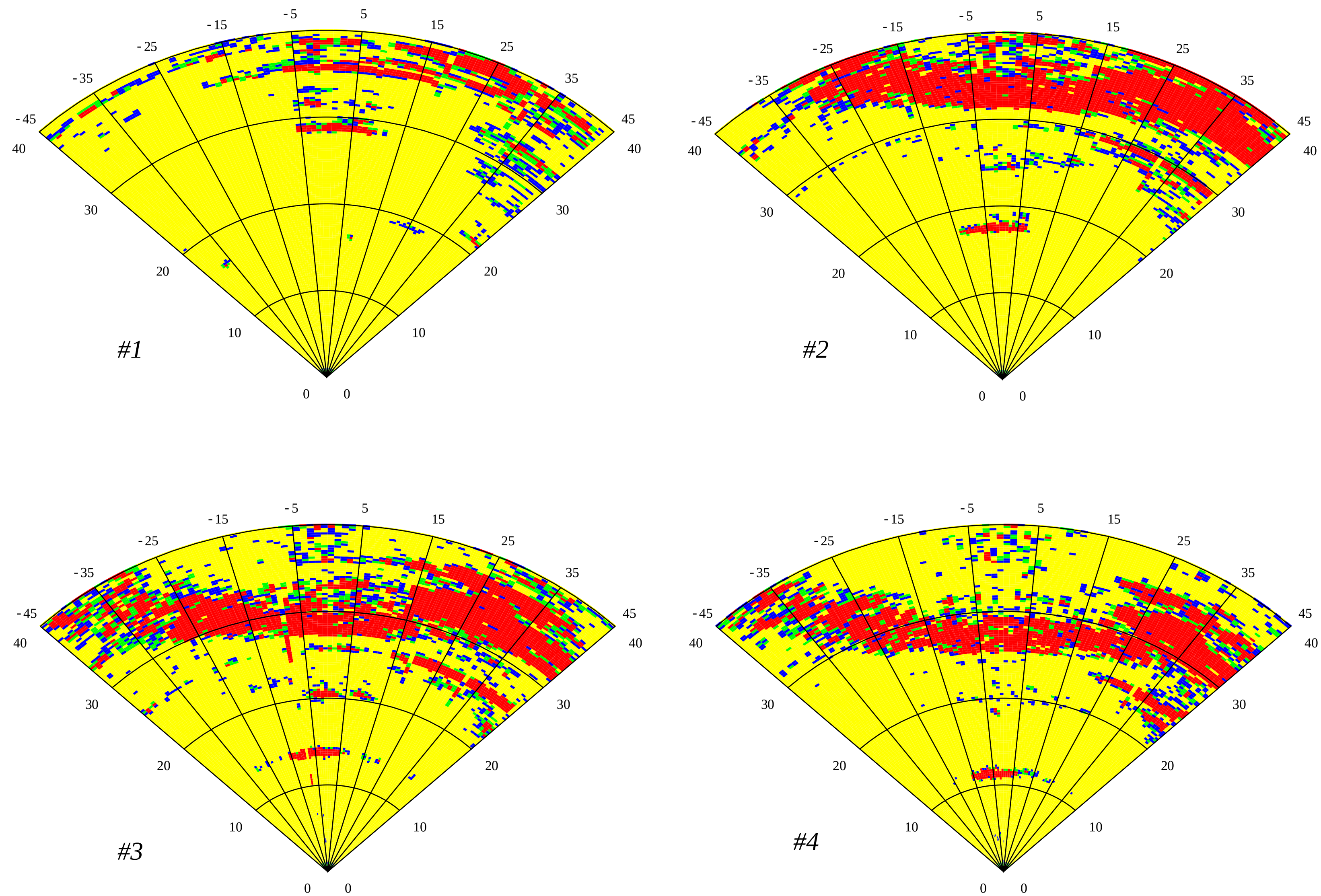

Fig. 3: A sequence of sonar images for a sample gate approximately 10 feet in front of a wall as Subjugator draws closer to the gate (red indicates a strong return, while yellow indicates no return).

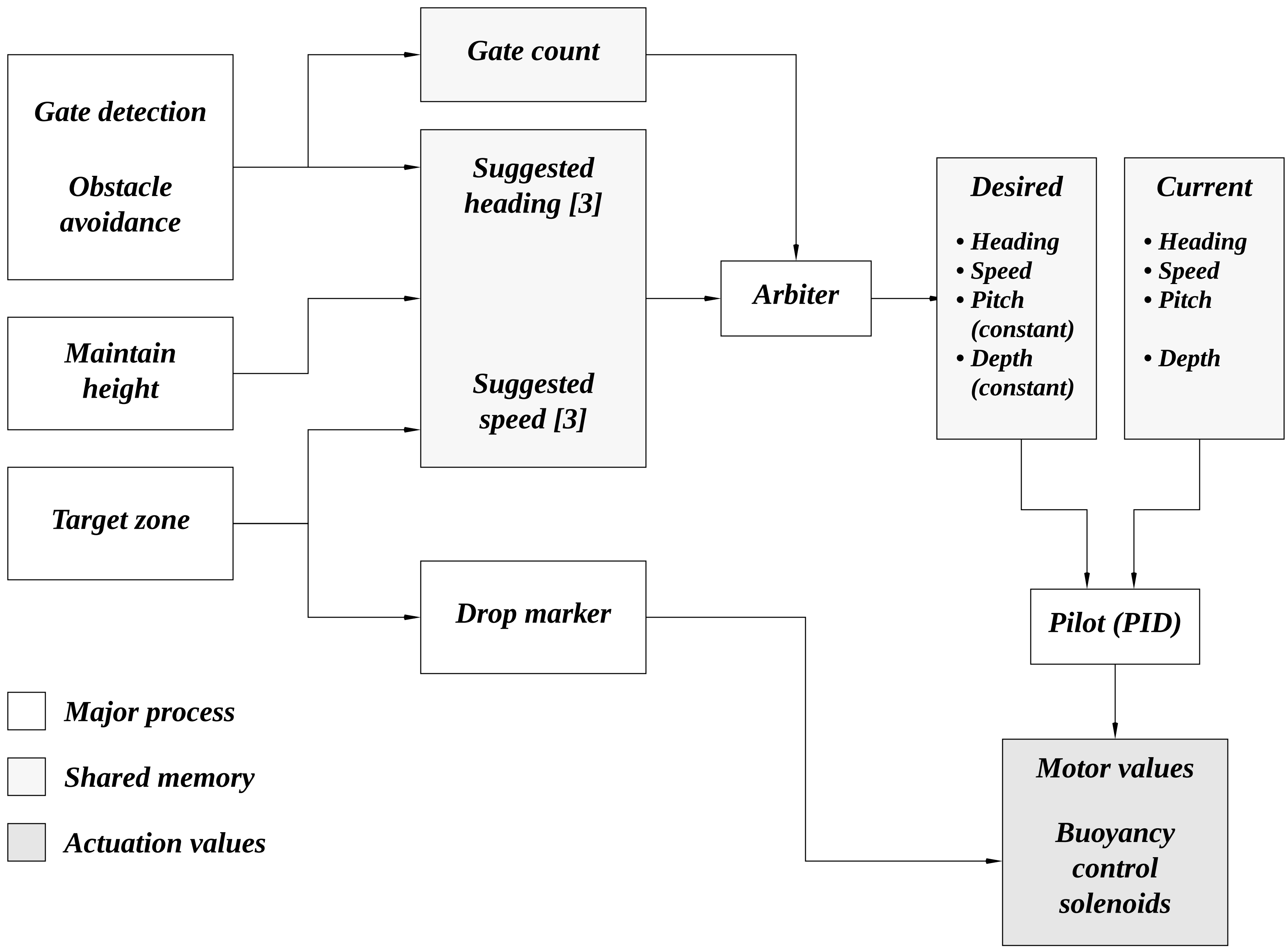

Fig. 4: Overall software architecture for SubjuGator.

obstacle) is encountered, at which time it will tell the arbiter either to steer away from large obstacles or move towards an identified gate.

4.2 Maintaining a constant height

Following the isobath that the gates are 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 5, the sub 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 sub 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.

4.3 Target zone

The *target-zone* process is the lowest priority process until all of the gates have been successfully navigated. At that point the arbiter aims the sonar downward by actuating a piston; it then promotes this process to the highest priority. To find the target zone, the sonar will be used to follow the pipe placed from the last gate to the target. Using the same image processing algorithms as used by the gate detection and obstacle-avoidance process, the sub will interpolate a line along the pipe and a corresponding heading. The target zone will be identified by the sonar and the depth marker deployed.

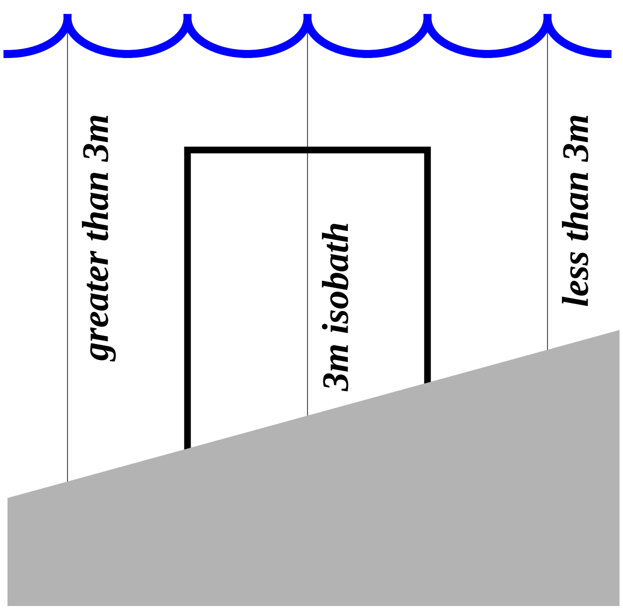

Fig. 5: SubjuGator will follow a 3m isobath along the pool perimeter.

4.4 Arbiter

The heading is selected by the *arbiter* on a priority basis. *Gate-detection/obstacle-avoidance* has the highest priority followed by *maintain-height*. *Target-zone* takes over once the last gate is passed.

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 is always presenting a suggested heading and speed.

Once the last gate is passed, the *target-zone* process uses the sonar which will be rotated downward by 90°. The sonar guides the sub along the pipeline and to the drop zone.

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. This changes after the last gate, when the *arbiter* makes *target-zone* the highest priority.

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 least 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,

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

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 *K* in equation (1) above can involve much trial-and-error. In order to short-circuit this process, we have implemented *Q*-learning to automatically tune the gains to achieve the best response over 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. Strategy

Using the knowledge that the gates are in a specific isobath around the competition area, SubjuGator will simply follow that isobath while looking for gates. When a gate has been identified and judged to be within some "critical distance," the sub will be permitted to maneuver if necessary to pass through the gate. After which it will return to the isobath and continue its mission. Each gate navigated will be counted, and after six gates SubjuGator will begin searching for the pipeline to locate the target zone. The target marker will then be released, and the sub will surface.

6. Conclusion

SubjuGator not only offers an excellent platform for research in underwater sensing and robotics, but also teaches valuable skills for working as part of a larger team and project. With generous support from several corporate sponsors, as well as much hard work, we have tried to improve on the design of SubjuGator from last year, and look forward to a challenging and exciting time in Panama City at the AUV competition.

7. 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 CSS for their sponsorship and hosting of this exciting competition.