SubjuGator 2020: Design and Implementation of a Modular, High-Performance AUV

A. Pulido, M. Rawson, R. Fabien, J. Mejia, E. M. Schwartz
andrespulido@ufl.edu, juan.mejia@ufl.edu, marshallrawson@ufl.edu, rosemondfabien@ufl.edu, ems@ufl.edu

***Abstract* – Here we present SubjuGator 2020, the ninth 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 adaptive control, hardware improvements, and software innovations. This model includes a controller area network (CAN) bus, onboard general-purpose graphics processing unit (GPGPU), deep learning and point cloud processing, 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 23 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 new generation SubjuGator 9 vehicle.

Moreover, for the past few AUVSI RoboSub competitions, SubjuGator 8 served as the primary development and competing platform. For the 23rd annual competition, Subjugator 9 will join the previous iteration to mark the first year in which two submarines will be deployed at the same time. The design goal for Sub9 was to divide the tasks between each submarine to maximize the amount of possible points the team can achieve.

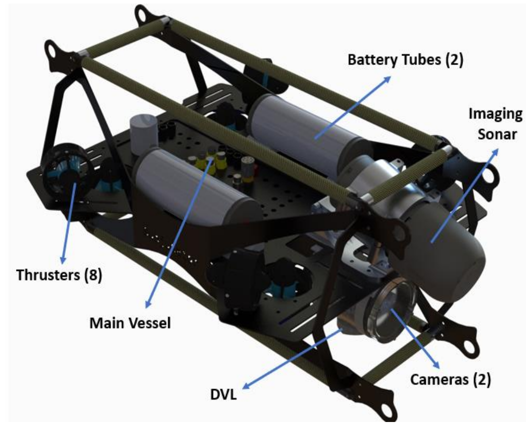

Fig. 1. Subjugator 9

Since the need for object detection and pose estimation relative to an object, to later perform some action around said object, has been increasing every year, different sensor are used on both submarines two perform all the tasks. SubjuGator 9 searches for regions of interest satisfying certain constraints such as the start gate by using an Active Imaging Sonar and then and a monocular and a stereo depth estimation. Subjugator 8 employs the use of sonar and cameras to locate regions of interest, and upon correct discovery, preforms defined maneuvers to solve the task in which a pneumatic manipulator is needed, either object manipulation or torpedo shooting. To minimize error, software design employs several filtering, and error correction techniques.

II. Vehicle Design

The ninth generation SubjuGator AUV has the capabilities to meet and exceed the challenges of the competition.

A. Hardware Design

A major feature of SubjuGator 9 is the ability to sustain operation after a failure, which can be of mechanical, electrical, or software. 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 if 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.

To unify the different modules into a durable, light weight, and modular platform, a chassis was constructed from anodized Aluminum sheet metal sections which are bent to increase rigidity and durability of the system. The central plate or "Spline" provides most of the modular system attachment points while a set of carbon fiber tubes at the top and bottom provide system attachments for mounts designed for Subjugator 8. This structure provides several 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

B. Electronics System

Subjugator 9 consists of a robust set of industry standard as well as student designed electronic components. The heart of the electronic system features the main processing unit which incorporates a Jetson NVIDIA and NUC CPU. With the combined efforts of these devices, Subjugator 9 is capable of processing at high bandwidths while having the flexibility to interface with many devices.

Peripheral to the main processing unit is a whole suite of devices to aid in navigation, cooling, safety, and communication. The cooling system includes fans and a water pump with appropriate heat sinks to keep Subjugator 9 running at optimal thermal efficiency. The navigation system includes a doppler velocity log and inertial measurement unit. The communication system has both inter and intra communication mechanisms. The Tether allows an ethernet interface that is leveraged when a hard-wired connection with Subjugator 9 is necessary, such as testing. The Acoustic modem allows the robot to wirelessly communicate with other aquatic devices. This is a crucial aspect in the system because it will facilitate inter-robot communication in the future.

The safety system incorporates both battery monitoring and emergency shut-off components. The battery monitoring module is one of the student-designed circuit boards that allows Subjugator 9 to monitor power consumption. This is important when deciding when to safely change batteries on the robot. Furthermore, the thrust/kill board is another student designed board that allows the robot to control thrusters as well as cut power to them, creating a safe shut-down feature. If all else fails, there is also a manual shut-off feature that can be triggered by placing a magnet on the appropriate location of the vehicle; thereby cutting power to the thrusters. This action is facilitated by hall-effect sensors and relays.

Other student designed boards include a power merge board, servo controller, and system status board. The power merge board safely combines the power of two 24 V batteries to create one 24 V power source that is routed throughout the rest of the system. The servo controller allows the robot to take on a myriad of servos, of varying power, and command them with utmost controllability. The system status board provides real-time diagnostics of Subjugator 9 through CAN and wireless interfaces.

The remaining components of the Electrical system include other commercial off the shelf products. The power system consists of two 24 V Lipo batteries with safety several relays connecting to eight Blue Robotics T200 thrusters and the power merge board mentioned above. The system also includes an Imaging Sonar that is a Sylphase designed device that gives the vehicle the capacity to accurately track a point source of sound in an aquatic environment.

There are several other components throughout the electrical system which include, but are not limited to, cameras, electronic speed controls, and pressure sensors. A comprehensive visual of the Subjugator 9 Electrical System is shown in Fig. 2.

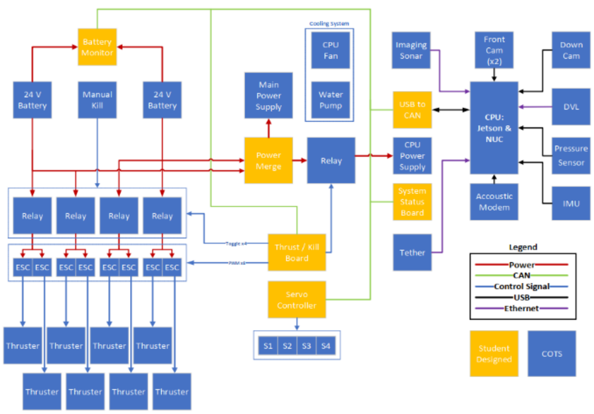

Fig. 2: Electrical Architecture

C. Software Design

SubjuGator 9'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*

[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.

(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.

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] which enables monocular depth predication 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].

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. 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.

III. Experimental Results

Due to limitations imposed by physical distancing, mechanical design and manufacturing planning for the new submarine was prioritized to start the manufacturing process (following social distancing protocols) as soon as access to the laboratory is granted. Software has been tested against recorded data from previous pool testing, while missions were simulated with the seamlessly integrated Gazebo Simulator (Fig. 3).

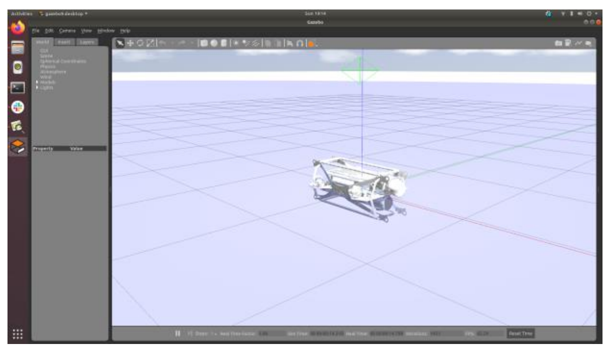

Fig. 3. Subjugator 9 simulated in Gazebo

Additionally, using video conference tools, the team met once a week to discuss ideas, designs, and algorithms. Faculty advisors were able to provide recommendations and feedback of the team's progress once a month. Overall, due to proactive decisions and discussions, along with prioritization, communication, and planning, the team was able to effectively balance remote design, software testing and online coursework.

IV. ACKNOWLEDGMENTS

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.

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	Intel	NUC 10	Intel® NUC 10 Performance Mini PC - NUC10i5FNKPA	$140
CPU	Nvidia	Jetson	Jetson Xavier NX	$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	
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	