SubjuGator 2018: Design and Implementation of a Modular, High-Performance AUV

D. Volya, A. Albritton, N. Suhlman, D. Olis,
K. Cohen, E. M. Schwartz[1]
dvolya@ufl.edu, aalbritton@ufl.edu,
grymestone@ufl.edu, daniel.olis@ufl.edu,
kcohen@ufl.edu, ems@ufl.edu

***Abstract* – Here we present SubjuGator 2018, an updated version of SubjuGator 2017. SubjuGator was made by UF's Machine Intelligence Laboratory, a team of 17 students, most of whom are young undergraduates. The current version of our autonomous underwater vehicle (AUV) focuses on robust control, hardware improvements, and software innovations. In particular, this model includes new thrusters, imaging sonar, bilge pump, upgraded passive sonar system, and other design improvements. In this paper we also address testing, competition, and teamwork strategies which were modified based on previous experience, changes to competition rules, and structure of our team.**

I. Competition Strategy

Leveraging 21 years of autonomous underwater vehicle (AUV) development experience at the University of Florida, which has produced 7 prior individual platform designs, the SubjuGator family of AUVs has progressed to accommodate advances in sensors, computing, and mission requirements leading to the design of the current generation SubjuGator 8 vehicle.

Moreover, for the past few AUVSI RoboSub competitions, SubjuGator 8 served as the primary development and competing platform. This experience provided strategy decisions for the 21st annual competition, namely in minimizing problems and accentuating advantages. One example involves unreliable thrusters, which hindered maneuverability. Decisions were made to replace half of SubjuGator 8's thrusters, which involved adding an external pressure vessel.

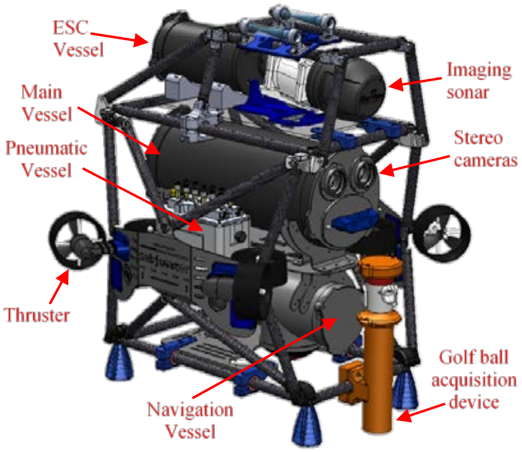

Fig 1. SubjuGator 8.

Compared to previous years, this years' tasks require an increasing level of autonomy, thus providing an additional source of consideration for strategy decisions. The 21st annual AUVSI RoboSub competition consists of tasks that require first searching for a task of interest, followed by stable maneuvering and alignment. This differs from previous years in that, in order to obtain maximum points, a priori external knowledge of tasks' positions and orientations cannot be hard coded, but rather found – such as finding the Casino and going through the gate when SubjuGator starts at a random choice for initial orientation. Thus, it is crucial that SubjuGator is capable of not only searching for the appropriate task, but also remembering past information and knowledge.

Hence, SubjuGator searches for regions of interest satisfying certain constraints by using an Active Imaging Sonar, Passive Sonar, and relative position information. The position information is stored. Upon finding a region of interest, cameras are used for further validation

---

[1] Authors are member of the Machine Intelligence Laboratory (MIL, www.mil.ufl.edu) at the University of Florida, Gainesville, FL 32611, USA.
- Alan Albritton and, Kip Cohen: students in the Mechanical and Aerospace Engineering (MAE) Department
- Daniel Olis: student in the Electrical and Computer Engineering (ECE) Department.
- Daniel Volya and Nicholas Suhlman: students in the Computer and Information Science and Engineering (CISE) Department.
- Eric M. Schwartz: MIL Director, affiliate faculty of the Center for Intelligent Machines and Robotics (CIMAR), and ECE faculty member.

and discovering intrinsic properties, such as color or dice number. Upon correct discovery, SubjuGator preforms defined maneuvers to solve the task. Given previous experience, in order to increase the margin of error and remove some of the burden on software, hardware design decisions were made such that SubjuGator would still be able safely solve tasks. For example, by using suction, SubjuGator has more leeway in aligning to the chip dispenser and then picking up chips (golf balls.) Moreover, to even further minimize error, software design employs a number of fallbacks, filtering, and error correction techniques.

Finally, since the composition of the team this year consist of predominantly new members, decisions were made to master feasible tasks while prioritizing test time – hence encouraging learning through trial by error. Thus, many of the design strategies took into account team capabilities and experience, while working on top of the infrastructure left behind from previous members.

II. Design Creativity

The eighth generation SubjuGator AUV has the capabilities to meet and exceed the challenges of the competition. With a light-weight carbon fiber framework surrounding an aluminum core, and high-power vectored thruster configuration, SubjuGator 8 has the speed, modularity, and maneuverability necessary to accomplish the competition's numerous tasks within the allotted time.

A. Hardware Design

A major feature of SubjuGator 8 is the ability to sustain operation after a failure has occurred, where the failure can be of mechanical, electrical, or software origin. To achieve this goal, the vehicle is designed so that during a subsystem failure, the vehicle as a whole is still capable of completing a task, or at the very least, safely returning to a recovery point to be removed from the environment. As an example, the redundant eight thruster design allows for the vehicle to maintain full six degrees of freedom control in the event that on-board software detects a thruster failure.

Design for fault tolerance also motivates a modular system structure, with each module performing specific tasks while communicating with other modules' systems. Modules are each encapsulated in their own pressure vessel.

To unify the different modules into a durable and light weight platform, a space-frame type chassis was constructed from carbon fiber tubes and three aluminum sheet sections. This structure provides a number of key features:

- Protection of the pressure vessels and external sensors from collision
- Thruster mounts farther away from the center of mass for improved orientation control
- Versatile mounting space for new auxiliary devices, additional vessels, sensors, etc.
- A sturdy support structure for handling and seating the platform on land

1. Navigation Vessel

The sensors and components necessary to pilot an underwater vehicle are abstracted into their own vessel. Figure 2 shows a model of the navigation vessel. The raw data from all of the sensors is combined on a student designed circuit board.

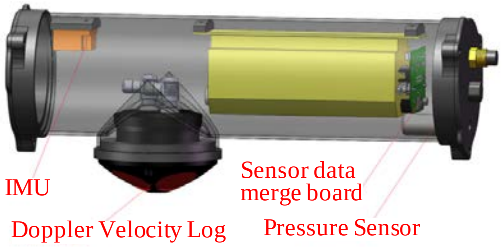

Fig 2. Model of the navigation vessel.

2. Electronic Speed Controller Vessel

After encountering several complications with the internal motor controllers in the old thrusters, an effort has been made to switch over to new thrusters. To vary the speed, external motor controllers are necessary, along with an interface to use the existing TIA-485 architecture and output a Pulse Width Modulated (PWM) signal. A pressurized vessel is used to house the external components without excessively modifying the Main Vessel. The motor controllers are secured to the aluminum hull with epoxy with a coating of thermally conductive filler to dissipate heat. Additionally, a custom circuit board is fastened to an internal mounting shelf.

3. Passive Sonar

The ability to track a point source of sound in the water is encapsulated into the passive sonar pressure vessel. It contains a passive sonar amplification and filtering board (Figure 3), necessary power regulation, and USB communication. An Analog Digital 4-channel Data Acquisition ADC (ADAR7251) is used to simultaneously sample, amplify, convert, and filter the four incoming signals. The board was designed by Sylphase – a startup founded and run by a former MIL student – and can simultaneously track two distinct frequency acoustic sources.

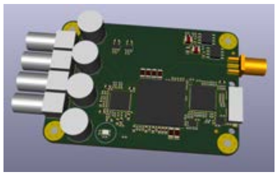

Fig 3. Passive sonar PCB.

4. Pneumatics System and Actuators

SubjuGator 8 integrates three types of independently operated pneumatic mechanisms (a golf ball release, torpedo launcher, and marker dropper) into its design. The mechanisms are used to complete mission specific tasks and are controlled using five of twelve pneumatic solenoid valves which are housed in a separate, compact pressure vessel (Figure 4). This design allows for quick-disconnect fittings to facilitate easy addition or removal of pneumatic subsystems.

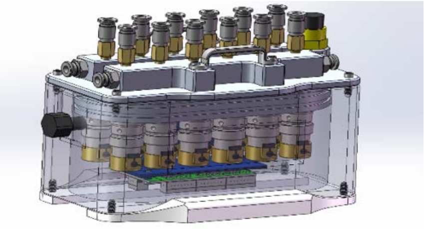

Fig 4. Pneumatic solenoid housing/controller.

5. Golf Ball Acquisition Device

To retrieve and release golf balls, the team has created a golf ball acquisition device that uses a 24 *V* bilge pump to produce a negative pressure around the golf balls and suck them into the nozzle. The nozzle features ribs to both center the balls and prevent clogging. At the end of the nozzle, a reverse-acting pneumatic cylinder keeps the balls secured inside, even when the bilge pump is off. When it is time for the balls to be released, the cylinder retracts, allowing the balls to fall freely into the bin.

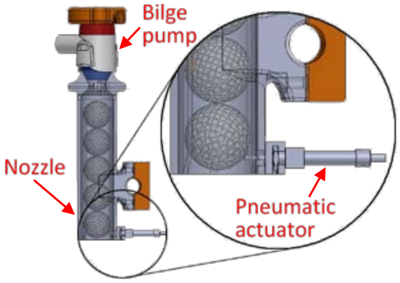

Fig 5. Golf ball acquisition device.

B. Software Design

SubjuGator 8's software stack is built on the Robot Operating System (ROS) Kinetic. After RoboSub 2013, MIL made (and continued to make) our repositories public in hopes that other projects would make use of them. We are in the process of improving our documentation, to further encourage external use. Our ROS Teledyne Blueview Driver, along with the rest the software is open-sourced, and available on GitHub[2].

1. State Estimator

The state estimator uses an inertial navigation system (INS) and an unscented Kalman filter (Figure 6). The INS integrates inertial measurements from the IMU, producing an orientation, velocity, and position prediction. Due to noise and unmodeled errors in the inertial sensors, the INS prediction rapidly accumulates error. The Kalman filter estimates the state by comparing the output of the INS prediction against the reference sensors, which are a magnetometer, depth sensor, and Doppler Velocity Log (DVL). By correcting the INS using the errors

---

[2] All code is located at https://github.com/uf-mil.

- The mil_common repository contains code common across all of MIL projects.
- SubjuGator repository contains code specific to SubjuGator.

estimated by the filter, the vehicle maintains an accurate estimate of its state.

2. Trajectory Generator and Controller

The trajectory generator and controller work together to move the vehicle to its desired waypoint. The trajectory generator is based on a nonlinear filter that produces 3rd-order continuous trajectories given vehicle constraints on velocity, acceleration, and jerk [3]. The constraints can be adjusted on each vehicle DOF, potentially being asymmetric. The generator can be issued any series of position and/or velocity waypoints, allowing greater flexibility of commanded inputs, while guaranteeing a continuous output and remaining within vehicle constraints [4].

The controller is responsible for keeping the vehicle on the trajectory and correcting for disturbances such as drag and thruster variation. Our trajectory tracking controller implements a nonlinear Robust Integral of the Sign of the Error (RISE) feedback control structure [5]. This controller was developed by a past MIL member and outperforms most tracking control designs available in literature.

3. Mission Planner

The vehicle's mission planner is responsible for high level autonomy and completing the competition tasks. It is implemented using a Python coroutine library and custom ROS client library (txROS[3]) to enable writing simple procedural code that can asynchronously run tasks with timeouts, wait for messages, send goals, etc., thus enabling a hierarchical mission structure that can concisely describe high level behaviors, such as commanding waypoints and performing visual feedback.

4. Vision Processing

Traditional techniques, namely image segmentation via adaptive thresholding followed by contour analysis, are used to find many of the competition elements. Moreover, techniques such as Laplacian of Gaussian and difference of Gaussians are used for blob detection.

Deep neural networks are also used to assist traditional computer vision techniques. In particular, the architecture known as *Faster Regions with Convolutional Neural Networks* (Faster RCNN) [6] is used, which is trained by using transfer learning and with the inception v2 model [7]. After the feedforward step, Faster RCNN returns regions of interests (ROI), which are then passed through traditional computer vision techniques for further verification and segmentation. The training data is labelled by the team using a collaborative labeling tool for machine learning: LabelBox (see Figure 6).

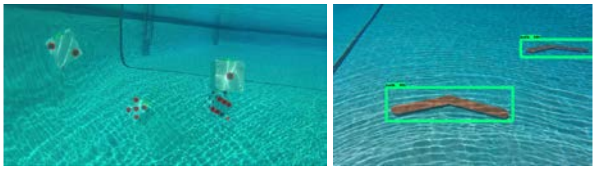

Fig 6. Sample vision processing; left: Laplacian of Gaussian; right: Faster RCNN.

After segmentation, the three-dimensional pose of the object is estimated by using a priori knowledge of either the distance or the size of the object, or by using multiple observation points and a least squares cost function. Moreover, a stereo camera system is used to further check these estimates. Using one Point Grey Chameleon camera and one e-con See3CAM CU20 we generate robust 3-D information of our world when operating in favorable conditions. Internal camera calibration and distortion parameters are obtained using [8].

5. Imaging Sonar Processing

A ROS Driver was developed to abstract the closed-source Blueview Software Development Kit (SDK), enabling ROS to communicate with the Teledyne Blueview P900-130. The driver produces images along with range profiles in ROS.

Due to the nature of acoustics, error and noise is prevalent, leading to the development and adaptations of filtering algorithms. Using the returned ranges and the estimated SubjuGator pose, a 3-D point cloud is constructed, populating the world-frame over time. Statistical outlier removal is used to remove noise from the constructed point cloud. The resulting filtered point cloud is then

---

[3] The txROS GitHub is available at https://github.com/txros/txros.

examined for Euclidian clusters, with parameters such as maximum and minimum size. After clustering points into objects, higher-level mission software can interpret and react to 3-D position estimates and size, as shown in Figure 7.

Moreover, with the presence of a global filtered point cloud, tasks such as obstacle avoidance using Oct-tree representation for occupancy grids along with correcting for global state drift with simultaneous localization and mapping (SLAM) become possible.

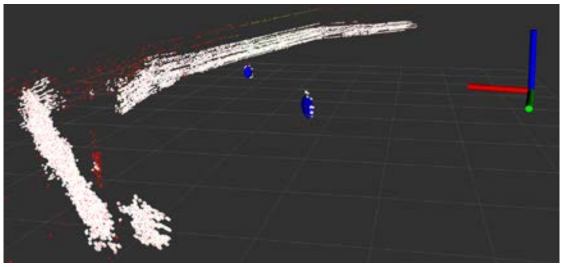

Fig 7. Populated point cloud, filtered point cloud in white, and clustered objects represented by blue ellipsoids. The two objects represent the poles of the start gate.

III. Experimental Results

In order to accumulate experience, foster teamwork, build work ethic, progress software, and insure stability in hardware, we scheduled weekly pool testing. Additional pool testing was scheduled when needed. As we approached competition deadline, pool testing was conducted every few days. Thus, minimal testing time was set by our weekly pool tests, and additional test time was determined by progress, issues needed to be resolved, and team or facility availability. Teamwork organization was difficult, especially in the beginning. This schedule served us well, providing consistency, efficiency, and productivity. The team started blogging weekly progress, which helped us to organize thoughts, discover weaknesses, and to establish priorities. Moreover, to ensure productive usage of the allotted pool testing time, hardware components and designs were tested beforehand, perception software was tested against recorded data from previous pool testing, while missions were tested with the seamlessly integrated Gazebo Simulator. Additionally, the team met once a week with faculty advisors to discuss ideas, designs, and algorithms. Importantly, our pool testing program was prepared in advance following input from faculty, team members, out-of-water component testing, and previous test results. Overall, due to proactive decisions and discussions, along with prioritization, communication, and planning, the team was able to effectively balance engineering and experimentation (and coursework).

IV. ACKNOWLEDGMENTS

The University of Florida MIL SubjuGator team would like to thank everyone who has supported us throughout the year, including the University of Florida's Electrical and Computer Engineering Department, Mechanical and Aerospace Engineering Department, and the students and faculty in UF's CIMAR (Center for Intelligent Machines and Robotics). We would also like to thank several former students who have contributed to our team financially and with advice. Each of the following corporate sponsors were gracious in assist with both monetary and product donations:

- Platinum Sponsors: Harris Corporation
- Gold Sponsors: UF Dept. of Electrical and Computer Engineering, UF Dept. of Mechanical and Aerospace Engineering, JD^2
- Silver Sponsors: Texas Instruments, Lockheed Martin, SolidWorks, IEEE Gainesville Section, Altera, Advanced Circuits, DigiKey

Finally, the Machine Intelligence Laboratory is also honored to have recently been sponsored by the University of Florida's Herbert Wertheim College of Engineering, mostly in support of our Maritime RobotX Challenge 2016 champion, NaviGator AMS (autonomous maritime system).

The latest SubjuGator developments can be found on our web page www.subjugator.org or by following us on twitter @SubjuGatorUF.

VII. REFERENCES

[1] P. Miller, J. Farrell, Y. Zhao, and V. Djapic, "Autonomous underwater vehicle navigation," *IEEE Journal of Oceanic Engineering*, Vol. 35, No. 3, pp. 663–678, July 2010.

[2] J. Nezvadovitz, "Symmetric Propeller and Nozzle Design for a Marine Robot," *University of Florida Journal of Undergraduate Research*, Vol. 17, Issue 2, Spring 2016.

[3] L. Biagiotti and C. Melchiorri, *Trajectory Planning for Automatic Machines and Robots*. Springer, 2008.

[4] P. Walters, R. Kamalapurkar, F. Voight, E. Schwartz, W. Dixon, "Online Approximate Optimal Station Keeping of a Marine Craft in the Presence of a Current." *IEEE Transactions on Robotics*, Vol. 30, No. 2, pp 486-496, April 2018.

[5] N. Fischer, S. Bhasin, and W. Dixon, "Nonlinear control of an autonomous underwater vehicle: A RISE-based approach," *IEEE Transactions on Robotics*, Vol. 30, No. 4, pp. 845-852, 2014.

[6] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, No 6, pp. 1137-1149, 2017.

[7] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision", *Arxiv.org*, 2015. [Online]. Available: https://arxiv.org/abs/1512.00567.

[8] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," *IEEE Proc. International Conference on Computer Vision*, Vol. 1, pp. 666–673, 1999.

APPENDIX

Component	**Vendor**	**Model/Type**	**Specs**	**Cost (if new)**
Buoyancy Control	No hardware		Positively buoyant; thrusters control depth	
Frame	Dragon plate	Carbon fiber	Space frame	
Frame	Student Design	Aluminum	Frame core	
Waterproof Housing	Student Design	Aluminum	Main vessel	
Waterproof Housing	Student Design	Aluminum	Navigation vessel	
Waterproof Housing	Student Design	Aluminum	Pneumatic vessel	
Waterproof Housing	Student Design	Aluminum	Downward camera vessel	
Waterproof Housing	Student Design	Aluminum	ESC Vessel	
Waterproof Connectors	SubConn	Wet-connect	External wet-mate connectors	
Waterproof Connectors	SEACON	Wet-connect	External wet-mate connectors	
Thrusters	VideoRay	M5		
Motor Control	VideoRay	Built-in to M5	48 VDC @ 76 Watt, RS-485	
Propellers	VideoRay	Stock on M5		
Thrusters	Blue Robotics	T200		$169
Motor Control	Blue Robotics	Basic ESC	7-26v, 30amp, PWM	$25
Propellers	Blue Robotics	Stock		
Actuators (Pneumatic)	Clippard		Double acting ½" bore, ½" stroke	
Battery	MaxAmps	LiPo	LiPo 5450 6S 22.2v	
Converter	Student Design		48 V to 24 V	
Converter	Student Design		Power over Ethernet (POE)	
Regulator	Many			
CPU	SuperMicro	Intel Xeon D-1540	COTS mini-ITX motherboard	
CPU	STMicroelectronics	STM32F4	Cortex-M4 ARM	
CPU	SparkFun	Teensy 3.2	Freescale K20P64M72SF1 (for Blue robotics motor PWM)	
Internal Comm Interface	Student-designed		TIA-485	
Internal Comm Interface	Various		USB	

APPENDIX

Component	Vendor	Model/Type	Specs	Cost (if new)
External Comm Interface			Ethernet	
Programming Language 1	C++			
Programming Language 2	Python			
Compass	PNI	TCM MB		
Inertial Measurement Unit (IMU)	Sensonar	STIM300	9-axis	
Doppler Velocity Log (DVL)	Teledyne	Explorer	600kHz	
Camera(s)	Point Grey	BlackFly	5.0 MP, 22fps	
Camera(s)	Point Grey	Chameleon	1.3 MP, 18fps, USB 2.0	
Camera(s)	e-con Systems	See3CAM CU20	2.0 MP HDR, HD at 45fps, USB 3.0	$89
Imaging Sonar	Teledyne	Blueview P900	130-degree FOV, 900kHz, 2-60 meters	
Hydrophones	Teledyne Reson	TC 4013	4	
Hydrophone components	Sylphase	Custom	Former Student-designed data acquisition PCB	
Hydrophone components	Analog Devices	ADAR7251	4-Channel, 16-Bit, Continuous Time Data Acquisition ADC	
Manipulator	Student Design			

Software Component	Libraries	Algorithm
Vision	OpenCV	Canny Edge Detection, Thresholding, Optical Flow
Machine Learning	TensorFlow, Keras	Faster RCNN
Acoustics	Scipy, numpy	Phase difference, Least Squares
Localization	Eigen	Unscented Kalman Filter
Mapping	PCL, OpenCV	Statistical Outlier Remove, Euclidean Clustering
Communication	ROS	

Team information	
Team size (number of people)	17
Electrical engineering expertise ratio	35% (6/17)
Mechanical engineering expertise ratio	41% (7/17)
Computer science/engineering expertise ratio	24% (4/17)
Testing time: simulation	250+ hrs
Testing time: in water	50+ hrs