SubjuGator 2024: Design and Implementation of a Modular, High-Performance AUV

L. Bjellos, L. Bonilla, C. Brown, A. Fernandez, A. Hamdan, D. Parra, Dr. E. M. Schwartz

***Abstract*— The Machine Intelligence Lab (MIL) at the University of Florida (UF) has continued development of the SubjuGator 8 autonomous underwater vehicle for 9 years. Our design strategy for this long-life system is to manage complexity through modularity and adaptability. Leveraging an existing vision processing infrastructure, basic vision tasks are prioritized for the competition, followed by locating the random pinger tasks, and manipulation for the bins, torpedo task, and octagon. Additionally, the design changes, testing, competition, and teamwork strategies discussed were adapted based on previous experience, changes to the competition rules, and the structure of our team.**

I. COMPETITION STRATEGY

SubjuGator's approach to addressing the trade-off between increasing system complexity and competition performance is centered around leveraging the foundation that SubjuGator 8 (Figure 1) has built from its continued development. This was done to ultimately increase performance and reliability of the system.

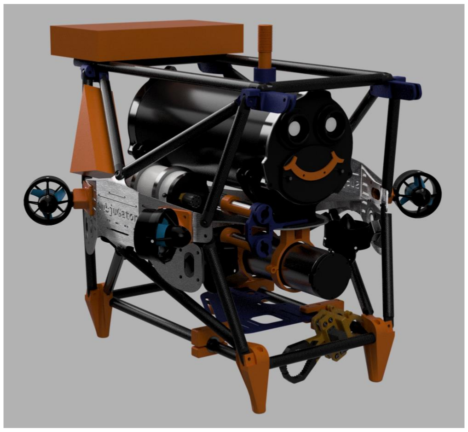


Fig. 1: SubjuGator 8 AUV

SubjuGator 8's design strategy has been to maximize modularity and adaptability. With that in mind, our strategy was to prioritize the competition tasks based on possible points to earn and our presumed ability to reliably earn them (see Figure 2). This provided us with an ideal score for which to aim. We then spent our time testing and refining the subsystems required to accomplish those tasks and adding features as needed.

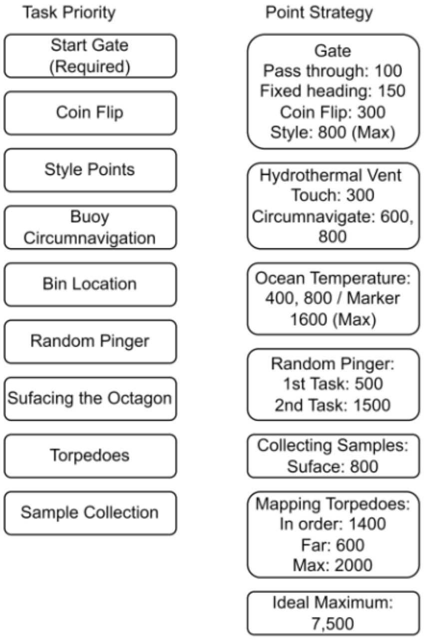


Fig. 2: Task priority and point strategy.

Through this process, we have decided to prioritize simple vision processing tasks such as Start Gate and Buoy Circumnavigation, then to integrate the hydrophone system for the pinger task, and servo actuators for the remainder of the points. SubjuGator is not currently aiming to perform the collection task at the competition. Time permitting, we will attempt them after satisfactory completion of higher-priority tasks.

To manage the complexity of a vehicle that can accomplish our goals, the design of SubjuGator 8 is divided into different sub-systems of related components (Figure 2). The sub-systems are then developed and tested independently before being integrated and tested together.

The team is divided into three teams that focus on Mechanical, Electrical, and Software design. Following this division and abstraction, the project is broken down into actionable tasks assigned to team members based on their strengths. We use a combination of GitHub issues and projects and Gantt charts to manage assignments. Our teams meet regularly to track the progress of the project.

We place emphasis on systems that have maximum impact for our score based on our task priority list, and divide responsibilities based on which subsystem of the vehicle that contains (Figure 3).

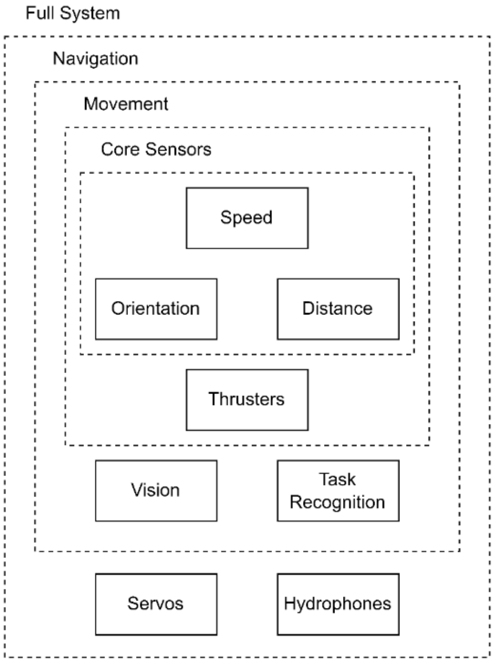


Fig. 3: SubjuGator 8 high level system overview.

For example, the highest priority task is the Start Gate, and we want to achieve our ideal maximum of 800 points for that task. To do this, Movement, Vision, and Task Recognition must all work in tandem. Movement involves the integration of our IMU, DVL, and pressure sensor data to orient the sub and command the thrusters. We test that the sub is following our commands and working as intended. Concurrently, our Vision and Task Recognition system would be trained to detect task features and determine an appropriate response. Once each is working individually, tasks are combined and then iterated as a system to improve the team's chances of completing those tasks during competition.

II. DESIGN STRATEGY

SubjuGator 8 has been developed and improved upon by many different members over many years. It was designed with modularity and redundancy to enforce reliability and adaptability while being worked on by different generations of teams.

Modularity improves the testability of our components and subsystems. We test individual components before adding them together to form a larger subsystem. Keeping in mind that a small issue in one component can be magnified as information spreads throughout the system, this modular approach means that we can pinpoint where an issue is occurring and swiftly identify what has gone wrong. Designing with testing and debugging in mind creates an efficient workflow.

Redundancy is included in our design strategy to defend against any non-conformances, interference, camera occlusion, and physical disturbances. All electronics hulls are sealed with double O-rings to protect our electronics. The vehicle can run and maintain power with one of the two batteries. At times (especially when the waterbed has a large vertical drop), the DVL data may be unstable, so the vehicle will weigh the IMU data via the state estimator (unscented Kalman filter) more heavily to compensate.

A. *Mechanical*

1) *Design of SubjuGator 8 Platform*

The mechanical structure of SubjuGator 8, based on a slotted anodized aluminum spine and carbon fiber rod frame, is extremely adaptable and has led to a continuously advancing submarine. The system's design began in 2014 and was first used in competition in 2015, which is a testament to its reliability. The spine was machined to include slots that allow various accessory components, including mounts for ancillary hulls, mechanisms, and a water-cooling radiator. Over the last several years of development, rapid prototyping and additive manufacturing has been leveraged to expeditiously redesign and reconfigure to fit the goals of the competition and capabilities of the team. This led to the creation of prototype, wet-side, and dry-side 3D printing standards that the team uses for all its designs.

The frame also includes mounts for eight Blue Robotics T200 thrusters, which are incorporated into a redundant design that allows for a full six degrees of motion even if both a single vertical and horizontal thruster fail. The system is designed to detect a failure and compensate using the remaining thrusters to ensure proper motion. On an aging platform, this has saved the team valuable testing time when a thruster had suddenly failed due to surpassing its functional service life.

2) *Hull Design*

SubjuGator 8's electronics are housed in one main and three ancillary hulls (Figure 4). All tubes are designed to have modular attachment points and can be moved to meet the expectations demanded by each year's challenges.

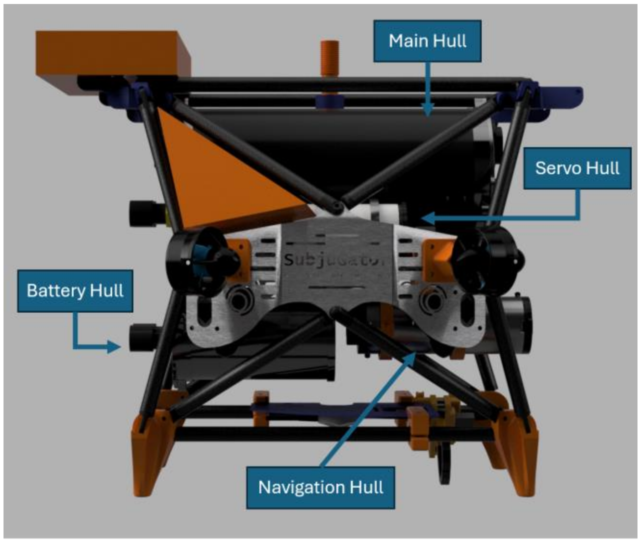


Fig. 4: SubjuGator electronics hulls.

The custom main hull houses the main computer and power electronics of the vehicle. It serves as the heart of vehicle and connects to the ancillary hulls, thrusters, and other auxiliaries via wet link connectors and epoxy-sealed penetrators.

The custom battery hull houses LiPo batteries and voltage alarms and is sized to house enough energy for two competition runs. It was chosen to be separated from the main hull to reduce water ingress and be thermally isolated.

The navigation hull and servo hull are COTS watertight tubes purchased from Blue Robotics. These were chosen to be separated from the main hull to isolate magnetic interference and increase accessibility. Much like the battery tube, this has the added benefit of reducing water ingress and thermally isolating the delicate sensors and controllers in the tubes. Both hulls also make use of a quick-release system that allows for greater accessibility of electronics inside, so that the vehicle can continue operation (with varying limitations) during maintenance. This was a high priority for the team, as it meant that something like a servo failure would not halt the team's limited testing slots at the competition. The attempt would both maximize time in the water and allow the supporting team members to fix the smaller issue on land.

3) *Design of Mechanisms*

SubjuGator 8 includes three independently operated electronic servo mechanisms: a ball dropper, torpedo launcher, and gripper. Each mechanism has been designed to be responsible for their own mission specific task (namely Ocean Temperature, Mapping, and Sample Collection). Each mechanism uses a high-torque, waterproof, brushless servo that is powered and controlled through the servo hull.

The ball dropper (Figure 5) uses a powered rotating drum to drop a marker upon command. The drum has a reloading port for fast reloads and a small hole at its base to ensure that only one marker drops at a time. The drum is mounted directly to the servo for simplicity, and the enclosure of the drum is mounted rigidly to the vehicle's frame.

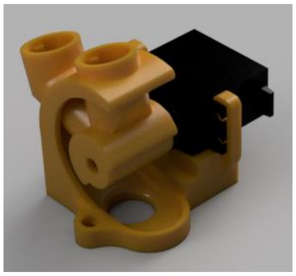


Fig. 5: Servo-actuated ball dropper.

The torpedo launcher (Figure 6) consists of two see-through barrels and uses a quick-loading lockable spring system that presses the torpedoes against a retention block. Due to the danger associated with pointed projectiles, it was the team's highest priority to create a safe and reliable system. Due to the positioning of the servo and the release system, there is no potential for a misfire as the torpedoes will remain in a locked state until sufficient torque is exerted by the servo to release the torpedo from the retention block.

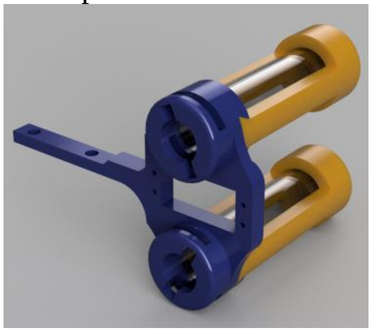


Fig. 6: Servo-actuated torpedo launcher.

The gripper (Figure 7) makes use of interlocking jaws mounted to a rigid frame. The lower jaw is rotated by a servo. Threads cut into the jaws to allow for simultaneous opening and closing. The jaws are easy to take off and replace and are additively manufactured to allow for rapid design iteration.

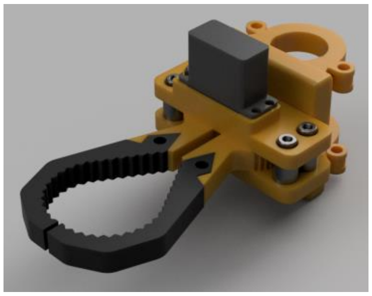


Fig. 7: Servo-actuated gripper.

B. Electrical

The electrical system in SubjuGator 8 (Appendix B) implements the design strategy by ensuring reliable power distribution and communications.

Power is provided through two LiPo batteries. Attached to these batteries are monitors that audibly announce when the batteries are low on charge, to prevent overuse. Monitoring in this way increases the lifespan of the batteries and their effective charges. The system can be powered by a single battery if the other battery malfunctions or is low on charge.

The electrical components that must dissipate the most heat, the main computer and the electronic speed controls (ESCs), are cooled with a custom water-cooling system. The heat is transferred from inside the main compartment and is dissipated to the surrounding water. The system can run at normal temperatures without this cooling, and it is present to provide extra durability and maintain optimal thermal conditions within the main compartment.

C. Software

SubjuGator 8's software stack is built on the Noetic version of the Robot Operating System (ROS). Our stack has grown to over 60+ ROS packages, all of which are open-source[1], allowing other teams to share the benefits from our work. Many of our packages feature extensive documentation, and we are constantly improving their documentation and features.

1) State Estimator

The state estimator uses an inertial navigation system (INS) and an unscented Kalman filter [1] operating on manifolds for more efficient handling of attitude singularities. The INS integrates inertial measurements from the IMU, producing an orientation, velocity, and position prediction. 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 DVL.

2) Trajectory Generator

The trajectory generator is based on a nonlinear filter that produces 3rd-order continuous trajectories given vehicle constraints on velocity, acceleration, and jerk [2]. Our trajectory tracking controller implements an adaptive proportional-integral-derivative (PID) controller. The integral gain can be configured to be adaptive to account

---

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

for wind up errors or be held statically if a suitable value has been found.

3) *Mission Planner*

The vehicle's mission planner is responsible for high level autonomy and completing the competition tasks by enabling asynchronous support in Python. This library has been developed and used for the past ten years and is continually being augmented and improved. We recently rewrote the module to use our custom *axros* Python package, an interface between *asyncio* and ROS 1.

4) *Vision Processing*

Traditional techniques, namely image segmentation via adaptive thresholding, followed by contour analysis, are used to find many of the competition elements, notably the orange path markers and explicit contours of objects in the underwater environment.

Deep neural networks are also used to assist traditional computer vision techniques. In particular, the architecture known as *You Only Look Once* (YOLO) [3] is used, which is trained by using transfer learning and with the darknet YOLOv7 model [4]. After the feedforward step, YOLO returns bounding boxes and object classifications. The training data is labeled by the team using a collaborative labeling tool for machine learning called *LabelBox* [5].

Additionally, 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 a front-facing camera, we generate robust 3-D information of our world when operating in favorable conditions. Internal camera calibration and distortion parameters are obtained using a standard printed calibration board viewed from multiple frames [7].

III. TESTING STRATEGY

Our overall test plan is to work through our list of prioritized tasks, with the goal of improving reliability. We test our components and systems using a structured feedback process where, after each session or step in design, we summarize and reflect on what went right and what can be improved (Figure 8).

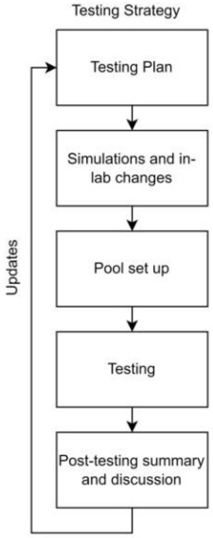


Fig. 8: General testing strategy.

Throughout development, the team has consistently tested in-water at least weekly to keep a pulse on our progress. This schedule informs the team of what each team should be putting their efforts into and what changes need to be made to the plan.

We enter each testing session with a clear goal of what we hope to accomplish and what we will be testing. We standardized the items we need to take to each session so that they are easy to pack and transport to the pool. We have streamlined our set up routine to maximize time with the sub in the water. With the sub in the water, we run a "pre-flight" check on the sub to ensure all thrusters, sensors, and actuators are fully functional.

After the pre-flight check, the team begins data collection and issuing commands to the sub. There is always at least one person in the pool with the sub ready to pull the kill wand (mission termination system) if necessary and to reorient the sub according to testing needs. The team has a collaborative culture during testing sessions where issues are ironed out in group discussion. For specific task testing, members create an environment that will be close to competition conditions.

After the testing session, a summary is taken of what tests were conducted, what added information we obtained, and what improvements need to be made before the next testing session. This adaptive approach allows us to fine tune the efforts to the specific issues the vehicle is facing.

Out-of-water simulation is done in Gazebo, a physics engine that allows the team to simulate SubjuGator 8's behavior in water. This testing is done between in-water testing sessions to tune the sub and prepare for the next session.

Sensors are tested by comparing their expected outputs with the ideal outputs determined through Gazebo. While SubjuGator 8 is in the water, we read the navigation data that is being collected by the IMU, DVL, pressure sensor and cameras. We can view this data both before and after the computer has processed them to detect where bugs might be occurring.

To improve the static/dynamic stability of the sub, a station-keeping routine (vehicle attempts to hold a constant position) can be started, and then external forces or motion commands can be applied. The vehicle's response is then visualized both in the water and in simulation through Gazebo. Using the results of these disturbances, a dynamic reconfiguring program can be used to update gains to the PID controller (even while the vehicle is still running).

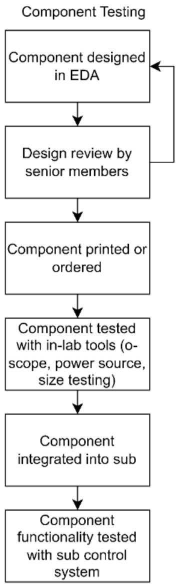


Fig 9: Component testing procedure.

Component testing (Figure 9) is done in multiple stages. Custom made PCB components go through a design review and changes are made before being ordered and printed. Once they are assembled, they are tested using an oscilloscope, power supply and electronic loads. If the PCB involves firmware testing, the code is written and tested. When complete, the component is integrated into the sub.

IV. ACKNOWLEDGEMENTS

The members of the Machine Intelligence Laboratory at the University of Florida are grateful to her many supporters. The culture at MIL builds a community of engineers encouraged to collaborate and teach.

The following corporate and MIL alumni sponsors were kind enough to assist with monetary and/or product donations along with technical support:

A. *Diamond Sponsor*

- L3Harris Corporation
- Erik de la Iglesia, MIL Alumnus
- Sylphase

B. *Gold Sponsors*

- Texas Instruments
- JD-Squared

C. *Silver Sponsors*

- SolidWorks
- Kevin Allen, MIL Alumnus
- Ralph Leyva, MIL Alumnus

D. *Bronze Sponsors*

- Jake Panikulam, MIL Alumnus
- Edward Kallal, MIL Alumnus
- Qualcomm
- DigiKey

V. REFERENCES

[1] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373), 2000, pp. 153-158, Doi: 10.1109/ASSPCC.2000.882463.

[2] 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.

[3] 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.

[4] 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.

[5] Collaborative labeling tool for machine learning. Commercial software (with free academic version). https://labelbox.com/.

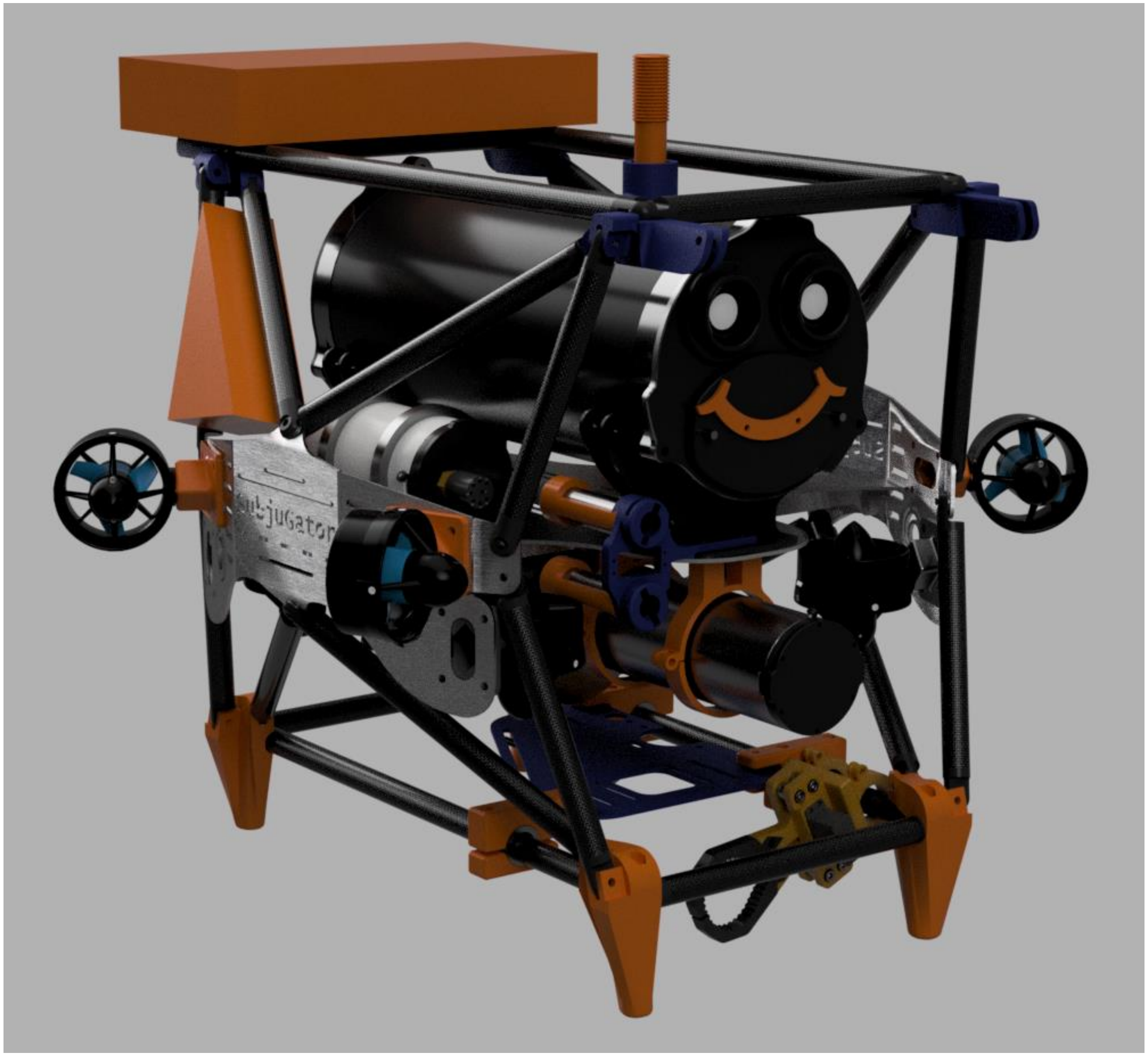
[6] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova. "Depth Prediction Without the Sensors: Leveraging Structure for Unsupervised Learning from Monocular Videos", *Arxiv.org*, 2018. [Online]. Available: https://arxiv.org/abs/1811.06152.

[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, 199

VI. Appendix A: Component List

Component	**Vendor**	**Model/Type**	**Specs**	**Origin**	**Cost (total)**	**Purchase Year**
Frame	JDSquared	Aluminum	Slotted structural hull	Custom	Donated	2014
	DragonPlate	Carbon fiber	Space frame	Custom	$800	2014
Waterproof Housing	JDSquared	Aluminum	Main hull, 100m depth	Custom	Donated	2014
	JDSquared	Aluminum	Battery hull, 100m depth	Custom	Donated	2014
	Blue Robotics	Acrylic 3”	Servo control hull, 200m depth	Purchased	$200	2022
	Blue Robotics	Aluminum 2”	Downward camera hull, 200m depth	Purchased	$80	2014
	Blue Robotics	Acrylic 3”	Navigation hull, 200m depth	Purchased	$250	2023
Waterproof Connectors	MacArtney	SubConn	External wet-mate connectors	Purchased	~$7000	2014-2024
	TE Connectivity	SEACON	External wet-mate connectors	Purchased	~$1000	2014
Thrusters	Blue Robotics	T200	Full Throttle FWD/REV Thrust @ Maximum (20 V): 14.8 / 11.1 lb f	Purchased	$1352	2021
Motor Control	Blue Robotics	Basic ESC	7-26v, 30amp, PWM	Purchased	$200	2021
3x Waterproof servos	Hitec	DB961WP	4.0V-8.4V, 0.15sec/60°	Purchased	$537	2023
8x Battery (2 used per run)	MaxAmps	LiPo	LiPo 5450 6S, 22.2v	Purchased	$2000	2023
Converter	Students	Power over Ethernet Hub	24V PoE	Custom	-	2014
CPU	ASRock	ASRock Z390M-ITX	mini-ITX motherboard, water-cooled	Purchased	$140	2019
	Intel	i9-9900k		Purchased	$500	2019
GPGPU	Nvidia	RTX 2080	Water-cooled	Purchased	$700	2019
Internal Comm Interface	Various	Various	USB	Purchased	~$500	2014-2024
External Comm Interface	Blue Robotics	Fathom Tether/Spool	Ethernet	Purchased	$1300	2024
DVL	Waterlinked	A50	4 beam, 5 kHz	Purchased	$7250	2024
IMU/magnetometer	VectorNav	VN-100	9-axis	Custom	$1200	2023
Pressure sensor	Blue Robotics	Bar30	300m depth, 0.2mbar resolution	Purchased	$85	2022
Algorithms	Adaptive PID controller					
Vision	OpenCV (Canny Edge Detection, Thresholding, Optical Flow), RCNN (YOLOv7)					
Acoustics	Scipy, Numpy (Time of Arrival, Least Squares, Fast Fourier Transform)					
Localization + Mapping	Unscented Kalman Filter on Manifolds implemented with Eigen					
Autonomy	Robot Operating System (ROS) Noetic					
Open-Source Software	All software is open source (OpenCV, Scipy, ROS, Numpy, PySerial, PyYAML)					

VII. APPENDIX B: SUBJUGATOR 8 RENDER



VIII. APPENDIX C: ELECTRICAL SYSTEM DOCUMENTATION