East Los Angeles College Robotics

Dana Page, Steven A. Lemos, Ediberto Medrano, Owen T. Wolf

I. Introduction

The East Los Angeles College team *Robotic Huskies* will construct a fully functional autonomous underwater vehicle (AUV) from scratch that will compete in all challenges for the 2016 RoboSub competition and lay a foundation of knowledge for future teams to learn from and build upon. The East Los Angeles College Robotics Club also views this competition as an opportunity for members to learn new skills and gain practical hands-on engineering experience.

II. Mission, Vision, Goals and objectives

A. Mission

The East Los Angeles Robotics Club wants to motivate aspiring engineering and technology students to continue their chosen educational pathways. The club encourages active engagement in the STEM field by providing opportunities for academic development, professional development, leadership development, chapter/club development and community outreach.

B. Vision

The East Los Angeles Robotics Club works to prepare its members to become practicing STEM professionals that will leave behind a legacy that future generations can look up to and aspire towards.

C. Goals

The East Los Angeles College Robotics Club members will design, build, test and modify a fully functional autonomous underwater vehicle (AUV) to compete in the 2016 RoboSub Competition and place in top ten finalists.

D. Objectives

- The team must create necessary prototypes and final competition ready drone capable of successfully completing all challenges in the 2016 competition using a modular design
- Build up from small prototypes to a final complex drone, with each prototype increasing in complexity and features--identified by code names Alpha, Beta, and the final product, Paco
- Systematically increase testing to expose flaws and refine design
- Build team members as individuals and gain experience working on a complex team project
- Extend team members knowledge base and lessons learned to future generations through outreach events and detailed documentation
- Lay a path for the future club members to follow

III. Design Strategy

In order to achieve stated objectives the team will employ strategic planning by setting and staying on a proper timeline, stay focused and disciplined in assigned tasks and not having a *good enough* attitude. Another important key for success is having respect for other team members and opinions, as well as keeping up team moral. The leadership will focus on finishing as many tasks as possible early on so that the team will have as much time as possible to test and debug for control, safety and stability of the robot.

As the team has decided to attempt all the obstacles present in the RoboSub competition, the design of the robot itself must allow for all the necessary components to successfully finish the course in its entirety. At the same time, it has been decided that the design should also remain as simple as possible in order to minimize points of failure. Keep it simple is the phrase that governs the teams decisions in the design phase.

IV. Vehicle design—mechanical

The team at East Los Angeles College is new to building such a complex machine, thus two prototypes were built in order to better understand some of the basic concepts and systems that would be needed for a competition ready drone. Codenamed *Alpha* and *Beta*, each increased in their size and complexity working up to a final product, named Paco.

A. Alpha:

The goals for the Alpha prototype is that it must be made of easily accessible and inexpensive materials, work in water, move in 3 dimensions and require minimal programming to control. Movement is achieved using three Blue Robotics T100 thrusters attached to the PVC frame and controlled by Electronic Speed Controllers (ESCs). The ESCs receive those signals from an Arduino Uno running a relatively simple program that reads, converts and relays information from a USB shield connected to an XBOX 360 controller. A 12-volt power supply will is used to provide power to the thrusters and Arduino. Neutral buoyancy is be achieved by adding weight and pool noodles to the frame as needed. The weights and pool noodles are held in place using plastic zip ties.

B. Beta:

The Beta prototype must be larger, more stable, and the carry a water tight container to house electronic devices which should safely route wires through the container via watertight plugs. Beta is controlled by a driver's station docked out of the water, transmitting data via Ethernet cable. After mastering manual control, more components are added modularly, such as the camera housings, torpedo launchers and the grabbing mechanism.

The frame of the Beta prototype will still be made of PVC pipes and joints found in most hardware stores. The watertight casing (hull) are also be made of a PVC pipe, O-rings, PVC end caps and small acrylic plates. These hull components are held together with marine epoxy and silicone based adhesive. Inside the hull are three ESCs, an Arduino Uno with an Ethernet shield attached, an internal temperature and an internal pressure sensor. DC power is delivered through a 100 foot tether from a battery outside of the water. Ethernet data is transmitted through a 120 foot category 5E Ethernet cable. The data tether is terminated at the Ethernet shield on the Arduino and a driver's console on the opposite end. DroneOS client built on C# processes manual controls from an XBOX 360 controller attached to the computer and executes some basic pre-programmed autonomous functions.

C. Final Build (Paco)

Though several designs were considered (see fig.1), the team ultimately decided the best way to hold all of the electrical components at a minimum of cost and machining was to use a Pelican 1500 case. The Pelican 1500 case is made from ABS plastic, provides reliable water tightness and can be easily modified to suit the team's specific needs. The Pelican case also provides 50 lbs. of buoyant force. The frame onto which all of the external components of the drone are attached is made from REV Robotics 1-inch extruded aluminum purchased from AndyMark.com and held together using #10-32 x 3/8 screws and #10-32 nylock jam nuts. In order to create slight elevation from the bottom of the frame, wheels were added to each of the four corners giving the drone approximately three more inches in height. The wheels also aid in the transportation of the drone.

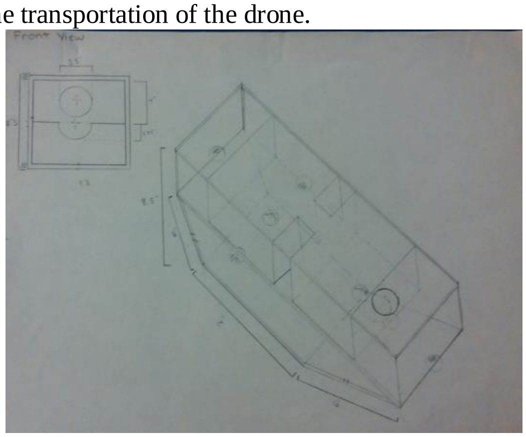

Fig. 1. Multi-hull concept sketch

D. Hull Penetrators

Any wiring needing to pass from the inside of the case to the outside is passed through hull using either 3-pin or 8-pin SeaCon Bulkhead Penetrators purchased from CrustCrawler.com. These penetrators allow us to easily pass electronic signals from the internal (watertight) part of the case to any and all external (water submerged) components with the added benefit of quick disconnect and connection change capability. Blue Robotics Hull Penetrators, purchased from BlueRobotics.com were also used, however the permanent-non-disconnect nature of the Blue Robotics Hull Penetrators required them to be used for more specific applications like the camera enclosures where the ability to disconnect and reconnect was not a priority.

E. Grabber Design

The design of the grabber was perhaps one the more challenging problems the team faced while designing the robot. Many different prototypes were considered, however when it came to the actual nuts and bolts of the grabber and how it would function, the team struggled to find a workable, reliable and easy-to-use solution. Ultimately the team decided a 3D printed grabber would allow for much faster prototyping and modifications. The basic claw design was found on www.thingiverse.com/thing:1480408 with the name Mantis Gripper. The claw wasn't originally intended to be used underwater, but as the team planned to 3D print it and use waterproof servos, there was no reason why it could not be used underwater. After downloading the claw's STL file from the website, it was modified using SolidWorks 2015 3D modeling software and made suitable for our needs. The modified claws were then 3D printed using PLA plastic on a MakerBot Replicator 2 (see fig. 2). The most efficient and effective way of mounting and securing the washers and bolts needed for the grabbers turned out to be gluing the washers and bolts down. Silicone was applied to the inside of the claw's pinchers to give the drone the grip that was necessary underwater. With that done, the claw was essentially finished, and a second claw was constructed with the same procedure as the first.

Fig. 2. 3D printed grabbers

Mounting the first claw was simple as a metal bracket was used to mount it onto the front of frame of the drone. This claw was the front facing claw positioned horizontally. The second yielded an obstacle as it was mounted onto a piece of custom cut acrylic, which was all mounted onto a bottom support bar of the submarine, it ended up protruding 5 inches out of the bottom of the frame. Therefore the vertically positioned claw facing down was made to be retractable in order to solve the protruding aspect. A simple solution was implemented of just using a hinge, a spring, and a gate latch (see fig. 3).

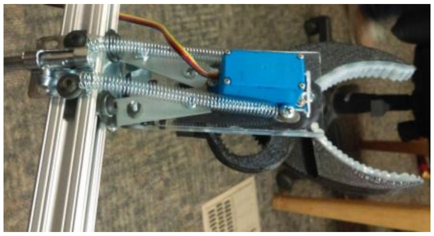

Fig. 3. Mounted retractable grabber

F. Torpedo

The initial idea of this torpedo design came from a desire to mimic an actual torpedo, one that has real propulsion and thrust. A dart that just shoots out of a barrel was not satisfying enough. Some ideas that were proposed included battery powered propellers or compressed air. The team considered the compressed air route would be much easier to make and cheaper since CO_2 cartridges are easily available. The torpedo itself is made from a 3/4" PVC pipe, a 12 gram CO_2 cartridge, commonly used for airsoft guns, three fins and a torpedo head was designed using SolidWorks and 3D printed using the MakerBot Replicator 2 with PLA material (see fig. 5).
The torpedo is loaded into a launcher that will puncture the CO_2 canister within the chamber to give the torpedo its initial thrust. The launcher is comprised of a PVC coupler to hold the torpedo, another 3/4" PVC pipe on the other end of the coupler that houses a spring-loaded mechanism that will strike a firing pin to puncture the CO_2 canister. Two modified PVC end-plugs were used to encase the spring and also hold the firing pin made from a brass tube and rod (see fig. 4).

Fig. 4. Torpedo launcher

Within the launcher is a zinc-plated steel hex bolt with a 9/16" socket affixed to the head. The bolt goes through a 45-lb. spring and PVC end-plug. The socket was used as a guided hammer to strike the firing pin as the diameter was close to the inside diameter of the PVC pipe, making a smooth and straight slide through the barrel. A hex screw was attached to the socket to be used as a trigger to decompress the spring. The spring is compressed by using a hex nut to tighten the bolt from the outside of the barrel.

Fig. 5. PVC torpedo and 3D printed head and fins

When the trigger is released, the 45-lb. spring will decompress and strike the brass firing pin to puncture the CO_2 cartridge inside the torpedo to cause pressure to build up within the chamber between the torpedo itself and the PVC coupling. The pressure of the CO_2 being released will give the initial thrust of the torpedo and continue to propel the torpedo forward until the CO_2 cartridge is empty.

G. Cooling

Within the hull the electrical components and the motherboard are housed, as well as part of the cooling system. Heat buildup within an enclosed system—like the sealed inside of the Pelican case—can cause real damage to electrical systems. With that in mind the team decided to add a cooling system for the main processor of the robot. There were three types of possible cooling systems that were discussed for the drone. The first two would use external sea water in order to cool the processor or circulate the hot air passively within the hull. There are major disadvantages to both, the first cooling strategy would have required some type of filter for the sea water as to prevent contamination of the system. Creating and implementing a filter still does not guarantee that there would be no contamination and that is why that strategy was abandoned. The second strategy of passively cooling the internal air would have its own limitations. There would be a possible chance of overheating with solely circulating the internal air of the hull. For these reason an active cooling system utilizing a sealed loop concept was chosen, which yielded the benefit of no contamination because distilled water is used inside of the sealed loop. The cooling system is both internal and external. The external components of the cooling system including hard line copper piping and two aluminum heat syncs which are directly mounted to the frame. Theoretically six heat syncs are necessary however in conjuncture with the coils being passively cooled by the air, and the cold temperature of the water once the drone is submerged the two heat syncs are sufficient. Inside the hull is the pump circulating the distilled water throughout the soft lined tubing which takes the heat created by the processor out. The hot water then moves down the hardline copper tubing which has its own cooling properties, as well as the two aluminum heat syncs which lower the temperature of the distilled water, resulting in a system that has a very low possibility of overheating.

H. *Camera/Camera Enclosure*

Water tightness, a visibility port and a way to transmit signal were the key requirements for the camera enclosures. The team started prototyping with PVC and ABS pipes and using the appropriate cements and binders to connect them as well as provide the first layer of water tightness. Plexiglas was cut to the required dimensions and used to make the viewport for each enclosure. The Plexiglas was kept in place and made water tight using marine epoxy around all edges and connecting surfaces. Blue Robotics Cable Penetrators were used to allow the signal wire to pass from the enclosures to the main hull with marine epoxy used to seal any spaces between the wire and the cable penetrator. To add another layer of water tightness a silicone based adhesive was used at connection points within the enclosures.

For our final competition ready camera enclosures aluminum casings were machined to desired specifications to replace the PVC and ABS plastic. Plexiglas and the Blue Robotics Cable Penetrators were still used along with marine epoxy to ensure water tightness and reliability.

I. *Thrusters*

There are a total of eight Blue Robotics thrusters on the drone, six of which are T100s and two T200s. The two T200 are positioned horizontally at the midpoint of the top longest piece of extruded aluminum on the frame, which allows for control of the forward and backward motion of the drone. The T200s have a maximum forward thrust of 11.2 foot-pounds and a maximum backward thrust of 9.0 foot-pounds. Thrusters were also on the frame in order for the drone to be able to run sideways. The remainder of the four T100 thrusters were placed on the extruded aluminum frame in order to propel the drone up and down. The T100 thrusters have 5.4 foot-pounds of maximum forward thrust, and 4.1 foot-pounds of maximum backwards thrust. All the thrusters are made from high strength, UV resistant polycarbonate. The bearings are high performance plastic bearings that do not rust, with a motor sealed with epoxy to ensure that it is watertight. The entire thruster is composed of high strength plastic, aluminum and high quality stainless steel. There are all 3.9 inches in diameter and weigh less than one pound.

F. Lessons Learned

During Alpha testing, the team discovered that the USB host shield reserves some of the Arduino pins for I/O (input/output) functionality. ESCs plugged into those reserved ports caused our thrusters to rotate randomly without any input or signal. Thus those PWM signal pins could not be used. Pins 5 and 6 of the Arduino board were redirected to different PWM enabled pins on the Arduino USB shield labeled *Tinkerkit* ports in the middle of the shield instead of the normal pins on the side of the board.

The entire construction process was thought out, drafted by hand and then modeled on SolidWorks, but still required much trial and error. It was not until construction that we moved different components around to see where they fit best in relation to each other, most of the time they were placed were they were planned. The take away of this experience was to begin the building session as early in the process as possible, even if it is just using placeholder components. Seeing the proposed settings of certain pieces of the robot were essential to understanding the best design decisions and needed modifications to the design.

V. Vehicle design—electrical

The design for the electrical systems is based on a simplistic, innovative and reliable system. The team used an Arduino microcontroller to quickly prototype and program the robot.

Paco features the following sensors:

- Temperature (Blue Robotics TSYS01)
- Depth (Blue Robotics MS5837-30BA)
- Gyroscope & Accelerometer (MPU-6050)

Pc Hardware:

- Computer Hardware Architecture
- Intel i5 Processor @ 3.5 GHz
- 4gb RAM
- 64GB Solid State Drive

Other Hardware:

- Genuine Arduino Mega 2560
- 8 Afro ESCs

Power:

- Custom power distribution board
- Two 10 Amp / hour batteries
- Optional tethered 40 AMP server power supply supplemented by SLA battery

VI. Vehicle design—Software

A. *DroneOS*

The purpose of the programming team is to create the artificial intelligence engine and a diagnostics suite for the drone. The system is set up to be easily modified for future RoboSub challenges or even for personal use. It includes a GUI (Graphic User Interface) system to preprogram basic tasks. DroneOS is also equipped with the ability to switch between manual control via an XBOX 360 controller or on screen buttons.

The DroneOS Console enables multiple users to connect to Paco to view live streamed video of raw and processed camera images. The Console also enables users to pass manual control to each other, modify mission plans, select which obstacles to complete, and acts as a diagnostics platform for debugging.

DroneOS Paco features a full diagnostic suite for determining any points of failure during untethered testing. During the initial test periods there were many issues with unexpected shut downs and restarts.

B. *Features*

The main goal of the software team is to create a robust and easy to use program Software Architecture. The software team wrote a custom operating system for Paco.
Languages used:

- C++ Arduino,
- C# for Client and Server Software

- XML for data storage
- Windows Server Core 2008 to run .NET applications in high speed and ease of use for future students.

For diagnostics, an Ethernet tether connected to a custom console.

C. Image Processing

Image processing is accomplished by using the open source EmguCV C# wrapper. EmguCV is based on OpenCV, another open source C++ image processor. There are 4 threads that process 2 camera feeds. One thread captures a camera image, processes it by looking for shapes in certain colors depending on the obstacle being undertaken. A processed frame and raw frame are both stored for each camera image. A separate thread runs a decision algorithm to decide what to do with the processed image for the respective obstacle. A separate thread handles broadcasting image and sensor data to all connected console clients.

Each camera processes at a default thirty frames per second (FPS). The process rate is adjustable for each camera depending on mission needs or to conserve processor time. The range is between 1 FPS and 60 FPS.

Additionally, a separate thread runs to broadcast the raw and processed frames of both cameras to any connected clients. The FPS of the live stream can also be adjusted accordingly. Each frame is compressed to less than a megabyte to conserve bandwidth over wireless connections. This has the added benefit of allowing us to broadcast images live on our website http://elacengineering.org.

D. Autonomy

Paco contains an advanced set of features to enable itself to navigate the course successfully. The main feature enables Paco to keep on a level course at all times. If Paco gets off track, an algorithm will run to enable it to try to backtrack and find its way. This is achieved using a combination of time-out features that determines if it has not met a certain requirement in a certain amount of time. Paco also features a system that calculates and predicts how long each obstacle is taking and can elect to skip certain obstacles if it does not think it can accomplish them in time, if there are other obstacles that have a better chance of being successful and completing—Paco will move on to complete those instead.

E. Communications

Communication between our Arduino and drone is a standard serial connection. Serial packets are built using a byte level packet creator for motor, servo, and message control. Sensor data is serialized using JSON. Synchronous TCP/IP is used to communicate between the drone and any connected consoles. TCP connections are AES-512 encrypted.

VII. Experimental Results

Testing and redesigning are essential parts to the engineering design process. With that in mind the team has been testing, redesigning and retesting all parts of the drone as often as possible.

Starting with the Alpha prototype every single component that has gone onto every version of the robot has been independently tested before it is mounted to the actual robot.

These individual component tests ensure that each piece that is connected to the finished final version of the robot has been proven to work long before it reaches the frame of the final *Paco* design. All eight Blue Robotics thrusters have been indecently tested, in the water, running off the Arduino microcontroller, as have each of the waterproof servos, grabbers and torpedo launcher. As each of these components have been tested separately and on their own, if any problems arise during later stages—it can be relatively assured that any problems are due to connection issues or programming errors and are not related to the individual components themselves.

The water tightness of the drone as a whole has been tested in several stages. With the Pelican case alone and unaltered, to ensure it is water tight from the beginning. Then after the frame was mounted to the Pelican case, which necessitated the drilling into the case, the whole assembly was again tested to see if it remained water tight. Next, the Bulkhead penetrators and cooling system were added and the entire thing was once again tested for leaks. This time leaks were detected from where the cooling system's copper tubing passed through the case, those leaks were sealed with marine epoxy and have remained water tight since.

It should be noted that all these tests were done is standard sized swimming pools with a maximum depth of 10 feet. In order to be fully assured of the quality of the seals and water tightness of the case, further testing is required at greater depths. Fortunately, East Los Angeles College has an Olympic sized swimming pool which the team may use to test the drone in.

Once shallow depth water tightness had been confirmed the team began testing the drone's maneuvering capability with remote controlled tethered testing. In this stage the drone is powered and controlled via tether cable, it is in no way running autonomously or under its own power. However, these test do allow us to get a better understanding of how the drone moves in the water as well as where the actual center of gravity is on the drone as it moves through the water and how much buoyancy it has. Tethered testing has been underway for approximately 3 weeks prior to the submission of this paper.

Tethered testing has also allowed the team to find unanticipated failures that could have proven disastrous if they had not been discovered prior to competition. One of the custom machined aluminum camera enclosure developed a leak during tethered testing. Luckily it was a small leak that was easily remedied by the liberal use of marine epoxy. But had that leak occurred at competition, there is the distinct possibility that one of the cameras could have been damaged and rendered inoperable—which would have left our robot half blind and severely hindered. The camera enclosure failure also prompted the team to recheck other point of possible failure just in case, ensuring added care and attention as we approach completion.

VIII. Acknowledgements

The following groups, companies and individuals have helped us on our road to competition and we would like to thank them all…without their support our RoboSub would be just a dream, they have helped us make it a reality

From East Los Angeles College:

- The Engineering & Technologies Department
- The Kinesiology Department
- The Associated Student Union (ASU)
- ELAC Athletics
- ELAC Plant Facilities
- ELAC Community Services

From the Montebello Unified School District:

- CTE Pathways

From local Industry:

- LEABBS and Jose Luis Solano
- Romakk Engineering
- Blue Robotics
- CrustCrawler Robotics

Software

- EmguCV
- Arduino
- Gource

2016 Team Members:
Steven Lemos (Team Lead), Owen Wolf (Software Lead), Ediberto Medrano (Electrical Lead), Richard Naverette (Business Lead), Adam Acosta (Mechanical Lead), Krystal Bernal (Software), Jose Medina (Electrical), Dana Page (Mechanical), Michael Robles (Business), Alex Jong (Mechanical), Bryan Solano (Mechanical), Brian Vasquez (Faculty Advisor)

IX. Outreach

Giving back to the community and sharing the joy of learning is something that can never be overdone. It has been our great pleasure to be involved with the Community Services Department of East Los Angeles College and their efforts to make Monterey Park a better place.

This year's RoboSub team has hosted and volunteered for several *Engineering FUNdamentals* session at East Los Angeles College. Engineering FUNdamentals Workshops are sets of 5 week courses for elementary school children ages 8-12, where each set has a concept theme for the kids to learn about. Our goal for the workshops is to introduce the next generation of students to the STEM fields through entertaining lectures, then solidifying that newly acquired knowledge with hands on experimentation and project-based application of those very same engineering concepts. Our members volunteer to teach and support these classes, which are offered through the ELAC Community Extension program. Some of the concept themes we have taught include Electricity and Circuits, and Programming with Arduino.