Autonomous Drone

Team Members:

Nick Contini

John Thomas

May 13, 2015

Abstract

The following is a report detailing Miami University’s entry vehicle in the Autonomous Aerial Vehicle Competition of 2015. All design specifications are provided at <http://www.flyaavc.org/rules>. To locate the ball the “right wall” method was used to navigate towards the second room while avoiding any objects. Upon entering the second room and reaching the back left corner the quadcopter would begin searching for the ball and ALVAR markers so that it could begin setting up the local coordinate system of the room.

Table of Contents

Contents

[Introduction 4](#_Toc419466735)

[Research 4](#_Toc419466736)

[Initial Research 5](#_Toc419466737)

[Object Avoidance 5](#_Toc419466738)

[Image Processing 5](#_Toc419466739)

[ALVAR Detection 5](#_Toc419466740)

[Design and Implementation 5](#_Toc419466741)

[Flight Control 6](#_Toc419466742)

[Navigation 6](#_Toc419466743)

[Object Avoidance Components 6](#_Toc419466744)

[Object Detection and Image Processing 7](#_Toc419466745)

[Safety 8](#_Toc419466746)

[Ground Station 8](#_Toc419466747)

[Discussion 8](#_Toc419466748)

[Conclusion 9](#_Toc419466749)

[Future Work 9](#_Toc419466750)

# Introduction

Last year a group of students got together to design a quadcopter for the Air Force AAVC competition held in Dayton, Ohio at Wright Patterson Air Force base. Since drones have become a large part of the military with their cost effective use and being very cheap to replace, the Air Force is looking for a way to use drones to locate a target within a building where GPS systems are denied. GPS is not always going to be available especially in large buildings or even deep underground in tunnels so the drone must be able to navigate without it. In such confined spaces GPS would also not be a very viable option for navigating a building as it would not be very accurate at times. Currently, this kind of infiltration and identification is done by soldiers which places their lives at risk and in such confined spaces can be deadly to a whole squad. If replaced by an autonomous drone the squad will be able to see what is in the next room or within a building without ever having to take a step inside and will be able to identify a target from a safe distance. This type of challenge is not easy to complete though and takes a lot of time to complete. In the first year that the team was working on the project we were unable to complete enough to be able to compete the first year. Though this was the case it set up a great foundation for when this became our senior design project.

In order to have a controlled environment that all teams can be prepared for and have an equal advantage in, some basic ground rules and test field were set up and prepared for the participating teams. For the rules, all teams must have a form of kill switch to stop all functionality built into their quadcopter so that should something go wrong it can be stopped immediately. All teams would only have ten minutes to complete the course but the amount of time in which the course was completed would not affect your overall score. The setup of the course began with the first large room which had a designated area where objects, likely boxes, would be set up to hinder the path of the quadcopter from being able to just fly straight through the room. All boxes could be no closer than five feet apart. After passing the first obstacle there would be a stationary wall towards the back center of the room that would need to be flown around to find the doorway to the next room. Upon entering the next room there would be a green ball on a pedestal placed randomly somewhere within the room as well as ALVAR markers that would be located on the walls. These ALVAR markers would be used to set up a local coordinate system to determine the approximate location of the ball within the second room. With these ground rules we determined what components would be most effective at navigating, measuring distance, and locating an object to add to the quadcopter.

# Research

Collectively, our group introduced multiple solutions to the tasks at hand which brought up the need to test what components or methods would work best as compared to the other options. Our decisions were made based on prior knowledge of using different components on other projects.

## Initial Research

Starting off we had to learn exactly how the quadcopter operates and how the flight controller controls each of the motors to make flight possible. With the basic knowledge of how the quadcopter works we moved on to determining how to control the flight controller without the use of a physical controller. It was determined that the best way would be to use either a Raspberry Pi or a Beagle Bone Black microcomputer. Initially some testing of using the programmable real-time unit on the BeagleBone Black was done due to its high efficiency. Unfortunately the greatly increased the complexity of the project as doing pulse width modulation in assembly proved to be very difficult and so because of this we decided to continue with programming on the Linux image of the Raspberry Pi.

## Object Avoidance

Both object detection modules that were presented need to make decisions based on how close to an object the quadcopter is so an Arduino Mega microcontroller was chosen to interpret data received from either sonar modules or a laser range finder and communicate the information received back to the Raspberry Pi. Both modules have their benefits and their cons in different situations so it was necessary to test multiple cases in which these devices will perform.

## Image Processing

When the quadcopter has made its way into the last room it will need to be able to do image processing on the objects that will be located within the room. For this our group looked into what image processing libraries that are available for use and decided to use OpenCV. Much of our decision is due to the open source license on the library which allows us to use it for free but some of our members also have had past experience with it and have shown that it is a very powerful library. With this library we will be able to locate the ball.

## ALVAR Detection

ALVAR markers are actually QR-like markers originally used for Augmented Reality functionality. A library has already been written for the purpose of detecting these markers. By using the library, it will be possible to implement a program that detects the markers and associates specific coordinates with each marker. This library is implemented in C++ and is open source. It depends on OpenCV, but luckily it is already part of our project libraries.

# Design and Implementation

After researching the performance of the different components introduced we began on a design that should allow the quadcopter to navigate the designated course. Each component chosen was decided upon for the benefits that it would provide over other components.

## Flight Control

To control how the quadcopter will fly a Raspberry Pi was used as it proved to be sufficient enough for pulse width modulation (PWM) control and the camera can be easily positioned at any angle. Since the Raspberry Pi does not have PWM pins on its GPIO header it must use software defined pulse width modulation provided by the RPIO library on its GPIO pins to control the motors of the quadcopter. In order to use this method to control the quadcopter it has to be recalibrated to work at different PWM values as compared to using a physical controller.

RPIO is a C library that is used in conjunction with the Python scripting language to provide software defined PWM. Using simple commands, a simulated PWM signal is sent out on a GPIO pin to the flight controller which the flight controller then sends to the necessary motor to allow for navigation during flight. Based on the objects around the quadcopter the Raspberry Pi will call different Python methods to use PWM to move the quadcopter in predetermined direction to avoid the objects.

## Navigation

Navigating the rooms by following the right wall was the simplest and most effective way we came up with to navigate the two rooms. The quadcopter navigates facing the direction of the second room at all times so that the quadcopter only has to focus on going forward. In the case where an object should be found in its way in the first room, the quadcopter will navigate around it since all boxes are known to be at least a distance of five feet apart, which is plenty of room for the quadcopter to navigate through. When the quadcopter reaches a corner the flight direction is changed to continue following a wall. Navigation is stopped when the quadcopter pics up a wall in front of it and to its left which initiates the end condition of the navigation part of the algorithm. It is after reaching this ending condition for navigation that the quadcopter will know that it is in the room containing the ball and the ALVAR markers and will then begin searching for the ball after first rotating 180 degrees.

## Object Avoidance Components

Since the quadcopter that is being used did not come with any components for object avoidance we introduced sonar modules as a solution to object detection. In the case of the object avoidance it was decided that we would use the sonar modules to detect when the quadcopter is getting to close to a wall or box. Initially concerns were raised that the downward draft caused by the propellers my cause interference with the sonar modules so tests were performed to verify this. After performing these tests it was determined that the downward draft did not cause interference with the operation of the sonar modules and since they are cheaper and still quite efficient up to about five feet, they would be a more feasible option over the laser range finder. In order for the laser range finder to function we would need to mount it to a servo or motor on the bottom of the quadcopter that would rotate when the quadcopter stops moving and record values to see how close the quadcopter is getting to a wall or box. This would require the Raspberry Pi to perform more difficult calculations, therefore slowing down the progression of the quadcopter through the room. Sonar modules do not draw as much power as the laser range finder, improving battery life, and only need to notify the Raspberry Pi when it gets too close to an object on one or two sides, simplifying the computational load. With these details in mind, we added four sonar modules to the quadcopter; one on the front, right, left, and back sides.

Unfortunately the Raspberry Pi does not have enough pins to control all of the sonar modules at once so an Arduino Mega has been added to the quadcopter to control the modules and send a signal to the Raspberry Pi when it needs to avoid an object. An Arduino Mega has more pins for PWM and serial communication as compared to the Raspberry Pi which makes it much more suitable for controlling the modules. When one of the modules senses that an object is within a certain threshold then the Arduino will send a signal to the Raspberry Pi to run a method that will make the quadcopter move away from the object. This process will be running as the quadcopter travels along one of the side walls to move towards the back end of the room to find the entrance to the next room.

## Object Detection and Image Processing

Target identification is executed on the ground station, and also is a simple approach. Using the OpenCV C++ library, the image sent from the quadcopter is split from one three-channel image to three one-channel images, so that the program can manipulate the red, green, and blue channels independently. Since the target is green, the program’s objective is to filter out all color but pure green. To accomplish this, the program averages the red and blue channels and subtract the result from the green channel. This effectively removes all white and near white pixels and dampens blue and red pixels. The program then uses OpenCV’s Hough Circles feature to locate a circle. If a circle is found, we assume that this is the target, and the vehicle begins to search the room for ALVAR markers.

ALVAR Detection

Once the ball is found, the quadcopter directs itself to the ball and hovers over the ball and begins searching for the ALVAR markers. Once an ALVAR is located, the vehicle will measure the distance to the ALVAR using the laser rangefinder. Finding a second ALVAR and distance, the algorithm uses the coordinates of the ALVARS as well as a system of distance formulas to find the ball’s coordinates within the room as shown:

of ALVAR 1

of ALVAR 2

of the quadcopter/ball

## Safety

In order to safely disarm the quadcopter in the event of a failure a disarm mechanism has been added to the device. To stop the device in flight a kill switch was set up on the flight controller that when switched will kill all communication to the motors, bringing the quadcopter to a halt. For this to work the receiver for the controller was left on the quadcopter and the pin for the kill switch was left plugged into the flight controller. The flight controller was then programmed to stop PWM signals to the motors in the event that that switch was turned on. In order to be able to compete in the competition this was a required feature of our quadcopter.

## Ground Station

Communication with the quadcopter is predominately done through SSH protocol over a private and secure Wi-Fi network. Initially, the takeoff script is run from the ground station to begin the flight sequence. After takeoff, the quadcopter will operate on its own. In the event of a catastrophic failure, the operator of the ground station can use the kill switch on the controller to stop the quadcopter and kill all input to the motors. When the ball has been found the Raspberry Pi will use the SCP protocol to send the image of the ball to the ground station for further processing.

Once the image of the ball has been received, the ground station will do further processing to isolate the ball in the image received. Commands are sent to the quadcopter over SSH to position the quadcopter over the ball and begin searching for the ALVAR markers. Once again, images will be sent to the ground station in order to detect the ALVAR markers. For each marker found, the quadcopter sends the coordinates corresponding to each marker and the distance measured.

# Discussion

When used responsibly, the drone can be an important tool for military recon squads. Consider the case of a bomb threat. It may be more beneficial to send a drone into the building to search for it rather than a human, considering the time for it to be set off is unknown. This would save teams the risk of sending an actual human to potentially lose his or her life. The same applies to any hostage situation or even a burning building. It also has an advantage over a ground unit, as the ground unit can easily have its passage blocked, whereas the drone can have several alternate routes that will less likely to be obstructed. Where the drone has a disadvantage is the fact that it is a less stable machine; if an unforeseen force hits the drone, there is a high risk of a crash. This is why the drone would best be used in an indoors environment where a gust of wind would most likely not arise. If used irresponsibly, this technology may be used for invasion of privacy, a topic that has been very controversial as of late. It is important that this is used by an organization that can be trusted, such as the military, for putting this technology in consumer hands could create many issues if used for something other than what is intended.

# Conclusion

The recon drone poses a challenging problem that encompasses stability control, location tracking, object avoidance, image processing, and search algorithms. The key to its implementation is integrating these ideas in such a way that it can operate efficiently and accurately without too much equipment. The drone is supposed to be a light, cheap solution to a recon unit, so strapping a heavy processing unit to control the drone is a less elegant solution. It is key for us to ask ourselves “what is the simplest way to achieve the desired result?” and after that is implement, strive for a more adaptable solution.

# Future Work

Most of our future work consists of integrating each modules' implementation into one working machine. Although we have working object detecting and distance measuring modules, we have not used both at the same time. It’s possible that once integrated, the solution may not be efficient enough, or one module may conflict with another, requiring a redesign of certain modules. The biggest challenge ahead of us is the image processing, as we have only done minimal exploration into the concept. Another possibly challenging and less controllable challenge is the set up of a communication protocol between the ground station and the drone. There are several ways to achieve this, whether it’s through Wi-Fi, Bluetooth, etc., but there will more than likely will be only one efficient way to implement this module. From a more conceptual standpoint, it is possible that our search algorithm may prove to be ineffective, so it is important that we test it thoroughly and get complete coverage for the several possible scenarios that can occur in the competition. After our experience with the Ardupilot, we have determined that it is possible that better flight controllers may have been released, such as the Erle-Brain flight controller. It runs Linux on a BeagleBone Black based microcontroller, allowing for a more configurable and powerful flight controller. The Raspberry Pi could also be upgraded to the latest model in order to take advantage of parallel processing using a quad core processor and an increased number of GPIO pins.