Illinois Autonomous Underwater Vehicle Design and Implementation of Raubvogel

Adrian Brandemuehl, Shubhankar Agarwal, Quanjie Yang, Vibhakar Vemulapati, Marty Puru, Jason Han, Marc Backas, Tanay Vardhan, Howard Shan, Saksham Gupta, Ananmay Jain, Ricky Cheng, Howard Shan, Justin Tu, George Chen, Scotty Liu and David Forsyth.

University of Illinois at Urbana Champaign

Abstract - IllinoisAUV is a relatively new team driven by undergraduate students drawn from diverse backgrounds and disciplines, who aim to explore the applications of cutting-edge technology and engineering by building an autonomous underwater vehicle. AUVs have immense applications in industry, government, and science. This year, IllinoisAUV has designed, built and programmed Raubvogel, with the goal of alleviating the problems faced last year with Enigma. As with last year, an important goal of the team is to build a low cost sub that makes up for the lack of expense in intelligent software design and algorithms.

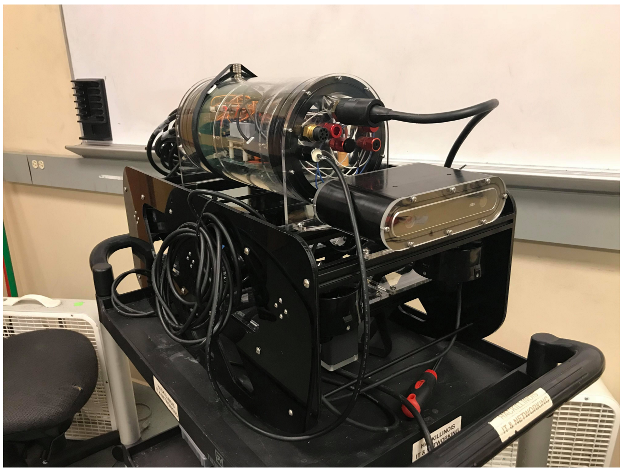


Figure 1: Raubvogel

1 Competition Strategy

IllinoisAUV is a software focused team that works on enabling interesting algorithms by building a platform that makes software development and iteration easy. Given that the team is focused on incremental improvements, we have built a submarine with fixes to last year's problems, while preparing the team for performing more complex tasks next year.

The lack of manipulation tools on Raubvogel limits our scoring strategy this year. We plan to complete all of the motion and vision tasks, including the gate, path, craps and buying gold chips. Completing anything more than the gate would be a significant improvement over last year, where our sub was only capable of making basic motions without vision control.

This year, we will also be working towards sensing the pingers, as we have the hardware to do so. The software is still under development and needs to go through rigorous testing. Pinger detection and localization is considered a stretch goal for us this year.

While RoboSub is a competition among many teams, our goal is to improve as much as we can on our own score compared to last year. With most of the mechanical and electrical problems that we faced last year fixed, we have improved our sensing, motion and detection algorithms for a more successful second year at RoboSub. This time around, we built for the ability to simply tune our vision algorithms, without the need to rewrite significant portions of the software just to have a moving submarine like last year. We have confidence that tuning and retraining of the vision algorithms is sufficient for a successful year at the competition.

The team has focused on building a solid base for the years to come that can be expanded and iterated upon without high costs or effort. With the laser cut design, manufacturing is extremely affordable and the team can reuse all of the other components. A platform for incremental improvements allows the team to focus on engineering towards the tasks at each competition in the future.

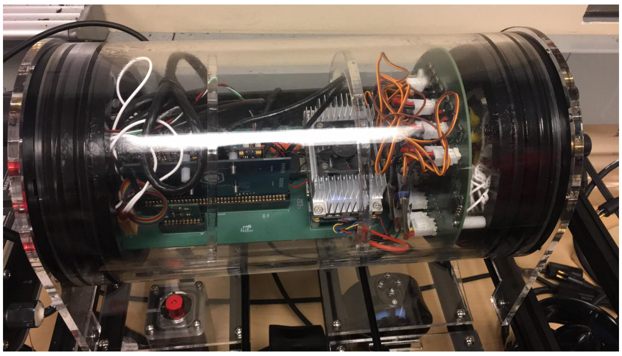


Figure 2: Raubvogel's electronics

2 Design Creativity

Illinois AUV is focused on developing smart software systems in order to gain competitive advantage over teams with significantly better mechanical and electrical systems.

2.1 Cheap, Iterable Mechanical Structure

IllinoisAUV has limited resources and manpower for designing and implementing complex mechanical systems. Our school does not provide students with easy access to CNC machining tools, and few metal machining tools are available. With these limitations, our team focused on building a submarine that was feasible to construct with only a laser cutter. Many teams use a CNC mill to machine a custom bulkhead for two tubes, giving them more room for electronics and connectors. Our electronics are small in comparison, requiring only one single tube to house for all of our electronics. Thus, we did not need to follow the design that many teams use. All structural parts of Raubvogel can be laser cut in less than an hour, making rapid iteration on the mechanical design simple and quick.

The frame and end caps are custom, and all other frame elements are off-the-shelf parts from BlueRobotics.

2.2 Compact Electronics

Many teams leverage full desktop computers, complete with motherboards and discrete graphics cards for controlling their subs. These constitute common vehicle space and power requirements, and require heavy extra batteries plus external microcontrollers for interfacing with actuators and sensors. Our team has avoided many of these problems by using the NVidia Jetson TX2, a credit card-sized computer that has many hardware pins for connecting to various sensors without any modifications or extra peripherals. Our electronics fit completely in a 6"x12" tube, and we do not have to open the tube for any normal operation. The electronics are visible in the tube in Figure 2.

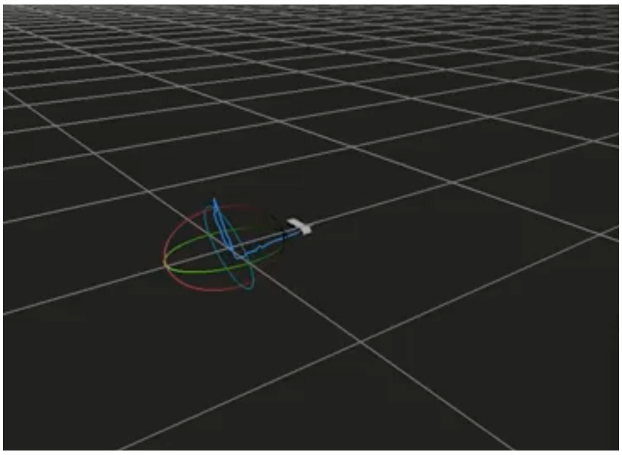


Figure 3: Position and Orientation Tracking with ZED camera

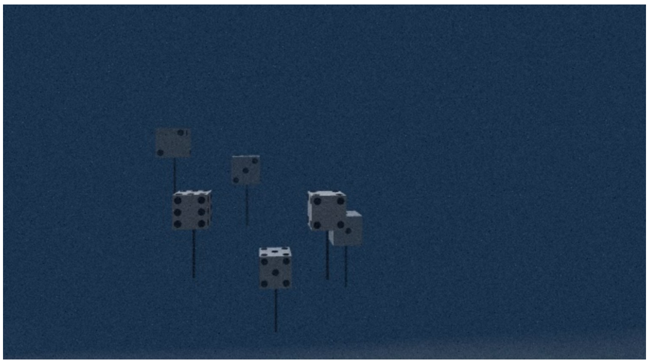
2.3 Simulator

Developing a physics-based simulator was one of the top goals for the team. We wanted to create a system where we can test our robot software stack in simulation to accelerate development and testing, even before a physical robot was complete. We decided to use open source UUV simulator [2] as the base platform for the simulator. UUV simulator is based on gazebo (physics based robotic simulator framework), and fits in perfectly with our ROS based software stack. Having a simulator made our water testing sessions much more productive. Our plan was to develop the simulator to the point where we can only change the object detection models for competition.

2.4 Deep Networks with Tracking

Similar to a few other teams, we are using deep networks for object detection. In our first year, we found that hand tuned classical computer vision algorithms were too sensitive to changes in light parameters. We are using the YOLO v3 [3] architecture for detection, running on the Jetson TX2. To conserve computing power for other processes, YOLO is limited to run at only 4-5 fps, as the GPU is needed for other tasks, such as visual odometry. In order to maintain target positions in between YOLO detection runs, we run a conventional object tracker initialized from the YOLO run. We obtained sufficiently positive results, and saw a significantly decrease in the computation required during our testing.

(a) Qualification gate



(b) Dice

Figure 4: Robosub tasks simulation

2.5 Positioning using Hydrophones

We have been working on an algorithm to estimate our position relative to the pingers in the pool. The pingers act as 2 satellites for our system. Given the distance between the pingers, we can estimate our 3D position with relative accuracy. Time to reach estimations of our position is dependent on the number of hydrophones available on the robot: 2 hydrophones require more time to reach the solution, while 3 hydrophones yields near-instant results. This algorithm gives us accurate results in simulation, but still needs to be tested in the competition setting.

2.6 Sensor Fusion

Our team uses multiple sources of sensor data for determining our position in the pool. This year, we have added a ZED camera, a commercial stereo depth camera, to the suite. The ZED camera provides visual odometry, which we fuse with an external IMU and bottom camera visual odometry [1]. Using these data sources, we intend to get high quality positioning at a low cost compared to a DVL. We will also integrate the hydrophone positioning system to our sensor fusion suite, depending on the accuracy of the hydrophone positioning algorithm.

2.7 Deep Reinforcement Learning for Visual Servoing

In classical visual servoing the target object is detected in a camera frame its in-frame coordinates are computed. A simple control loop is then used to center the target object on the screen. If the sub continues to move forward, it will eventually hit the target.

This approach requires accurate detection of the object and proper control signals, which can be difficult for complicated systems. We decided to experiment solving this problem using deep reinforcement learning to make a controller to hit objects in a simulated environment. Reinforcement learning solves the difficult problem of correlating immediate actions with the delayed returns they produce. In this approach we have an agent (our robot) interacting with an environment (dice in the pool) and our agent gets positive and negative reward over time based on the set of actions it takes. We maximize the expected reward over time, that is, training the agent to take actions which will give the maximum score over time. The system consists of a neural network to detect current state in the environment and another neural network to make decisions based on the estimated state. We do not have to explicitly detect the object, and the agent learns to take the correct actions overtime by replaying the scenario in simulation.

In simulation, the agent has converged to a decent controller. We were able to hit the dice 30% of the instances after 2 days of training in a constrained action state (only controlling the yaw). We plan to test this approach in Robosub 2018 pool to see if the results generalize to the real world.

3 Experimental Results

Up to this point, our experiments have taken place in simulations. Many of the manufacturers we worked with for parts had long lead times, and orders were placed late due to funding issues. Now that we have the sub physically completed, we are moving to testing in the water while continuing the tests we ran in simulation. In future years, we will continue to use the sub built this year with some modifications, so we can begin to test the sub early in the year.

We started testing basic software stack in early January using the robot, Enigma, from last year. Due to water leaks and general poor performance, we were unable to continue testing for long periods of time. Raubvogel was put in the pool for the first time in mid June.

As our mechanical and electrical team developed Raubvogel, our simulator proved extremely helpful

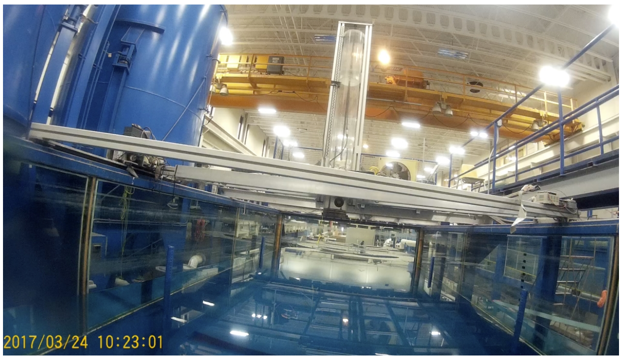


Figure 5: Hydrosystems Lab Testing Facility

in testing our software stack. We created the whole course of 2018 tasks in our simulator, improving our ability to refine our software decision making, if not our object detection models. Having a physics-based simulator meant that we were able to simulate most of the sensors on our submarine, making simulation testing much closer to actual water testing.

4 Acknowledgement

Illinois AUV would like to specially thank our faculty advisors and mentors: professors David Forsyth, Volodmyr Kindratenko and Alexander Schwing, who has given us direction and financial support through our many hours of development and testing. Additional thanks goes out to Van Te Chow Hydrosystems Lab at University of Illinois for providing pool for our testing facilities (5). Finally, we would like to thank our sponsors: Uber ATG, Microsoft, Nvidia, UIUC SORF, ACM@UIUC, Viasat, Oceancomm, Sparton, TE Connectivity and Advanced Circuits.

References

[1] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *IEEE International Conference on Robotics and Automation*. IEEE, 2014.

[2] Musa Morena Marcusso Manhães, Sebastian A. Scherer, Martin Voss, Luiz Ricardo Douat, and Thomas Rauschenbach. UUV simulator: A gazebo-based package for underwater intervention and multi-robot simulation. In *OCEANS 2016 MTS/IEEE Monterey*. IEEE, sep 2016.

[3] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.