SubjuGator 2019: Design and Implementation of a Modular, High-Performance AUV

D. Volya, J. Brown, A. Puldio, N. Suhlman, E. M. Schwartz
dvolya@ufl.edu, jaxonbrown@ufl.edu, andrespulido@ufl.edu, grymestone@ufl.edu, ems@ufl.edu

***Abstract* – Here we present SubjuGator 2019, an updated version of the eighth generation of SubjuGator. SubjuGator was made by mostly undergraduate students in UF's Machine Intelligence Laboratory (MIL). The current version of our autonomous underwater vehicle (AUV) focuses on robust control, hardware improvements, and software innovations. In particular, this model includes a controller area network (CAN) bus, onboard general-purpose graphics processing unit (GPGPU), deep learning and point cloud processing, new thrusters, and other challenge-specific designs. 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 22 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 22nd annual competition,

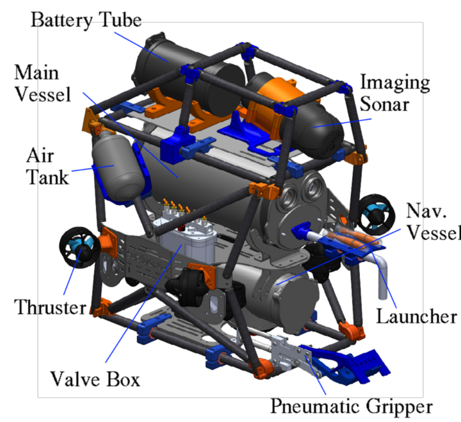

Fig 1. SubjuGator 8.

namely in minimizing problems and accentuating advantages. One example involves unreliable thrusters and noisy electronic communication channels, which hindered maneuverability and control. Decisions were made to replace SubjuGator's thrusters, switch to using CAN bus for communication, and introduce a GPGPU. The GPGPU required more room in the main vessel, leading to the removal of the batteries from the main vessel to an external pressure vessel.

Compared to previous years, this years' tasks require an increasing level of object detection and pose estimation, thus providing an additional source of consideration for strategy decisions. The 22nd 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 correct vampire and going through the smaller portion of 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 monocular and stereo depth estimation. Upon finding a region of interest, image data is used for further validation and discovery of intrinsic properties, such as the type of vampire or location of target hole. Upon correct discovery, SubjuGator preforms defined maneuvers to solve the task. To minimize error, software design employs a number of fallbacks, filtering, and error correction techniques.

Finally, since the team this year consists of predominantly new members, we prioritized mastering a select number of feasible tasks while maximizing test time. Thus, many of the design strategies took into account team capabilities and experience, while working on top of the infrastructure left behind from previous teams.

II. Vehicle Design

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 a vectored thruster configuration, SubjuGator 8 possesses 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.

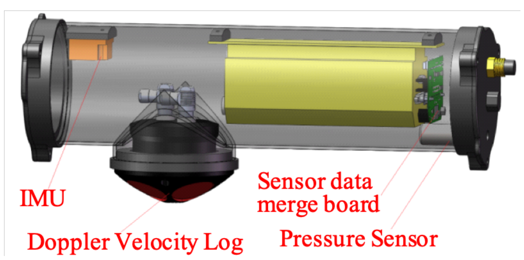

Fig 2. Model of the navigation vessel.

2. Electronics and Kill System

SubjuGator 8 now features a CAN-based communication system to interface software and electrical systems. Using a student-designed USB-to-CAN interface, other student-designed systems include the following PCBs: thrust and kill, battery monitor, system status, and actuator control; each can communicate with the motherboard over a single, more reliable and noise resistant CAN bus. Since this bus is more noise-tolerant, SubjuGator has had fewer communication issues with hardware than in the past several years. This new system also decreases the size requirements of communication systems, helping us save space for the additional electronics in the main vessel.

The new kill system features hall-effect based kill triggers (returning to a design last used in SubjuGator 7). A hall-effect sensor is mounted close to the outer wall of the pressure vessel. When a magnet with the proper polarization is placed against the outside wall of the pressure vessel, the hall effect sensor is triggered and relays are tripped, cutting power to the thrusters. When the magnet is reversed, the hall effect sensor releases the hard kill, allowing the software to restore power to the thrusters when ready.

3. Power Vessel and Water Cooling

With the switch to Blue Robotics T200 thrusters, external electronic speed controllers (ESCs) were added to the main vessel, and the power system was migrated from 48 V to 24 V. Additionally, due to the increasing demands in software, a GPGPU was added to offload certain operations from the CPU. To facilitate the space requirements of the ESCs and GPGPU in the main vessel, the two 24 V Li-Po batteries are now housed in an external pressure vessel, with an additional student-design cell voltage monitoring board. To quickly dissipate heat, water cooling covers for the CPU, GPGPU, and ESCs (with a custom water block) were added. Bulkheads and Dowty Bonded Seal Washers enable water to be pumped to the outside of the vessel where it is dissipated through a radiator.

4. 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.

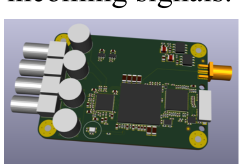

Fig 3. Passive sonar PCB.

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.

5. Pneumatics System and Actuators

SubjuGator 8 integrates three types of independently operated pneumatic mechanisms into its design (grabber, torpedo launcher, and marker dropper). The mechanisms are used to complete mission specific tasks and are controlled via the six 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.

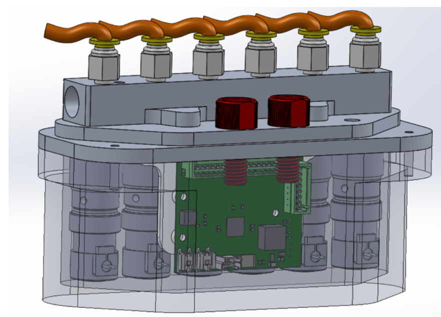

Fig 4. Pneumatic solenoid housing/controller.

The entire system is powered by an air tank, which is regulated down to a working pressure of 100 psi via two in-line regulators. The actuator pressure vessel also includes a student designed actuator board, which drives the solenoids while communicating with the main computer.

The grabber is powered by a linear pneumatic cylinder mounted to plates. Since this year's tasks the submarine only needs to manipulate PVC tubes of different diameters, the cylinder actuates a 3D printed ABS gripper designed specifically to manipulate PVC from various angles and shapes.

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 provide tutorials and documentation for all parts of the code, to aid future members and further encourage external use. Our ROS

Teledyne Blueview Driver, along with the rest the software is open-sourced, and available on GitHub[1].

1. State Estimator

The state estimator uses an inertial navigation system (INS) and an unscented Kalman filter. 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 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.

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 proportional-integral-derivative (PID) controller with feed-forward velocity and acceleration terms to anticipate drag and buoyancy.

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) 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.

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) [4] is used, which is trained by using transfer learning and with the inception v2 model [5]. 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 5).

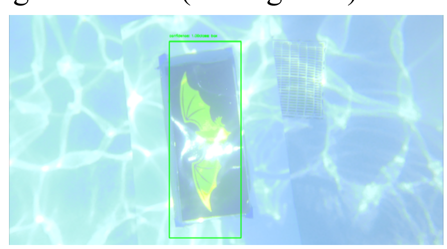

Fig 5. Sample vision processing for Garlic Drop using Faster RCNN.

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; by using multiple observation points and a least squares cost function; or by processing a 3-D point cloud either from a stereo camera system or imaging sonar. Additionally, this year, by modeling object motion, a dynamic scene can be reconstructed by an unsupervised learning technique [6]

[1]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.

which enables monocular depth predication (Figure 6) and serves as an initial guess for object pose prediction. 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 [7].

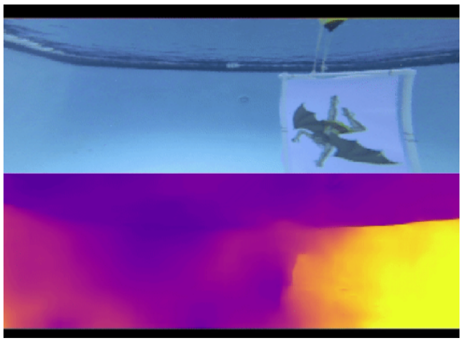

Fig 6. Original image (top) and predicted monocular depth (bottom).

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 examined for 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.

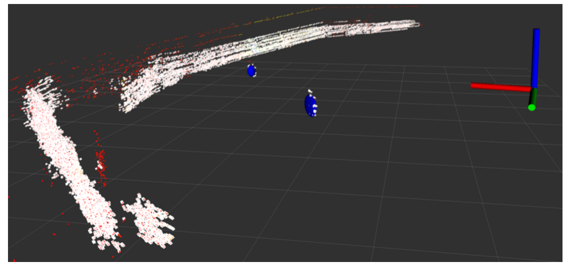

Fig 7. Populated point cloud, filtered point cloud in white, and clustered objects represented by blue ellipsoids. The two objects represent the poles of the start gate.

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. 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:

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

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

[3] 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.

[4] 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.

[5] 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.

[6] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova. "Depth Prediction Without the Sensors: Leveraging Structure for Unsupervised Learning from Monocular Videos", *Arxiv.org*, 2018. [Online]. Available: https://arxiv.org/abs/1811.06152.

[7] 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.

Subjective Measures			
	Maximum Points	Expected Points	Points Scored
Utility of team website	50	45	
Technical Merit (from journal paper)	150	145	
Written Style (from journal paper)	50	45	
Capability for Autonomous Behavior (static judging)	100	100	
Creativity in System Design (static judging)	100	90	
Team Uniform (static judging)	10	10	
Team Video	50	50	
Pre-Qualifying Video	100	100	
Discretionary points (static judging)	40	39	
Total	650	624	
Performance Measures		Total: 4450	
	Maximum Points		
Weight	See Table 1 / Vehicle	120lbs	
Marker/Torpedo over weight or size by <10%	minus 500 / marker		
Gate: Pass through	100	100	
Gate: Maintain fixed heading	150	150	
Gate: Coin Flip	300	300	
Gate: Pass through 60% section	200	200	
Gate: Pass through 40% section	400		
Gate: Style	+100 (8x max)	400	
Collect Pickup: Crucifix, Garlic	400 / object	100	
Follow the "Path" (2 total)	100 / segment	100	
Slay Vampires: Any, Called	300, 600	600	
Drop Garlic: Open, Closed	700, 1000 / marker (2 + pickup)	700	
Drop Garlic: Move Arm	400	200	
Stake through Heart: Open Oval, Cover Oval, Sm Heart	800, 1000, 1200 / torpedo (max 2)	1000	
Stake through Heart: Move lever	400	200	
Stake through Heart: Bonus - Cover Oval, Sm Heart	500		
Expose to Sunlight: Surface in Area	1000	1000	
Expose to Sunlight: Surface with object	400 / object		
Expose to Sunlight: Open coffin	400		
Expose to Sunlight: Drop Pickup	200 / object (Crucifix only)		
Random Pinger first task	500	500	
Random Pinger second task	1500	1000	
Inter-vehicle Communication	1000		
Finish the mission with T minutes (whole + factional)	Tx100		

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	Blue Robotics	Aluminum	Downward camera vessel	
Waterproof Housing	Student Design	Aluminum	Power Vessel	
Waterproof Connectors	SubConn	Wet-connect	External wet-mate connectors	
Waterproof Connectors	SEACON	Wet-connect	External wet-mate connectors	
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		Power over Ethernet (POE)	
Regulator	Many			
CPU	ASRock	ASRock Z390M-ITX	mini-ITX motherboard	$140
CPU	Intel	i9-9900k		$500
GPGPU	Nvidia	GTX 1080		$600
Internal Comm Interface	Student-designed		CAN	
Internal Comm Interface	Various		USB	
External Comm Interface			Ethernet	
Programming Language 1	C++			
Programming Language 2	Python			
Compass	PNI	TCM MB		
Inertial Measurement Unit (IMU)	Sensonar	STIM300	9-axis	

APPENDIX

Component	Vendor	Model/Type	Specs	Cost (if new)
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	Time of Arrival, Least Squares
Localization	Eigen	Unscented Kalman Filter
Mapping	PCL, OpenCV	Statistical Outlier Remove, Euclidean Clustering
Communication	ROS	

Team information	
Team size (number of people)	12
Electrical engineering expertise ratio	33% (4/12)
Mechanical engineering expertise ratio	33% (4/12)
Computer science/engineering expertise ratio	33% (4/12)
Testing time: simulation	290+ hrs
Testing time: in water	40+ hrs