A.Q.U.A.L.U.N.G (Autonomous Quadrotors Utilizing Attack Logic Under Navigational Guidance)

Yousef Al-Khalaf, Christopher Culver, Shane Parker, and Bianca Wood

Dept of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — The objective of this project was to build a team of four quad rotors that would maneuver themselves autonomously and play laser tag competitively against human opponents. This research has the potential to make contributions to the advancement of autonomous robotics and practical application of artificial intelligence techniques. The goal is to make the team of quad rotors act and respond in real time and make intelligent tactical decisions to provide a genuine challenge for their human opponents.**

***Index Terms* — computer vision, mobile robots, motion detection, robot programming, technological innovation.**

I. INTRODUCTION

The A.Q.U.A.L.U.N.G. project's most direct practical application is for entertainment purposes; to be able to play laser tag competitively against human opponents. The quad rotors would add a different type of opponent than laser tag players are used to with these small, agile, aerial robots. Not only can this project be applied for entertainment purposes, but it can also be used for training of police or military personnel. The quad rotors could be used to observe and record training simulations currently being conducted by these factions, and allow a more thorough review of these training sessions.

Creating an autonomous robotic system, such as the team of quadrotors used in the A.Q.U.A.L.U.N.G. project, requires the implementation of dozens of technologies and the design of a coherent means of combining them. To facilitate an organized approach, every component of the project is matched to one of four subsystems. These subsystems are: artificial intelligence, power management, flight control and computer vision.

Each subsystem is interconnected with the others in a manner that is straightforward, making their integration a more manageable task than trying to connect all of the systems to each other individually. The general organization is that every subsystem feeds information to the AI system, which uses this information to make real-time tactical decisions. Then the AI subsystem sends navigation and fire control commands to the flight control subsystem, or a shutoff command to the power management subsystem when applicable. Additionally the AI subsystem assists the Computer Vision system by providing information about the environment, such as expected edges given an individual quad rotor's position and orientation.

II. ARTIFICIAL INTELLIGENCE

The artificial intelligence component of the project is very important because of the autonomous nature of the goals. The decision making portion is the heart of the AI system, the position of the quad rotors will be registered and a best possible move will be calculated. The AI system will also receive information from the camera and will decide whether to shoot/not shoot and possibly navigate based on sight, it it sees an enemy turn toward it/ follow it and attack.

A. *Path Finding*

Path finding is an integral part of this project as the quad rotors will be moving autonomously. Without a well-defined path finding algorithm the quad rotors can potentially crash into obstacles, walls or even the human opponents.

While there are a few alternatives, the path finding algorithm that is expected to perform best in this critical application is the A* algorithm. A* is a best-first search method commonly employed in path finding solutions. [1] The algorithm sorts through a given representation of the area through which the path must be found, and returns the path with the shortest distance. The algorithm employs a heuristic to improve its performance, but the heuristic must be chosen carefully, as underestimating makes the algorithm less efficient, and overestimating can cause it to not return the shortest path. [1] There are three common representations of the environment that are used with the A* algorithm: a grid representation, a way point graph, and a navigation mesh. Regardless of the specific representation initially research, an interruption feature was going to be included, primarily to avoid collision with a human opponent if they move into the path of the quad rotor. The quad rotors are able to be gen an updated path before they finish following a path they are currently on; the system will remain dynamic at all times. This will allow for the quad rotors to safely navigate in the presence

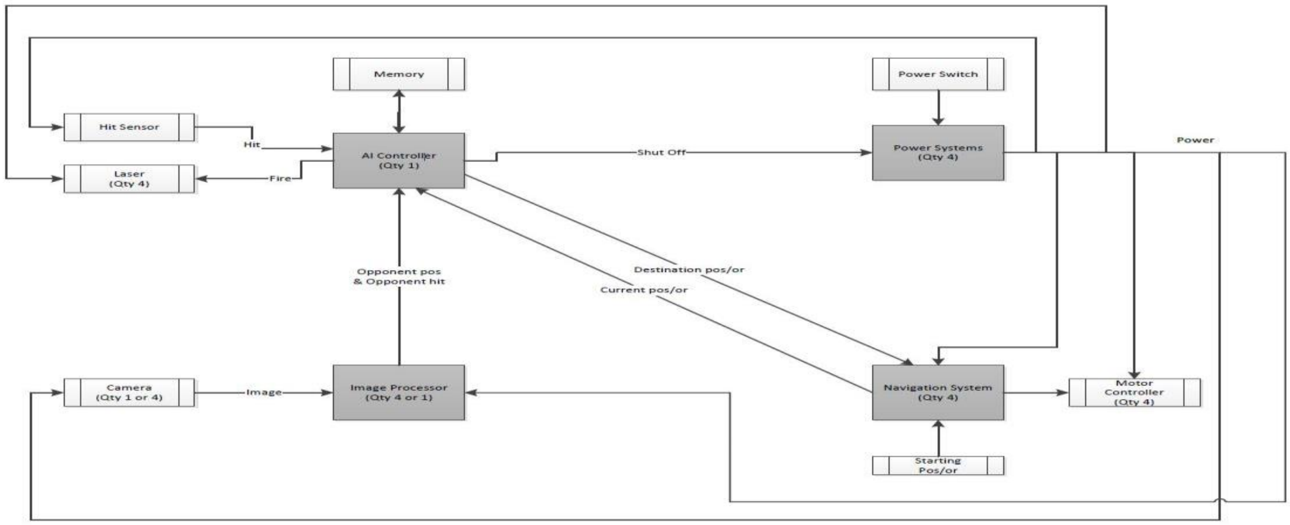


Figure 1: Main Subsystems Block Diagram.

of dynamic obstacles as well as update their position objectives in response to the behavior of their opponents.

A navigation mesh path is the most direct path of the three options (grid representation, a way point graph, and a navigation mesh) because it allows for the most direct path. A navigation mesh allows for a description of the parts of the environment that can be moved in freely by populating it with convex polygons. [2] Each polygon will store information about its vertices and the neighboring polygons that can be traveled to The center of each polygon is a node that the A* algorithm will use as a starting point for creating the path to follow, but since information is available describing all moveable areas, the path can be improved, creating the most direct possible path through the environment. An example of a navigation mesh is shown in Figure 1.

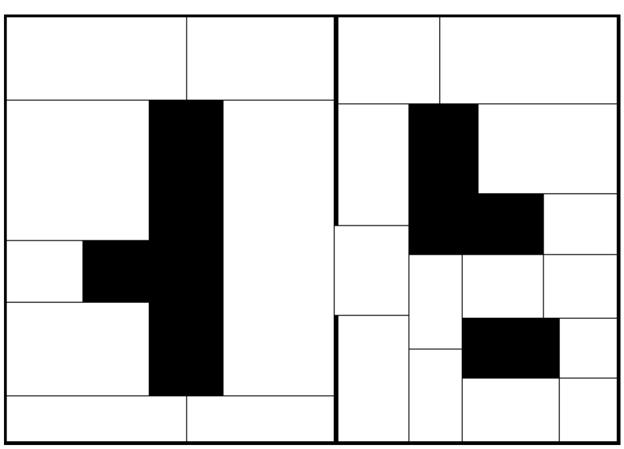


Figure 1. Navigation Mesh Representation of an example environment

The heuristic used is the Euclidean distance from the evaluated node to the target node if there are no obstacles in the way. The use of polygons to represent large chunks of the search space allows for improved efficiency in the paths, and for rapid determination of the path to use. The positive aspects of using a navigation mesh to represent the environment make this method very attractive. The downside lies within the complexity added with creating more efficient, direct paths. The processing time is not really a concern, as altering the path only involves a small number of nodes once the A* algorithm has finished running. Rather, it requires the developer to design a final step that wraps the path around any obstacles and draw straight lines as often as possible. The method for this step that will be used in the A.Q.U.A.L.U.N.G. project is to draw a straight line from the starting point to the ending point, and then shrink the path the A* algorithm returns to this line. The result is the best possible path that can be found using line segments.

B. Decision Making

The decision-making aspect of the AI subsystem is the high level control that binds together and directs the actions of all of the other functions and algorithms contained within the subsystem. This part of the AI looks at all of the data presented from the other subsystems, as well as data returned from within the AI subsystem, and then sorts through this information for the most relevant bits and uses these to decide on the next course of action for each quadrotor. The high level decision-making technique for the AI in the A.Q.U.A.L.U.N.G. project that is employed is a customized variant of a state machine.

A state machine typically is composed of a handful of differing states, each one defined by the actions performed during the state, and the switching conditions to other states. When in a state, the actions will loop endlessly until the conditions occur to switch to another state, which will then loop its actions. This allows for a multitude of behaviors to be designed and run with explicit conditions when one behavior switched into another. The primary change that is employed with the use of a state machine in the project is that every state performs the same actions.

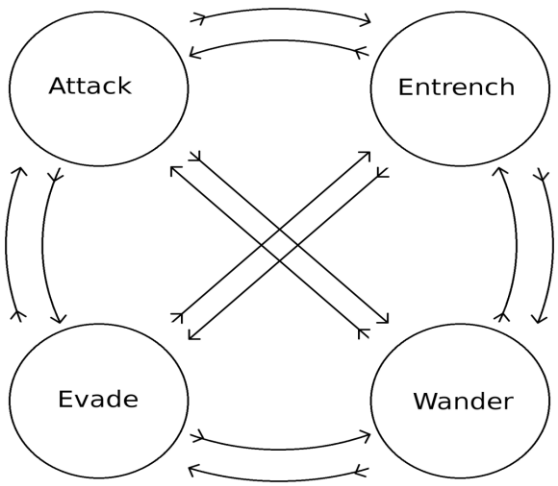


Figure 2. State diagram for behavioral processing

During each state the tactical planning system is constantly evaluating the environment and providing a list of suggested target locations to the state machine based on the prioritization of environmental features. Each set of target locations is weighted based on the features of the set, such as maximized control over critical control points, or proximity to cover zones, and so forth. The state machine will then make a determination of which set to use. Once a set of target locations is chosen, these will be provided to the A* path finding algorithm. The A* algorithm will return a sequence of locations, representing the vertices of the path discovered.

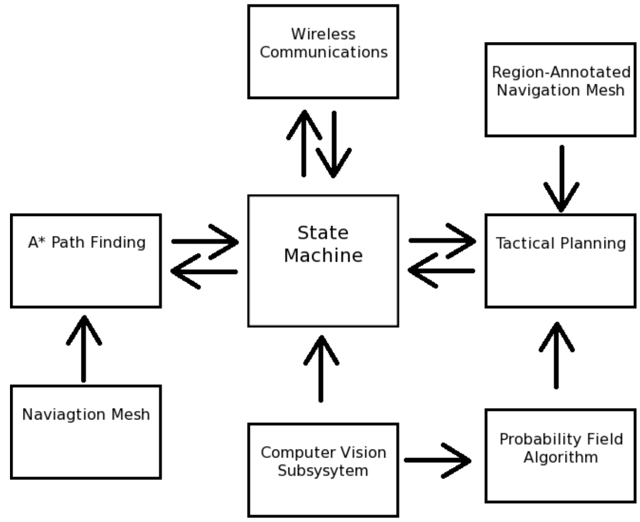


Figure 4: Organizational Layout and Data Flow of AI Subsytem

III. POWER MANAGEMENT

Power management for this type of system requires a lightweight battery that is rechargeable and easily interchangeable. Lithium-ion polymer (Li-Po) batteries are commonly used by hobbyists that work with helicopters and airplanes because of their ability to deliver high capacity for a reasonable cost. For this project, it only makes sense to pick this type of power as well.

Choosing two batteries allowed for the distribution of power to be more effective than just using one battery. Half of the components that need power will be powered by one battery and the other half by the other, allowing for a longer flight time while at the same time not weighing down the quad rotor.

A. Power Distribution

It was also necessary to regulate the voltage going to the batteries. The batteries are going to connect to the printed circuit board and the PCB is the main power distribution center in conjunction with the voltage regulator that came with the Arduino microprocessor.

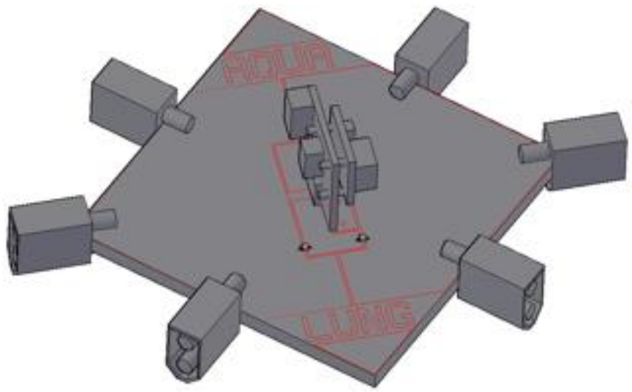


Figure 5. AutoCAD image of the PCB

IV. FLIGHT CONTROL

Flight control is the main component of stabilization and a partial component of the navigation, which is executed by IMU stabilization and dead-reckoning, respectively.

A. *IMU Stabilization*

The first act of stabilizing the aircraft comes with the set up of the motors and propellers. As shown in Figure 3 motors 1 and 3 have to be rotating clockwise while motors 2 and 4 rotate counterclockwise. This cancels out the torque of each motor and creates stability in the yaw, the spinning in the z direction. Just as the motors need to spin in certain directions, the propellers need to be pitched in certain directions. The propellers on motors 1 and 3 being pitched to achieve lift for clockwise motion, and the propellers on motors 2 and 4 being pitched for lift for counterclockwise motion.

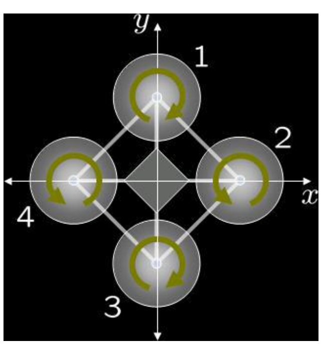


Figure 3. Motor orientation

For the stabilization in the pitch and roll, an Inertial Measurement Unit (IMU) was used. IMUs are electronic devices used on moving vehicles that measure its velocity and orientation, as well as gravitational forces. Two components are quintessential on an IMU, an accelerometer and a gyroscope, a third component is sometimes added for farther accuracy when tracking position, this component is a magnetometer. For this project, an accelerometer and gyroscope were incorporated. With the IMU correctly mounted onto the aircraft during flight, the movement of the aircraft can be computed at all times.

As a quick example, let's use the motor configuration in figure 3 and relate it to figure 4 to describe motion of the aircraft. If the IMU sends raw sensor data to the microcontroller, the microcontroller will then do some calculations to determine the rotation of the aircraft in all three axises, yaw, pitch and roll. At this particular point in time the yaw and roll have not changed angle, however the pitch has changed angle negatively. This means that the quadrotor in figure 3 would be leaning to the right, and to stabilize the microcontroller needs to output more voltage to motor 2, as to speed up the rotation. While it sends less voltage to motor 4, slowing down rotation, this action will level out the quadrotor. This check and balance is done many times a second the whole time the quadrotor is in flight, thus guaranteeing stabilized flight.

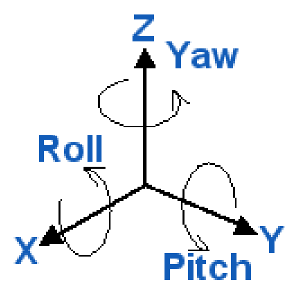


Figure 4. Rotational Movement

B. *Navigation (Dead-Reckoning)*

In order for the AI controller to send out instructions on where the quadrotors will go next, the quadrotor has to relay its position in space back to the central computer. This feat will be completed with dead reckoning, a method for finding ones position using previous position, and velocity over a time period. The accelerometer outputs acceleration, so to get position out of it gravity will have to be removed and the integral will need to be taken. Once this is integrated the velocity is now found, one more integral will yield a position. The gyroscope outputs a rate of change, which is velocity, so this output will only need be integrated once to find position.

Errors in the integrations are corrected by being filtered out. Accelerometers usually have more noise than gyroscopes, which leads to faster drift. Drift being how far off of the actual position the measurement is reading. Since a double integration is needed, any residual bias causes errors that grow with the square of time, so it does not take long for dead-reckoning that is uncompensated to become unstable with accelerometers.

With all this in mind, another navigation scheme will have to be implemented along with dead reckoning in order to give the A.Q.U.A.L.U.N.G. the accuracy it needs to complete its mission. The use of computer vision is an answer to this challenge, as it is a system to check up on the dead reckonings position and reset the system if needed. The way it works is the quadrotor will roam the playing field relaying the position it thinks it's in back to the AI. Whether its position is correct or not will be determined by symbols throughout the playing field, these symbols will all be distinct and have a fixed location. So if the quadrotor believes it is in the back left corner of the playfield however the camera sees the symbol for the middle left of the playing field, the AI can tell the microcontroller to reset its location to the middle left of the playing field and continue with its mission.

This duel system is also not the most accurate navigation tool known however for the purposes of the A.Q.U.A.L.U.N.G. project and with the resources available, it is a fine solution to the navigational problem.

V. COMPUTER VISION

This project is very software intensive. At the basis of all the hardware, there is tons of code that backs it up. All of the coding for the computer vision portion was written using the openCV libraries in Microsoft Visual Studios 2012.

A. Motion Detection

A.Q.U.A.L.U.N.G's motion detection will deal strictly with the optics portion of the six ways of detecting motion. Using the openCV libraries combined with AForge source libraries, A.Q.U.A.L.U.N.G was able to be programed in visual studios to be able to detect humans on the stream. To do this, two frames difference motion detector along with simple background modeling motion detector were combined to create a custom way of detecting humans quickly and accurately. Two frames difference motion detector strictly deals with comparing and contrasting two consecutive frames and finding the main differences. If the comparison goes above a certain threshold, it may be concluded that motion is present.

The next form of motion detection algorithm that was implemented was the simple background modeling motion detector, which focuses on finding the difference between a frame representing the background, "known-environment," and the current video stream. The coding and algorithm is very simple and quick. It uses techniques as training the algorithm to detect any changes in the frame with respect to the original, non-interfered basic background frame. The main reason for deciding to most likely use this detection algorithm is because as seen in figure 6 the highlighting is extremely accurate since the algorithm can instantly and easily detect the un-known object "human" in the current frame.

It is also possible to use a combination of the previous two detection algorithms. Doing so the best of both algorithms are combined into one, very powerful, accurate and quick detection. First, the program and algorithm will be fed the original non-interfered known environment snapshot frame and the algorithm will always use this as a reference. As soon as motion is detected, simple background modeling motion detector will be the first to accurately detect it. Once the target, "human" begins to move, two frames difference motion detector algorithm will come to life and keep a very accurate update on the current position of the target detected. A.Q.U.A.L.U.N.G may simply have to use a custom combination of the two motion detector algorithms to combine into one very strong and accurate one.

Figure 6. Simple Background Detection

B. Video Processing

Starting at the camera on the quadcopter, the first frame ever sent to the main computer will be that background of the main known environment. With that information, the computer may now know what do detect changes in. For example, if a person walks across the room, the detecting algorithms will understand that the person was not originally in the known environment and treat it as a target. The way this process happens is the following:

1) Camera uploads stream to offsite server, usually a website at a minimum resolution of 520p.
2) The main computer system will then access the stream that has been uploaded and will link to it in real-time downloading the video frames.
3) After the link is established and the stream is up and running on the computer then all necessary thresholds will be applied to the frame.
4) Once motion is detected within that frame/stream we move onto next step.
5) It is confirmed that this object is in-fact a human and mark it as a target.
6) Next, we may be able to resize the processed image to have it in the correct size that the processing algorithm may work with efficiently.
7) This is the most important step. Here, the motion processing algorithms go to work efficiently and accurately either shading or bordering in the target.
8) Lastly, the completed processed image/frame is ready to be sent on to the targeting algorithm to fire the laser at.

C. Wireless Communication

This project is working entirely off wireless communication between the quad rotor prototype and the processing computer to make real-time updates and adjustments to each and every motor at an exact specific time. To do this, a reliable, accurate, and quick wireless device is mounted on the quad rotor; including an Xbee microchip operating at a 2.4 GHz frequency band. Also used is a Zigbee wifi chip, as seen in figure 7.

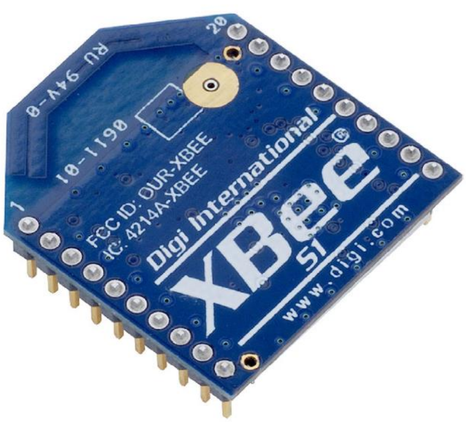


Figure 7. XBEE Microchip

D. Camera

In order for the motion detection and processing algorithms to work correctly, the camera must be of high quality and light-weight to not draw any unnecessary power from the batteries to keep the quadcopter in flight. It was decided that we would be using a wireless HAMY 2.4 GHz camera for this. The camera is very light weight, 2 ounces, has an extremely small casing, as seen in figure 8. The main concern was the way the camera received power, through a 9 volt battery. This was later compensated for while tweaking the stability of the quadcopter. The camera sends the video stream to a DVR-USB wireless 2.4GHz 4-channel receiver, which is connected directly into the main frame. The range for accuracy of the stream, would be at a distance no more than 20 feet.

Figure 8. 2.4GHz Wireless Camera

VII. Testing

A. Hardware Testing

To test the hardware systems of the project, a gradual process of allowing increased range of motion for the vehicle was employed. Initially each of the motors was tested individually with a brief, five second control signal. This was done with no propeller attached to the motor. The next phase was to try controlling all four motors simultaneously, again with no propellers. The next phase was to attempt controlling all four motors at the same time, but with independent and varying signals. Once it was determined that all four motors were responsive to control and could be controlled independently of each other with the given setup, the quadrotor was firmly secured and the propellers were attached. This test demonstrated that the quadrotor is capable of delivering a substantial excess of power, allowing for very rapid acceleration. This would allow for the quadrotor to move far more quickly than would be safe for indoor user, or with any direct interaction with human beings. As such the maximum output allowed has been limited to ensure operation within a safe range.

The next phase of testing required moving out doors and giving the quadrotor a greater range of motion, while still keeping it restricted. This testing stage was purely to see if the quadrotor could maintain flight under its own power, without any turning or advanced control. This test demonstrated that the quadrotor is capable of self-powered and maintained level flight, but the addition of any wind caused drastic instability.

The final hardware test is free motion flight, with a user directly controlling motion through the use of a ground station controller. This testing is still ongoing.

B. *Software Testing*

A large portion of the software is still being developed or finalized, and as such testing of these components is not completed. That being said several software components have been tested.

The first such component is the communication protocols over the Xbee modules. To test the system a simple application was designed on each end of the communication to send a simple message, and if it was received and understood echo it back. This allowed for verification that commands were properly encoded and interpreted through the connection. This also allowed for a range test of the modules themselves. The effective reliable range is approximately 100ft.

The next software component that required testing was the flight control software for the microcontroller, as well as for the ground station. To fully test these components independently of the hardware is not possible, as such the full capability of the stabilization algorithm and reliability of the automated control system have still to be tested in a rigorous manner. The initial testing of this software has revealed that the ground station is able to track the position of the quadrotor in real time and send control commands without any noticeable lag. On the microcontroller side, testing has shown that the flight control algorithms are functioning properly to turn control commands into the appropriate motor control signals.

VIII. Conclusion

The ambitious scope of the project necessitated the utilization of diverse systems and components. These range from basic voltage regulation, to wireless communication, to advanced computer intelligence systems.

Several limitations were encountered in the systems and components that necessitated the employ of creative problem solving techniques or adaptation of the constraints of the project as a whole.

The most substantial of these limitations is the range to quality trade-off of the wireless video system. While the system does stream a video that is interpretable by the human eye over a greater than 50 feet (depending on environmental conditions such as line of sight and interference from other devices), the quality degrades rapidly to a point where modern video processing would not be able to detect human figures or accurate motion in real time.

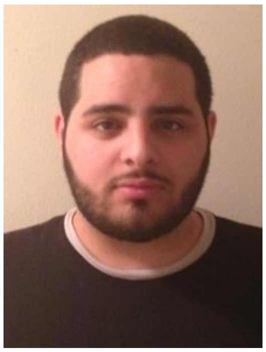
Another limitation encountered is the fact that the microprocessor package used in this project is designed to be used with a traditional radio control set-up. All the features to calibrate the motor response range and tune for level flight require an RC unit to be hooked up to the system. Such a unit represents a substantial investment that was not budgeted for and thus is not feasible to obtain. To move forward with the project despite this set back involved calibrating the entire flight system by hand. This meant an extensive trial-and-error process to manually set functioning and responsive control ranges.

Despite numerous setbacks in the acquisition of the proper parts, and unforeseen difficulties in implementing various solutions, this project has been, and continues to be exceedingly gratifying. Each member feels a genuine sense of accomplishment and joy every time something starts working as is desired and expected. While costing far more than planned in time and stress, no member of the group would choose another project if given a second chance.

Acknowledgement

The authors wish to acknowledge the assistance and support of their sponsors; John Enander, Jeff Moler, and Tim Parker.

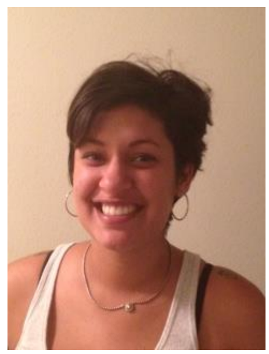
AUTHORS



Yousef Al-Khalaf, a senior in computer engineering at the University of Central Florida. He is going to be graduating with his Bachelor’s in May 2013 and is planning on pursuing a Master’s in Business at Florida International University.

Christopher Culver, a senior in electrical engineering at the University of Central Florida graduating with this BSEE in May 2013, with a physics minor. He is presently working with Lockheed Martin, Missiles and Fire Control where he has been for the past year and plans to work full-time after graduation as a RF Engineer.

Shane Parker, a senior in electrical engineering at the University of Central Florida graduating with his BSEE in May 2013. He is currently doing research in intelligent systems with Dr. Avelino Gonzalez and after graduation plans to pursue his PhD at the University of Central Florida.



Bianca Wood, a senior in electrical engineering at the University of Central Florida graduating with her BSEE in May 2013. She is currently working with Lockheed Martin, Mission Systems and Training, where she has worked since May 2009. She plans on staying with Lockheed after graduation as an Electrical Engineer and later on pursuing a Master’s. She is a member of the Society of Women Engineers.

REFERENCES

[1] (Lester, 2005) source from SD1 paper
[2] (AI-blog.net reference)