LEVIATHAN: The Design and Implementation of Duke Robotics Club's 2017 AUVSI Competition Entry

Andrew Buie, Nathaniel Brooke, Mark Chen, Neil Dhar, Kelsey Evezich, Estelle He, Sameer Khan
Austin McKee, Morton Mo, Vinit Ranjan, Logan Rooper, Samadwara Reddy, Raiyan Sobhan
Ramil Shaymardanov, Will Stewart

In 2017 the Duke Robotics Club returns for their second year in recent time to the AUVSI RoboSub competition with a faster, smarter, and leaner reconstruction of their faithful robot, Leviathan. Like last year, Leviathan is the brainchild of dozens of Duke undergraduate engineers and computer scientists working tirelessly throughout the year through the completely student-run Duke Robotics Club. Leviathan's outstanding waterproof mechanical design is the result of extensive research, simulation, modeling, and in-house CNC machining. In dozens of hours of testing no capsule has ever leaked, and the vehicle has proven highly maneuverable in every direction. Electronically Leviathan combines four brushed motor drivers, three hydrophones, two cameras, two inertial measurement units, a doppler velocity log, an altimeter, an on-board computer, and a unified power system to create a fully featured platform for informing and running the code base. The Python software stack retrieves the sensor data and fuses it to obtain a probabilistic estimation of state. It uses this state estimation to power controls, optimally derived from a model of the robot. Finally to navigate, a convulational neural network recognizes and tracks surrounding obstacles at 60 fps, allowing motion planning to plot the best course through the water. Leviathan was the result of incredibly hard work by students at Duke University, but equally as important, the project owes its continued success to the club's long-time mentors and sponsors, The Lord Foundation and the Duke Student Government Student Organization Funding Committee.

I. INTRODUCTION

LEVIATHAN returns this year to the 2017 AUVSI RoboSub competition with the Duke Robotics Club for their second time competing in a competition since 2008. Work on the project, split between three different subsystem teams (subteams), began in Spring 2014. The mechanical subteam was responsible for the electronics enclosures, actuators, frame design. The electronics subteam was responsible for the power architecture, the sensing systems, the onboard computation hardware, and the firmware for the microprocessors. The software subteam wrote software to control the robot, handling everything from sensor fusion to motion planning to computer vision. Lastly, the testing team was responsible for ensuring the quality of the other subteams' contributions as well as running integration tests in the pool.

II. HARDWARE DESIGN OVERVIEW AND STRATEGIES

A. High-Level Approach

Leviathan was originally designed in the context of a brand new team approaching the AUVSI competition with no previous experience. Because of that, the design process began with a detailed study of the competition rules, previous score results, and previous design entries instead of a traditional iteration on a previous entry. Research revealed that many teams struggle to implement even basic functionality, and many others are unable to use sophisticated features because of reliability problems. This research led the team to prioritize navigation-based tasks and to build a highly maneuverable AUV. The goal was to create a design that is sufficiently modular that new instrumentation and robotic manipulators could be added later without difficulty, and a design that would be robust and reliable enough to function as designed at the competition, something that many teams struggle with. Competition rules and requirements were weighted against soft factors to develop the following additional design constraints:

- Make maintenance and troubleshooting as easy as possible
- Focus on navigation-based tasks, but allow for modular upgrades
- Give up size/weight optimizations for potential functionality
- Give up size/weight optimizations for greater independence of the subteams
- Favor reliability and robustness over complexity and additional functionality

However, last year's competition gave the team invaluable insight to the tradeoffs these design choices effect, and with this knowledge the team aimed to take the existing design and bend it towards an updated set of goals. These are the mistakes that were made, and this is what was learned:

- The robot was too buoyant, requiring in return too much weight to keep it underwater
- The robot uses much less power than expected, rendering a double set of batteries unnecessary
- The plan to minimize main capsule openings by having separate battery capsules was not realized because the main capsule usually had to be openend before each test for troubleshooting anyways
- The minimally-structured main capsule interior, while easy to initially wire, proved challenging to

troubleshoot and a major reliability concern
- Thermal design must be considered at a module-by-module level, rather than total capsule heat

Each of these insights was leveraged to improve this year's design.

B. *Vehicle Design*

1) *Hull*

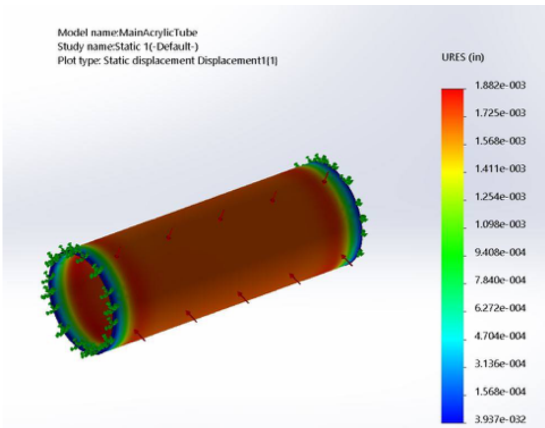


Fig. 1: Simulation of the Capsule

One of the mechanical subteam's top priorities was designing a capsule system that can be opened quickly, requires little maintenance, and never fails. The vehicle hull system consists of a single main capsule and two camera capsules.

Last year's competition demonstrated that the main capsule was even more reliable than we had originally projected. The reliability of the main capsule, combined with weight and buoyancy considerations, led to the removal of the two separate battery capsules from the design.

As part of optimizing for weight, the weight of each mechanical subassembly was found in SolidWorks. While the team intuitively thought that the best way to cut back on weight would be to reduce the aluminum frame size, it was found that there were only a few pounds of excess aluminum on bot (which to even save, would require a complete re-machining), compared to almost 10 pounds of weight in battery capsules. The team also considered shaving down the sealing flange and plug on the main capsule. This was also ruled out because while the large aluminum flange looks quite heavy, only a pound would have been saved by lathing it down, which did not justify the risk of damage to the main capsule or intoducing strain not seen in the simulations which could impact the watertightness of the design.

The sealing flange bore-seal design combines the robustness and simplicity of bore seals with features that make removing the end cap much easier. The acrylic-mating face of the sealing flange is machined to match the exact piece of acrylic tubing with which it seals. Mating the removable endcap directly with the acrylic was intentionally avoided; the tolerances of acrylic tube are imprecise enough that even one of the correct nominal size can be outside the recommended parameters of the chosen o-ring. This metal flange creates a surface that can be used with a jack screw to remove the endcap with no risk of cracking. Although the metal flange adds size, weight, and construction complexity, it provides a durable interface to the endcap and ensures the o-rings have a perfect sealing surface.

The removable endcap was originally designed with two o-ring glands so that either a single o-ring could be used for removability, or an additional one could be added for higher reliability. After test, there was no noticeable difference in difficulty of removing the endcap, so both o-rings were ultimately used. The removable endcap also serves as the interface for the vehicle's SubConn and SEACON waterproof connectors. On fixed side of the capsules, valves ensure that the removable endcaps are immobilized by vacuum during removal.

Fig. 2: Capsule exploded view

Last year's permanently sealed camera capsules were replaced with resealable capsules because of upgrades to the cameras. This year's cameras were considerably more expensive, and the risk of damaging them from trying to open a permanently sealed capsule outweighs the risk of leaks.

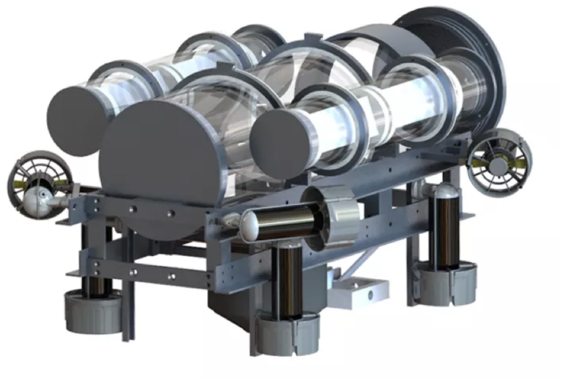
2) *Frame*

The frame consists of two aluminum "cross sections" that hold the main and battery capsules in place with polypropylene bushings to prevent the acrylic tubes from scratching. These cross sections attach to an aluminum box frame designed with extra mounting area for components to be added or moved during integration and testing. The frame was designed so that the center of mass is directly beneath the center of buoyancy, making Leviathan self-righting. In this way, good mechanical design simplified the work of the computer science and electronics subteams, an overall design success.

3) *Battery Pods*

C. *Bridge*

The bridge is the structure on which all main capsule electronics are mounted. The bridge is designed hold all of the electronics and wires neatly, while also meeting the thermal requirements of individual modules. Between the



two rails of the bridge, cards are mounted, each holding a subsystem, and each customized for that subsystem. For example, the motor driver cards have integrated fans, and supports to allow airflow between the motor drivers.

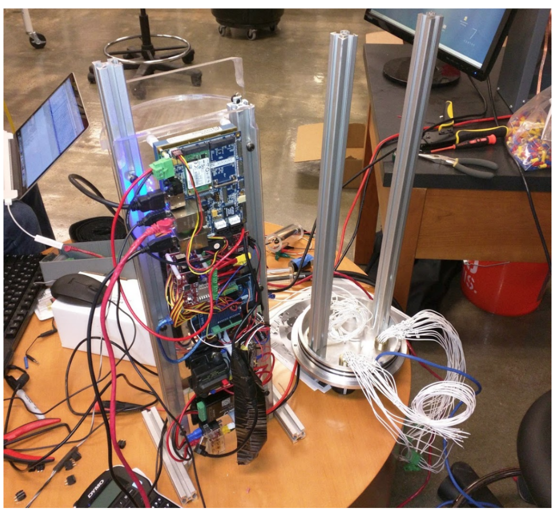


Fig. 3: The Bridge

Mounted on the bridge is an Acromag 6400 single board mil-spec computer, a ConnecTech carrier board, 3 Atmega microcontrollers, four dual-motor brushed motor drivers (not pictured), two IMUs, an acoustics filtering board (not pictured), a USB hub, a switchable thermal breaker, adjustable DC-DC power converters, and a fuse box. The fuse box and thermal breaker provide adequate overcurrent protection in the event of an electrical failure. The digital components interact over the USB hub network, and all external connections are made through detachable plastic connectors.

D. Actuators

In line with the goal of focusing on navigation based tasks, only thrusters for vehicle motion and marker droppers were implemented. SeaBotix BTD150 thrusters were chosen because of their easily sealed electrical interface, their simple mechanical mounts, and their discount through Seabotix's generous sponsorship. Four vertical thrusters are mounted on each of the four corners, giving the vehicle freedom to move up and down the Z axis (heave) and around the pitch and roll axes. Four horizontal thrusters attached on the bow and stern at 30 degrees allow the craft to sway, surge, and yaw. Testing showed that this angled configuration offered greater sway control at the cost of less efficient surge, a reasonable tradeoff when considering the relatively short length and duration of the competition course. Two magnetic marker droppers have been implemented as well. Drivers inside the main capsule, when signaled, drive a solenoid, retracting a magnet. When the magnet retracts, the force applied to the steel marker decreases, and the marker falls.

E. Sensors

Leviathan uses a combination of motion, vision, and acoustic sensors to understand its own state and the world around it. Localization and mapping of the AUV's environment is one of the hardest challenges of underwater robotics and was the driving factor behind many of our design decisions. The sub uses a single [TODO: insert camera name here], a Teledyne Doppler Velocity Logger, an Omega PX309 pressure transducer, an array of three Aquarian Audio Products H1c hydrophones, a VectorNav VN-100 IMU, and a thermocouple to monitor the internal capsule temperature of the robot. This raw sensor data is processed and sent to the computer where it is combined and used to determine the output of the sub's actuators.

F. Model Dynamics

The robot is modeled as a rigid Newtonian body subject to gravity, a buoyant force, non-linear drag forces, and its own thruster forces acting on the robot's mass and own inertial tensor matrix. Attitude is represented as a unit quaternion, and the entire state is represented in the local North-East-Down (NED) frame. This model of the robot is continously linearlized around the current state and fed into both the Kalman Filter and the Linear Quadratic Estimator controller. Quaternions were chosen in this context for their lack of singularities and for their computational speed, an important factor when simulating at high frequencies. The condensed state vector is shown below.

$$\mathbf{q}_4 := \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} \cos(\frac{\beta}{2}) \\ e_x \sin(\frac{\beta}{2}) \\ e_y \sin(\frac{\beta}{2}) \\ e_z \sin(\frac{\beta}{2}) \end{bmatrix}$$

$$\mathbf{x}_k := \begin{bmatrix} \mathbf{r}_3 \\ \hat{\mathbf{q}}_4 \\ \mathbf{v}_3 \\ \omega_3 \end{bmatrix}_k \quad \mathbf{u}_k = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_8 \end{bmatrix}_k$$

Perhaps unsurprisingly, the most difficult part of using such a model is the number of physical constants involved, many of which are difficult to find experimentally. To this end, the computer science team devised an experiment to estimate the most difficult constants: the inertial tensor matrix values and twelve independent drag constants. The robot was driven around underwater for minutes at a time, logging sensor data. Assuming the sensor data is zero-mean and only corrupted by gaussian white noise, the data was naively fused together without filtering into a sequence of thousands of state vectors. After defining an error function, an off-the-shelf minimization technique can then be run massively in parallel in the cloud to find the optimal set of hard-to-find physical constants that lets our model most closely match the recorded data.

$$err(\mathbf{c}) := \sum_k |f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{c}, dt) - \mathbf{x}_{k+1}|^2$$

The results are noisy but show a strong tendency towards what in all cases are physically reasonable results which could then be qualitatively tested in the simulator. Most importantly, the values found through this novel approach led to great performance by the robot in both sensor fusing and controls.

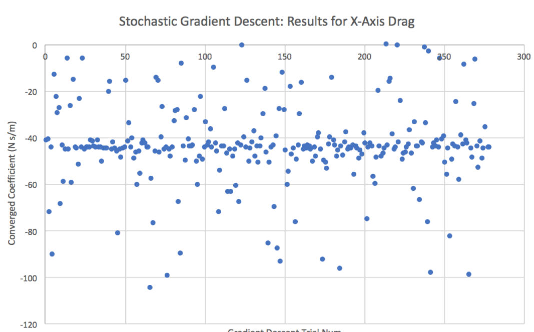


Fig. 4: Stochastic gradient descent results for x-axis drag

G. Controls

This year, controls were rewritten to use a Linear Quadratic Regulator built from a linearized version of the robot model. Because of the complexity of the model and the fragility in manually coding in the large gradient functions, it was decided to as much as possible represent the math symbolically. However, wanting to keep the code base purely Pythonic, the team created a flexible math and physics library that can operate on both floating point numbers (NumPy) and symbolic expressions (SymPy) agnostically. All of the robot's controls are then represented symbolically, allowing for higher-order operations like derivatives to allow linearization. At run-time, these large functions are then compiled by Theano into tensor graphs to run in C, boosting execution time by an order of magnitude and enabling previously impossible uses of the model, such as the gradient descent described above.

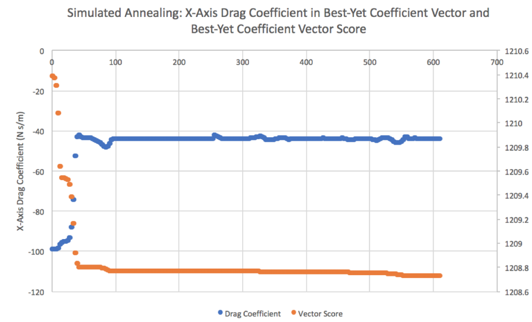


Fig. 5: Simulated annealing with the same data

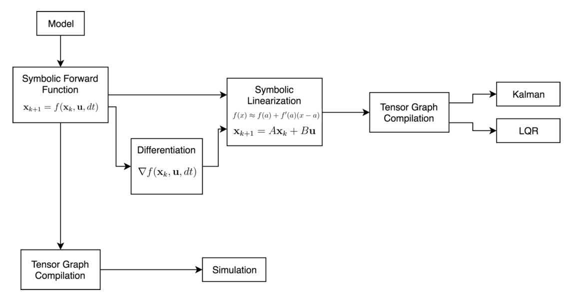


Fig. 6: Simulated annealing with the same data

H. Software Infrastructure

Both to avoid the steep learning curve for new members and to not be restricted to a certain platform, the software team decided to again forgo ROS in favor of a homemade solution. Leviathan uses a completely Pythonic infrastructure of microservices which pass messages to each other through the Redis Publish/Subscribe paradigm. A Python hypervisor then starts and monitors the services, and a websocket-enabled Django server offers live GUI feedback and control of all data streams including video. A series of scripts and microservices then provide the ability to simulate controls, sensor fusion, and even computer vision in real time, streaming the output to the web GUI where Leviathan and its environment are rendered in 3D using Three.JS.

1) Acoustics

A passive hydrophone array is used to triangulate the location of the Benthos ALP-365 acoustic pinger.

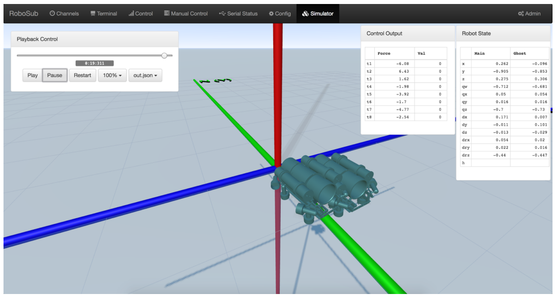


Fig. 7: Live streaming the robot state to the web interface

The task's foremost obstacle is reaching a minimum analog-to-digital sampling rate on a microcontroller. In order to sample the 40 kHz signal, an open source handwritten assembly routine for the Arduino Due was used. The signals from the three Aquarian H1C hydrophones individually pass through pre-amplifiers, a 8th order butterworth low pass filter board, and then to the microcontroller where they are band-pass filtered to isolate the pinger's signal. From the cross-correlation-peaking/Time-of-Arrival/difference-of-phase of the three signals, the location of the pinger can be calculated. Currently, the array fails to pick up every ping– preliminary investigations point to the microcontrollers poor sampling rate so an exploration of other microcontrollers with faster ADCs is necessary. An ultimate design would require an array that accurately triangulates the pinger within a reasonable passive listening time, aiding the guidance-decisions made by the main computer.

2) Computer Vision

The computer vision module is designed to provide estimates of objective positions relative to the robot. Two complementary methods are used in tandem. The first, as traditional computer vision, cleans all images using contrast stretching and then thresholds in the RGB and HSV color planes to identify objectives. The Hough Transform then detects lines and circles via a voting algorithm, and separately contour analysis finds all contours in the image and evaluates them as possible matches with the objective. Once the objectives are marked on the image, determining position reduces to a geometry problem with a few unknown constants that are solved for with calibration images.

The second is through a convolutional neural network (CNN). After hand-segmenting a dataset of thousands of images taken last year at the competition, a 25-layer CNN designed by Google for mobile device object classification (SSD MobileNet) was reconfigured for RoboSub. Pre-configured weights trained by Google on the COCO image database were used as a starting point, with only the last few layers needing retraining for obstacle detection. The network is able to classify and give a bounding box to all obstacles in a video stream at 60 FPS while still maintaining 90% accuracy.

Ultimately, a combination of both approaches is used by Leviathan. In most cases, the machine learning is used to find a bounding box, and then the traditional computer vision techniques are used within. However, with different obstacles only one or the other is used.

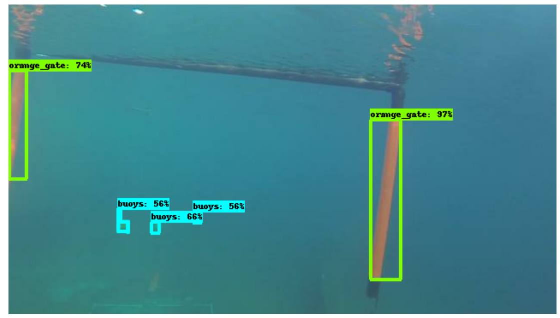


Fig. 8: Obstacle detection with SSD MobileNet

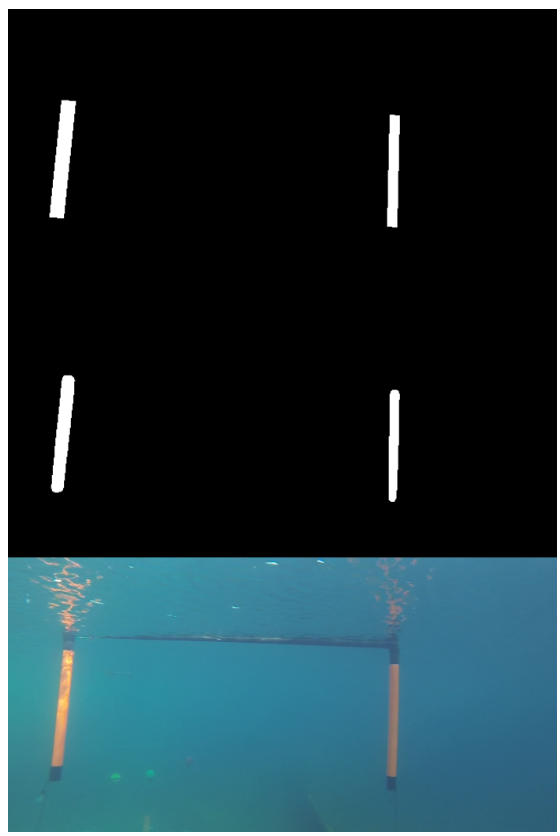


Fig. 9: Cleaning and rectangle extraction using traditional CV methods

III. Testing Process

Vigorous testing of the mechanical systems has been considered core to the entry's success and was utilized throughout the construction process. The initial designs were guided by bounded Solidworks simulations, placing theoretical bounds on the hull thicknesses needed to maintain integrity. Mechanical components were then stress tested early and often, with both the hulls and battery pods having completed multiple ten-hour trials at over 15 feet in depth. The electrical components were also tested often, but full-system tests were intentionally kept to a minimum because of the disastrous damage a single leak could cause the electronics.

The testing subteam developed specific tests for each subsystem that could be run in-lab. The team also recognized the rarity of full-system tests; together with the software team a program to record and play back all system parameters and values was created so that each test could be re-simulated and each run scrutinized. Data from these runs could be viewed in real-time or could be downloaded to a computer for later analysis.

IV. Experimental Results

At the time of this writing, Leviathan is a bare-bones vehicle capable of full freedom of motion at low speeds (roughly 2 knots translational speed). The vehicle has two functional cameras for identifying obstacles below and in front of it, three positioning sensors including a DVL for navigation, and a single pressure hull. The robot is capable of sophisticated motion planning and obstacle interpretation through vision, but currently has no external actuators for non-navigational tasks. It has enough battery capacity to run for around 20 minutes with medium duty-cycle thruster usage (30-50%). Lastly, by the time of the competition, it is hoped that the vehicle will have also gained acoustic localization functionality for the final task.

Acknowledgment

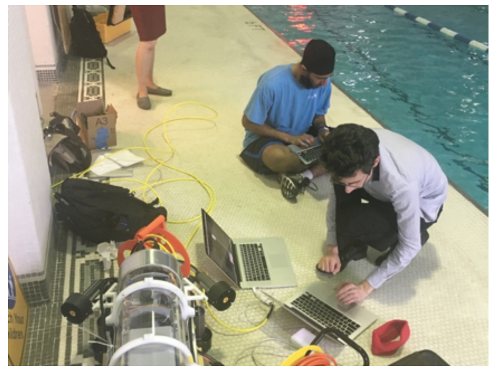
Duke Robotics is a club within the Pratt School of Engineering. We are supported by the faculty here at Duke, and gratefully acknowledge the faculty and administrative staff here who have helped and continue to help make our club a success. Special thanks to Professors Michael M. Zavlanos, Kris Hauser, and George Konidaris, and our Undergraduate Program Coordinator, Lauren Stulgis.

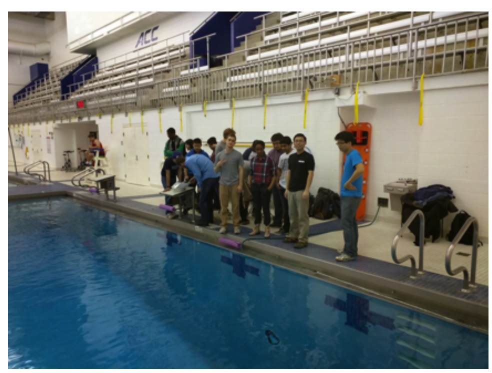
Appendix

A. Outreach

Duke Robotics Club recognizes the importance of working with their local community and inspiring future generations of STEM students. They have worked with both local middle schoolers and high schoolers, helping Durham Academy Middle School coach a pilot FLL team and robotics afterschool program and mentoring Team 900 Zebracorns navigate the FIRST competition. They have also advised countless teachers' curriculums through a partnership with Project Lead the Way. Even though each member's time could have been spent bettering Leviathan the team still acknowledges how much more science and robotics can advance with each class of students. Duke Robotics Club wants to encourage as much innovation as possible, and they are proud to be able to spark ideas in generations of students to come.

B. Team Photos





References

[1] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murhpy, *Speed/accuracy trade-offs for modern convolutional object detectors*, CVPR 2017.

[2] R. Tedrake, *Underactuated Robotics: Algorithms for Walking, Running, Swimming, Flying, and Manipulation*, MIT 2016.