2018 ASEE Annual Conference & Exposition
June 24–27, 2018
Salt Palace Convention Center
Salt Lake City, UT
Paper ID #21232

Beyond the Sea Perch

Dr. Thomas R. Consi, Massachusetts Institute of Technology

Dr. Consi is the education director at the MIT Sea Grant program. His research interests include underwater robotics, biomimetic robotics and marine animal biomechanics. Dr. Consi is passionate about engineering education and has developed and taught several hands-on lab-oriented courses primarily in mechatronics and marine robotics.

Miss Jocelyn Frances Lorrey, Massachusetts Institute of Technology

Mechanical engineering undergraduate at MIT, class of 2018.

Ms. Michelle Kornberg, Massachusetts Institute of Technology

Michelle Kornberg is an MIT sophomore majoring in Mechanical and Ocean Engineering.

Beyond the Sea Perch

Abstract

The Sea Perch ROV has long been used to introduce students to marine robotics and to ocean engineering in general. The vehicle is manually operated with three switches controlling its three thrusters. We have enhanced the Sea Perch in two ways, both designed to increase the educational utility of the vehicle. First, we developed a flight recorder that sits in the Sea Perch payload bay and records 3-axis orientation, acceleration and rotation. Second, we developed an onboard computer for the Sea Perch that enables students to greatly expand the capabilities of the vehicle through the incorporation of new sensors and actuators. We present the design of the flight recorder and our first experience using it in an educational program. We also present the design of the onboard computer system.

Introduction

Ever since the publication of their little yellow book "Build Your Own Underwater Robot and Other Wet Projects" by Harry Bohm and Vicki Jensen [Westcoast Words, Vancouver, BC] thousands of students have built and flown Sea Perch remotely operated vehicles (ROVs) (www.seaperch.org). Made of PVC pipe, toy motors, switches, and other low-cost easily obtained parts, the Sea Perch has provided students with hands-on experience in marine robotics and introduced them to basic concepts in ocean engineering and to STEM concepts in general [1], [2]. The simplicity of the vehicle's design makes it tractable to beginning students (from middle school on up) with little or no building experience, yet once students see it maneuver in a pool they are immediately inspired to add capabilities. We have developed two systems to enable students to add capabilities to the Sea Perch as well as to enhance its utility as an ocean engineering and STEM teaching platform.

As it is the Sea Perch is an excellent tool for teaching basic hydrostatics in a quantitative way. The displacement of the vehicle, its overall weight, and the distribution of weights can be measured and the centers of gravity and buoyancy can be calculated. Experience with hydrodynamics, the forces acting on the moving Sea Perch, in contrast, is only intuitive and primarily comes from experience driving the vehicle and watching its response. The main challenge here is to obtain quantitative information of the vehicle's motion. To solve this problem we have developed a "gray box" flight recorder that records the performance characteristics of the vehicle for later download and analysis. A single-chip inertial measurement unit was interfaced to a microcontroller-based data logger and the entire battery-powered system was mounted in a small waterproof box that was mounted in the payload bay of the Sea Perch.

The second enabler that we developed was a simple on-board computer system consisting of a microcontroller, four high current motor driver boards and a wired communications system to a surface or topside computer. This configured the Sea Perch in the way that most ROVs are designed – an onboard computer receives and executes commands from a topside computer that provides a user interface to the pilot/scientist. This system overcomes the major limitation to adding capabilities to a conventional Sea Perch; each added motor or sensor needs at least two wires to be added to the tether. This causes the tether to quickly grow in diameter and stiffness making the vehicle less controllable. In addition, the pilot will become overwhelmed with the number of switches he/she must rapidly operate in order to fly a vehicle with more than three thrusters. In our system a topside computer receives input from the pilot and sends those commands to the vehicle's onboard computer. The communication link uses a fixed number of conductors in the tether that does not change as new subsystems are added to the vehicle. Thus the tether diameter does not become a limiting factor in adding new devices to the Sea Perch. The onboard computer interprets the commands, controls the actuators, reads the sensors and sends data back to the pilot. The onboard computer can potentially handle local control tasks such as obstacle avoidance or maintaining a fixed altitude above the bottom greatly relieving the pilot's job of controlling all aspects of the vehicle in real time. Finally, with the addition of an onboard battery pack the tether can be cut and advanced students could experiment with a truly autonomous vehicle.

Sea Perch "Gray Box" Flight Recorder

A block diagram of the system is shown in Fig. 1 and the complete system is shown in Fig. 2. The core of the flight recorder is a Bosch BNO055 intelligent 9-axis absolute orientation sensor (Bosch Sensortec GmbH, Reutlingen/Kusterdingen, Germany). This MEMs sensor was purchased mounted on a small printed circuit board that includes a crystal, voltage regulator and other discrete components necessary for its operation (Adafruit BNO055, Adafruit Industries, NY, NY). The BNO055 contains a 3-axis accelerometer, 3-axis gyroscope and 3-axis magnetometer. An on-chip 32 bit microcontroller processes the raw sensor data with a sensor fusion algorithm and outputs the measurements in physical units. The chip interfaces to the data logger via an I^2C serial interface. The data logger consists of an Arduino UNO microcontroller (Arduino.cc) equipped with a microSD card add-on board (aka shield) (Model DEV-12761, Sparkfun Electronics, Niwot, CO). The shield contains a small prototyping space in which we mounted the Bosch sensor board, a pushbutton switch and two LEDs that constitute the simple user interface. Power was provided by a 9V alkaline battery. The entire system was mounted in a small waterproof box (Fly Caddy, Otterbox, Ft. Collins, Co) that was strapped into the Sea Perch payload bay with plastic cable ties (Fig. 2B).

The program to run the flight recorder was written in the Arduino programming language which is similar to C/C++. The software utilized the Adafruit Sensor.h general sensor library and the Adafruit BNO055.h library for the Bosch sensor. Upon power-up the microSD card and the Bosch sensor are initialized and the program halts if either initialization fails and blinks an LED, red for the microSD card and green for the Bosch sensor. In case of an error, the user must fix the problem and reset the Arduino in order to proceed. After successful initialization the red LED lights and the program waits for the user to press the pushbutton to start data acquisition. Upon a button press the green LED lights (red LED off) and data is acquired and stored on the microSD card at 10Hz. Pressing the button again halts the data acquisition process, closes the file on the microSD card, turns on the red LED and turns off the green LED, and the program waits for user input. The user could push the button again to invoke another data acquisition session with the data stored in a new file, or the user could pull the microSD card and insert it into a laptop computer for data transfer, plotting and analysis. The LED user interface is designed to let the user know when it is safe to pull the microSD card (red LED on) or when the card is being written to and should not be disturbed (green LED on). To repeat the procedure the user inserts the card, resets the Arduino and the program begins from the beginning (void setup() function).

The system records XYZ angular orientation, XYZ angular velocity, XYZ linear acceleration, and time in milliseconds from the start of the data acquisition period. The rate of data acquisition was chosen to be 10Hz but this can be easily changed by the students by modifying a constant in the source code and recompiling and uploading the modified program to the Arduino. The data is stored in an ASCII tab-delimited file that can be opened with MATLAB or EXCEL.

In actual operation the student opens the box, initializes the system, starts the data acquisition and then closes the box and attaches it to the Sea Perch (Fig. 2B). At the end of the experiment the student retrieves the box, opens it and removes the microSD card to transfer the data to a laptop computer. During Sea Perch operation there is no communication between the flight recorder and the user.

The system was incorporated into a week-long introduction to marine robotics for high school students. This program, called E2, was sponsored by the MIT Office of Undergraduate Outreach Programs with the goal of providing unique and inspiring hands-on STEM experiences to talented high school juniors from underrepresented and underserved communities (https://oeop.mit.edu/). Our group of 14 E2 students worked in pairs to build standard Sea Perch ROVs from kits (www.seaperch.org) and learned to fly them in the 3000 gallon tank in the MIT Sea Grant Teaching Lab. They were then introduced to design of the flight recorder: its components (Bosch sensor, Arduino data logger), and the theory of accelerometers, gyroscopes and magnetometers. This introduction is why we like to refer to our flight recorder as a “gray box” device in contrast to a “black box” found on aircraft. They were taught the design and operation of the flight recorder and how to use MATLAB to view and plot their data. The students incorporated the flight recorders onto their vehicles and ran them in extreme maneuvers such as: high acceleration, crashing into the wall of the tank, and throwing the vehicle into the water. One pair of students modified the thruster configuration of their Sea Perch to enable it to perform barrel rolls. While they were running their vehicles the students carefully recorded the time, from the start of data acquisition and when interesting events (such as hitting a wall) occurred. In this way they could look for the occurrence of the event in their data plot and see the resulting acceleration and rotation. At the end of the week the students presented their work to the entire E2 program. Figure 3 is an example data plot showing acceleration when a Sea Perch was crashed into the tank wall.

Sea Perch Onboard Computer

An Arduino Mega microcontroller was used as the Sea Perch onboard computer (see block diagram in Fig. 4). A major difficulty with designing a Sea Perch onboard computer is the need to drive some relatively high-powered motors. The DC brush motors commonly used have a stall current of 3A which exceeds the current rating of commonly used Arduino motor shields.

To solve this problem we utilized a two dual motor shields that can supply 12A per motor (Dual VNH5019 Motor Driver Shield, Pololu Corp. Las Vegas, NV).

Reliable communications over a 30 foot tether precluded the use of the standard Arduino serial interfaces: TTL serial, USB, I^2C and SPI. We chose the RS-485 communication protocol for its ability to work over tether-length distances, its simplicity and its ability to be networked (for future additions). One of the Arduino's TTL serial ports was converted to RS-485 with a converter board (Sparkfun BOB-10124). Communications was half-duplex to save most of the wires in the tether for power. We utilized a CAT-5e cable for the tether; one of the 8 twisted pairs was used for RS-485 communications and the remaining 3 pairs carried power. At the surface, a second TTL serial to RS-485 converter board was used to convert back to TTL serial for communication with the topside computer.

The sensor suite for the first iteration of our system consisted of the same Bosch BNO055 sensor board as used in the Sea Perch flight recorder. Sensor data was simply logged on a microSD card or transmitted to the surface, it was not utilized in the control of the vehicle.

The topside computer consisted of another Arduino Mega interfaced to a wireless Sony Play Station 2-style game controller (Model RC-01 v4, Lynxmotion/Robotshop, Swanton, VT) via a TTL serial port connected to the game controller's wireless receiver. The game controller served as the ROV Pilot's control interface. The joysticks and buttons on the controller could be assigned to any one or combination of the Sea Perch motors. For first tests we simply had the game controller emulate the manual Sea Perch switch box. The Arduino could also display messages and vehicle data on a laptop computer screen using the serial monitor function of the Arduino integrated development environment (IDE). Both the topside and onboard Arduinos were programmed in the Arduino programming language.

In operation the user manipulates the buttons and/or joysticks of the game controller which streams data to the topside Arduino. The game controller data are translated into a series of ASCII commands and sent to the Arduino on the Sea Perch which interprets and executes the commands. We operated in this direct control mode for the first iteration of our software. Local control utilizing feedback from sensors will be implemented in the future.

The onboard computer system was mounted in a NEMA 6P submersible fiberglass box (Part # 2266K14, McMaster-Carr, Elmhurst, IL). The tether and the cable to the motors passed through the box using waterproof cable glands. The box was mounted in the Sea Perch payload bay as shown in Fig. 5. The system was successfully tested in the tank in our lab.

Discussion & Conclusion

We have developed two enhancements to the Sea Perch ROV that significantly extend the educational utility of the vehicle, especially for high school students and college undergraduates. The gray box flight recorder takes students beyond the intuitive "feel" for how underwater vehicles behave to quantifying that behavior in terms of orientations, accelerations and rotation rates around three axes. This device connects the Sea Perch to more advanced concepts in high school physics and math. As built, the flight recorder's capabilities can be extended by simply utilizing more of the data that the Bosch sensor can produce such as heading relative to magnetic north. Concepts of sampling theory can also be explored by varying the sampling rate for a given external stimulus (e.g. regular waves) and observing the effects on the data plots. Our first educational experience with the flight recorder was positive. The students had fun smashing their vehicles in to walls, throwing them in the water, and otherwise driving them to extremes. This provided a powerful motivation for them to plot and analyze the resulting data.

The onboard computer pushes the Sea Perch into the realm of professional ROVs. It enables students to incorporate a variety of sensors and actuators onto their vehicles. The Sea Perch is still very limited in available power and space so the students will be challenged to design and build new and novel actuators that are not commercially available in order to perform interesting tasks. Often the Sea Perch and similar ROVs are used in contests and then left on a shelf until the next contest rolls around a year later. The onboard computer changes this dynamic by enabling students to make the Sea Perch a useful device for their own ocean or pond explorations year-round.

Acknowledgements

This project was supported by the MIT Sea Grant College Program, the MIT Undergraduate Research Opportunities Program, and the MIT Office of Engineering Outreach Programs.

References

[1] T.R. Consi, "Introducing Students to Ocean Engineering," in *Proceedings of the 2002 ASEE Annual Conference & Exposition*, Montreal, Quebec, Canada, June 16-19, 2002.

[2] G.A. Wright, R.C. Hurd, K.S. Hacking, and T. Truscott, "Using ROVs to Teach a Blended STEM Curriculum," in *Proceedings of the 121st ASEE Annual Conference & Exposition*, Indianapolis, IN, USA, June 15-18, 2014.

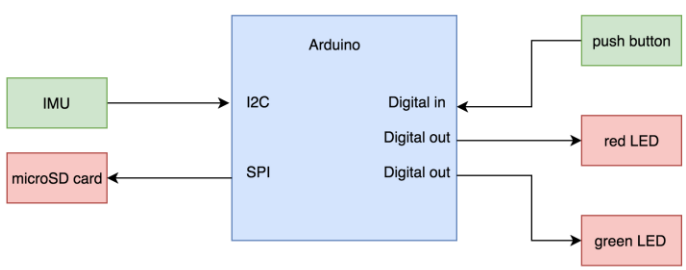

Figure 1. Block Diagram of Sea Perch Flight Recorder. IMU is the Bosch BNO055 intelligent 9-axis absolute orientation sensor described in the text.

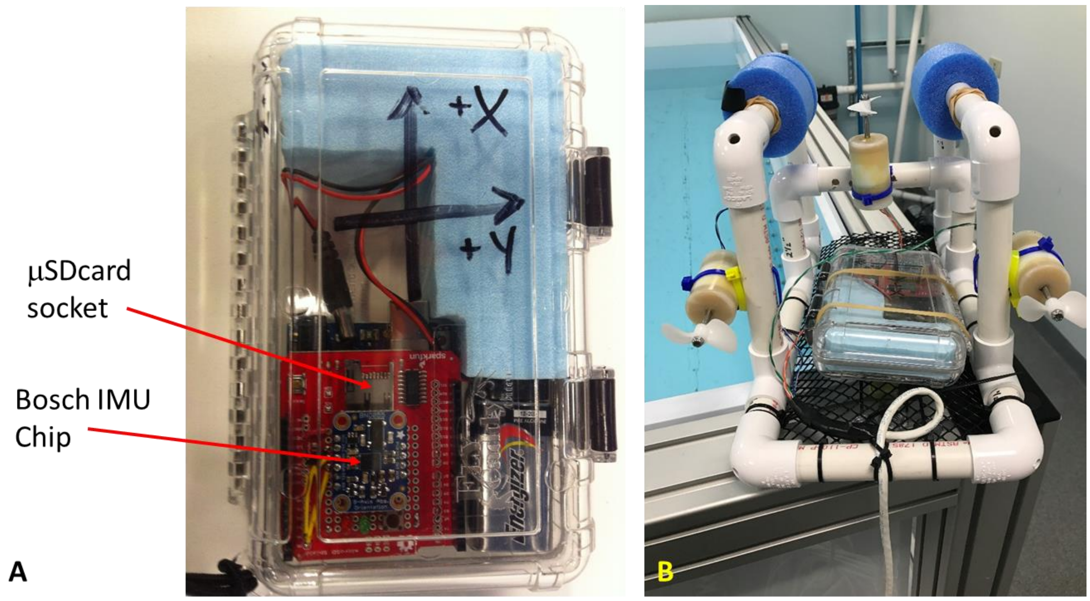

Figure 2. A. Photograph of the Sea Perch flight recorder in its waterproof box. The Bosch sensor is in the center of the small blue board soldered to the microSD (µsSD) card shield, the Arduino UNO is underneath. Axes designations are drawn on the cover to aid in orienting the device in the Sea Perch payload bay. The box is 16 cm x 11 cm x 4 cm deep. **B.** Photograph of flight recorder mounted in the Sea Perch payload bay.

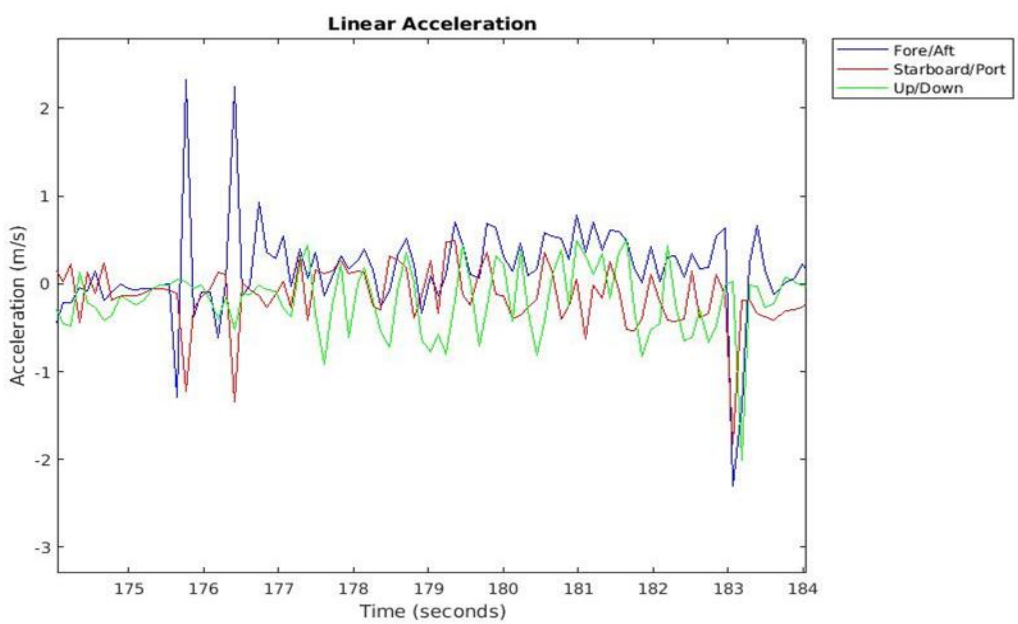

Figure 3. Acceleration vs Time plots of several Sea Perch crashes into the walls of the testing tank. Forward crashes are to the left, a rear-ender is to the right.

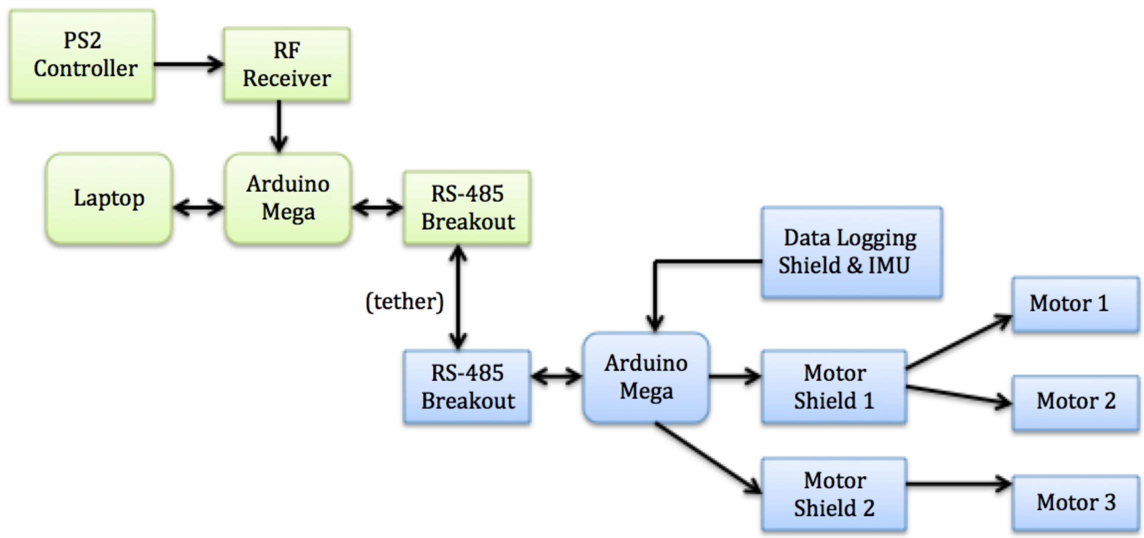

Figure 4. Block diagram of the Sea Perch onboard computer system. Blue boxes are onboard the Sea Perch, green boxes are topside.

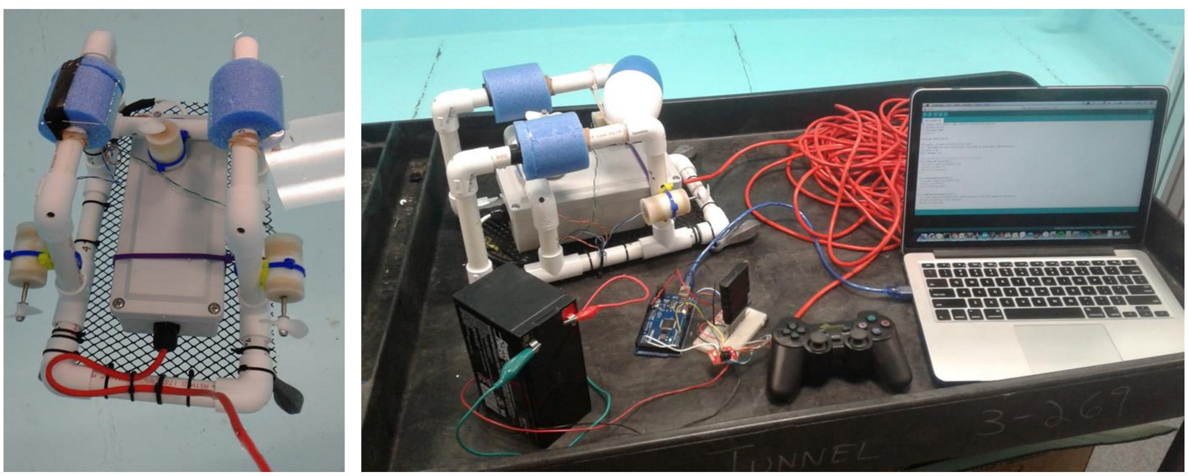

Figure 5. Left, onboard computer mounted in payload bay of a Sea Perch. The box is 21 cm x 9 cm x 8 cm deep. Right, complete system. Clockwise from upper left: Sea Perch with onboard computer, tether (30’ of Cat-5e cable), laptop computer, PS-2-style game controller, game controller receiver, Arduino MEGA topside computer, and 12V battery.