University of North Florida

OSPREY DIVERS

Team Photo Spring 2015: Not all members are present in this photo.

Website: www.ospreydivers.org

Team Members:

Marshall Curry (Team Lead), Eric Rutherford, Kerry Anyanwu, Brandyn Sloan, Samantha Busch, Wenjing Liu, Brian Zamore, Marcus Blessing, Brian White, Stephanie Olson, Vedran Pehlivanovic, Joel Avalos, John Hellmann, Stefano Di Bella, Demetrius Brown, Rob Grunwald, Brian White, Juan Mata, Jakob Howe, Nicholas Emord, Timothy Vu, Christopher Searcy, Ryan O’toole, James Rutherford, Michael Otero, Stewart Fletcher, Michael Burkhardt, Will Gualtieri, Ashlee Lithalangsy, Maria-Flora Jacobs, Charlie Santana, John Gans, Andrew Turgeon, Jeremy Cantor, and Jonathan Frias.

Team Advisors:

Dr. Daniel Cox – *Professor of Mechanical Engineering*

Dr. Brian Kopp – *Assistant Professor of Electrical Engineering*

Abstract

The University of North Florida's RoboSub Team, the Osprey Divers, would like to present AUV-SPREY, its first vehicle entry in AUVSI Foundation and ONR's 18th Annual RoboSub Competition, "Back to TRANSDEC," located in San Diego July of 2015. The Osprey Divers team was founded November of 2014 by members of UNF's TeleRobotics Club. The goal for the Osprey Divers was to design and build an Autonomous Underwater Vehicle for the competition. AUVSI's competition allows the club to get a unique systems engineering approach to solving a real world problem, while implementing and improving the expertise of the dedicated UNF students. The Osprey Divers is an interdisciplinary team of UNF students, bringing together students majoring in Electrical and Mechanical Engineering, Computer Science, Business, and any other major at the University of North Florida.

One of the main emphasis' for the competition and the Osprey Divers has been outreach, working with the school and other organizations to get students involved in the STEM field and inform them of how important STEM fields are to the community.

1. Introduction

Approximately 6 months of effort has been dedicated to the Osprey Divers' RoboSub this year. As a first year team, simple goals were placed. The goals for AUV-SPREY are to maneuver through the gate and participate in the buoy, time portal, and refuel bin (dropping markers) tasks. Once these tasks are mastered, the sub will be integrated for more capabilities. This year will set the momentum for the years to come. The team is broken down into subsystems which include frame, propulsion, camera, sensors, power, controls, and business.

The Osprey Divers had two main design ideas. Plan A was to implement an underwater quad copter propulsion system and use the Pixhawk flight controller, interfaced with an Arduino Mega to maneuver through the pool. Once research and tests were done on the first prototype, results concurred that the time until competition did not allow for proper control of this system. This design will remain as research and development for further years to come. Therefore, Plan B was executed, using the Arduino Mega as the central control board, interfacing with the raspberry pi2 and the camera, and the sensors, receiving inputs, and distributing the necessary outputs to the thrusters.

2. Frame

Below, in Figure 1, is the CAD model of AUV-SPREY, completed using Solidworks.

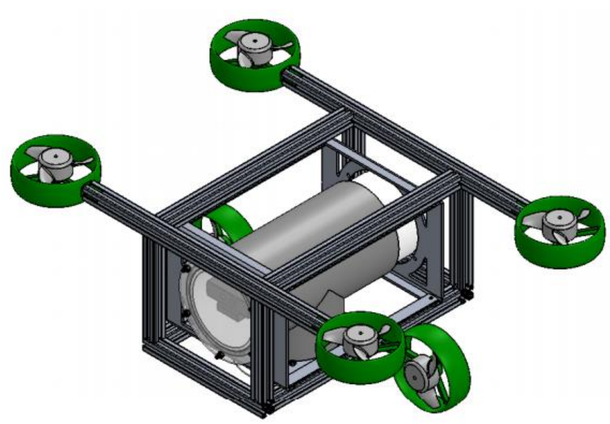

Figure 1. AUV-SPREY Solidworks CAD Model.

2.1 Main Hull

The final frame design consists of one main hull made of a schedule 40 PVC t-connector fit for 6” diameter pipe, and an extension of 6” PVC to give more space for components and electronics inside the sub. Two levels of racks allow for easy access to electronics. The three openings to the hull are each sealed with an acrylic window bolted to aluminum sheet metal and made watertight with custom fit 6” diameter O-Rings. The forward and bottom facing openings of the t-connector PVC allow for camera compartments. The camera is held in place with 3D-printed ABS plastic parts, the acrylic windows giving the camera an interface to the water outside of the sub. The rear opening, also sealed with an acrylic window has various waterproof connectors attached, allowing for a convenient interface to components on the exterior of the AUV.

2.2 Space Frame

The main hull is surrounded and mounted to a frame made of 80/20 aluminum. This allows for mounting external components to the sub and easy integration of future additions. There are six thrusters used in this design. Four thrusters mount to the four corners of the AUV to allow for elevation and roll control, and two thrusters on each side of the AUV to allow for horizontal movement and yaw. The AUV is slightly positively buoyant to alleviate stress on thrusters, reducing power consumption, and acting as a safety feature.

Overall, the calculated net buoyant force of the t-connector hull is 23.4 lbs. Therefore, slightly less than 23.4 lbs. in space frame, electronics, and external components allow for stability. Fine tuning was done to the buoyancy to make the AUV as stable as possible when stationary.

3. Propulsion

The propulsion system is made up of four thrusters controlling vertical movement and roll, and two thrusters controlling yaw and horizontal displacement. The thrusters were made from T-Motor, 380 KV brushless motors as shown in Figure 2. Shrouds and propellers were attached to the motor as shown in Figure 1. Electronic Speed Controllers were interfaced with the Arduino Mega to accept PWM commands to control the motors.

Figure 2. T-Motor, 380KV Brushless Motors.

4. Camera

4.1 Description of camera, software, Microcontrollers and protocol

The camera subsystem is made up of a raspberry pi 2 computer, Arduino Mega, embedded software and the creative Senz3D camera. The raspberry pi 2 is powered by a 900 MHz quad-core ARM Cortex-A7 CPU with a volatile memory of 1GB. The computer also includes 4 USB ports, 40 GPIO pins (General Purpose Input/Output), full HDMI port, an Ethernet port, combined 3.5mm audio jack and composite video, camera interface (CSI), display interface(DSI), Micro SD card slot and a VideoCore IV 3D graphics core. The raspberry pi 2 is used to perform the computational demands required for image processing, it also establishes serial communication via USB protocol between the cameras and also data transfer from the pi 2 to the Arduino Mega. The computer also provides power to the camera via the universal serial bus. The Arduino Mega is solely the master controller for the entire AUV system platform. It receives the requested data from the image processing component and then makes a decision based on the data received and then sends that data to whatever subsystem that requires that information. The embedded software used by the camera subsystems are as follows, raspbian, openCv framework, and language API (Application Programming Interface). The raspbian OS is a Linux distribution that is embedded in the raspberry pi 2's core. The image processing and color detection and machine vision framework was implemented using the openCv application programming interface. This library provides a rich set of easy to use modular functions. The implantation of the framework was via python language for the raspberry pi 2 and C/C++ for the Arduino Mega. This subsystem utilized the creative Senz3D camera. This is a relatively inexpensive camera but packed with power features such as 1.0 megapixel HD (1280 x 720) sensor, auto lens focusing, built-in Mic with noise-cancelation, and maximum frame of up to 30fps (frames per second). The camera captures the frame and sends that information via USB protocol to the image processing component for image analysis and processing. The camera is also used to search for objects as well.

4.2 Technical Decision Process

The initial prototype for the camera sub system was to use a pixy camera for visual sensing, interfaced with a PIC microcontroller to receive and interpret the data. The pixy implemented a blog detection system for image processing and did not provide a rich library of functions to work with. The camera had a limited use of seven color signatures and did not provide a way for the developer to implement a visual logging system for troubleshooting. The PIC microcontroller was a 16bit controller made by Microchip. The microcontroller packed an ample amount of features such as i2c, SPI, interrupts and more. Executing these features proved difficult, with limited online help and coupled with the time constraint it was decided by the camera team to abandon these technologies and to use simpler technologies that provided an ample amount of tutorials and references. Such technologies used were the raspberry pi 2 computer, Arduino Mega, and the creative Senz3D camera. The raspberry pi 2 was chosen because of its compact form factor and features such as its 900 MHz quad-core ARM Cortex-A7 CPU. Also the computer is relatively inexpensive meaning it can be easily replaced if damaged. The Arduino Mega was chosen as the master controller because of its ease of use, rich API library and plentiful references. The microcontroller is also relatively inexpensive. The creative Senz3D camera was chosen because of its HD and auto focusing features. The entire technology used by the camera subsystem was based on budget, ease of use and time constraints.

4.3 How the System Works

A Creative Senz3D Depth and Gesture Recognition Camera is mounted to the front of AUV. The camera has 720p HD quality video shooting quality, depth recognition, and features 3D sensing. OpenCV, Raspberry Pi, Arduino, and Python programming languages were used for computer vision. Software is written in Python to utilize OpenCV to visually detect object and identify color and shape. Raspberry Pi transfers the data to Arduino using serial communication protocol for further analysis and actions. Figure 3 shows the camera system diagram.

Figure 3. Camera Systems Flow.

4.4 Issues Encountered

Technical problems were a barricade at almost every step of development. The Osprey Divers' camera team initially chose the Microchip brand as the micro controller board. However, upon further research and testing, the functions used in the "C" language to work with the PIC micro controller board were too complex to understand in such a limited amount time afforded to our team. Getting the PicKit3 device (used for compiling and programming the target device, or PIC microcontroller) was an elusive task as well. To further complicate matters, public reference material was scarcely available to troubleshoot problems with the PIC microcontroller code. Later, the camera team came to realize that the Pixy image tracking camera was inadequate. The Pixy camera's shortcomings included the following: limited ability in capturing shapes, only capable of recognizing 7 colors, inability to properly redefine its API to suit needs and inability to record. Recording is critical since it is necessary for troubleshooting purposes. Getting the distance from the updated camera, to an object was not a straightforward task. Other methods had to be created to get a sense of how far away an object was. The camera nor the code had the ability to provide the focal length necessary to determine distance to the object. It seemed desirable to use the i2c communication protocol between the Arduino board and its slave devices. However, using i2c communication proved to be an overly exhaustive task considering the amount of initial setup for the software on the devices.

4.4 Synopsis of the Algorithm

OpenCV is an open source computer vision library that is available on a variety of different operating systems including Raspbian, a free operating system for the Raspberry Pi. OpenCV implemented in C++ but can easily be interfaced with high level programming languages such as python which was language of choice for this project. From experience, python is capable of expressing concepts in a fewer lines while maintaining readability of the code.

OpenCV provides a vast library of different computer vision and machine learning algorithms built into functions that can be conveniently called upon via python. The challenge is to identify what algorithms and tools available will be deemed useful for the various part of the competition.

4.4.1 Color Detection

All the challenges involved in the competition involve dealing with objects (gates, buoys, etc.) that are conveniently color coded. OpenCV's inRange() function takes an image and the color boundaries as parameters. In computing, the color of a single pixel is represented by 3 bytes with each byte representing the intensity of red, blue, and green respectively. The function returns an image that excludes the pixels that do not fit within the specified boundary. Figure 4 shows the original alongside an image returned from inRange().

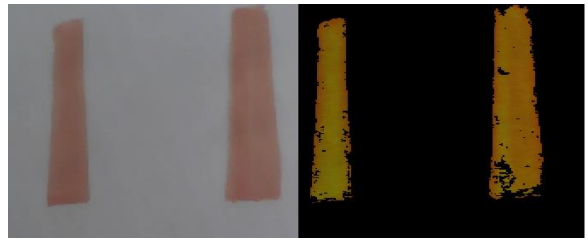

Figure 4. Original image alongside image returned from inRange().

4.4.2 Contour Detection

Geometrically, a contour is a continuous curve along a boundary that has the same color or intensity. The findContour() takes an image as parameter and as the name implies it return a list contours within an image. Each contour within the list is a collection of points or edges that represent the contour. This is used in combination with the inRange() color detection function to identify a shape of a colored object. Figure 5 shows the approximating bounding rectangle around detected object.

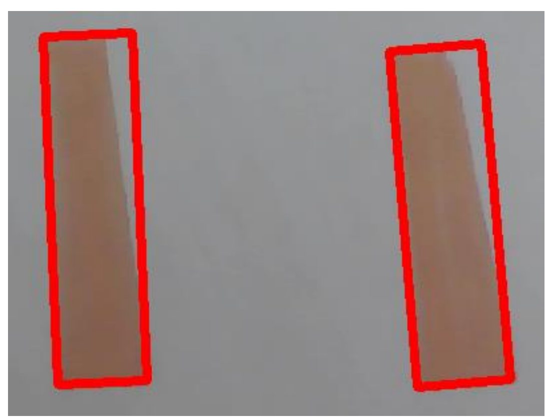

Figure 3. Approximating bounding rectangle around detected objects.

4.4.3 Hough Circle Detection

The HoughCircles() function takes an image along with a minimum and maximum size of radius given in pixels. It returns the coordinates where the center of circular objects are detected within the provided image. Using this in combination with color detection will be used to detect colored buoys. Figure 6 shows an example of a circle detected within an image.

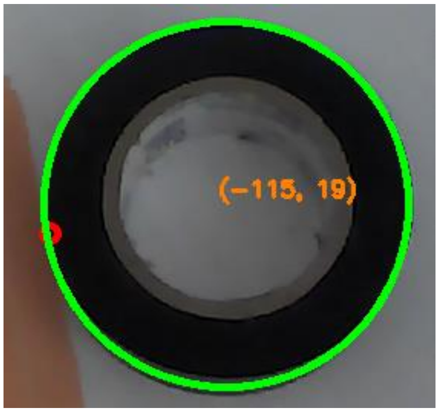

Figure 1. Circle detected within image.

5. Sensors

AUV-SPREY uses a system of sensors to manipulate the orientation and detect the surrounding environment. A microcontroller communicates with the rest of the sub to manipulate the sensors. An Arduino Mega microcontroller was ideal because the difficulty of use is low which was appreciated due to the limited time that the team had. The communication between the sensors and the Mega is i2c.

5.1 Magnetometer

A digital compass is used to sense the direction of the sub so it could be redirected to the next task. For the digital compass, the HMC5883L chip is used. The HMC5883L is a magnetometer that can measure the magnetization's strength of different objects. By sensing the magnetic field of Earth, the magnetometer is able to find the direction of North and set North as a reference point.

5.2 Gyroscope and Accelerometer

A gyroscope and accelerometer are used to maintain the balance of the sub. The gyroscope and accelerometer are included in the MPU-6050 chip. The gyroscope senses the change of the angles from the axis of the sub. The accelerometer measures the forces from acceleration on the sub.

5.3 Pressure Transducer

For the vehicle to know at what depth it was, a pressure transducer is used. The Model K2 pressure transducer calculates a voltage output that varies dependent on depth. To prevent the sub from going too deep or shallow, a threshold was made by finding the voltages of the limits and programming the Arduino to stay between the voltages. The voltage output from the pressure sensor is fed into the Arduino Mega's 10-bit ADC. The pressure range recognized by the transducer is 0-100 psi, making the output voltage range 0-4.5 volts. This voltage range was divided amongst the 10-bits of the converter and intervals for each depth were made. The controller reads the digital signal from the ADC and determines the depth of the sub.

6. Controls

The main control board used is the Arduino Mega 2560. The board communicates to the raspberry pi2 via serial communication through USB. The Arduino also communicates to external sensors using i2c. The Arduino Mega receives signals from the peripherals, being the camera and other sensors, processes the data, and distributes commands via PWM to the ESC's that control the six thrusters. An emergency stop switch is interfaced with the outside of the sub to provide emergency disconnect of all batteries from the circuits.

7. Conclusion

As a first year team, the goal for the Osprey Divers is to at least qualify and complete at least three of the tasks. After testing of the first design, the team opted for the backup plan because of simplicity. With the experience gained this first year, the team has momentum to produce a more sophisticated AUV next year. The goal is to keep integrating systems to make the sub better in every aspect. Most importantly, the experience that this competition gave, and will give in July, to the individual students on the Osprey Divers' team is priceless and one that is not always found in the classroom environment. As years progress, and more experience is gained, the Osprey Divers will be a force to be reckoned with.

8. Acknowledgements

This team would not have been possible without the support of the Osprey Divers' sponsors and the individuals that gave his and her time and dedication to the progression of the AUV. Sponsors include the College of Computing, Engineering, and Construction Management (CCEC) at the University of North Florida, Taylor Engineering Research Institute, Microchip, and UPS. Special thanks also goes out to Dean Mark Tumeo, Dean of CCEC, Dr. Murat Tiryakioglu, director of the School of Engineering at UNF, Dr. Don Resio of Taylor Engineering Research Institute, Dr. Daniel Cox, Mechanical Engineering advisor, and Dr. Brian Kopp, Electrical Engineering advisor. Last but not least, thank you to the anonymous donators to the go fund me account, the friends and family members who have donated, and the dedicated members of the Osprey Divers who donated, or as Dean Tumeo put it, "provided skin in the game."