AC 2012-3704: DESIGNING AN AUTONOMOUSLY NAVIGATING MODEL BUGGY

Mr. Richard James Choquette, University of Maryland, Eastern Shore

Richard James Choquette currently works for Computer Sciences Corporation, serving as a Flight Safety Analyst in the Range Safety Branch at NASA's Wallops Flight Facility in Virginia. His work involves supporting the analysis of the various risks involved with the flight operations of sounding rockets, UAVs, balloons, and expendable launch vehicles. Choquette has received an undergraduate degree in engineering with an aerospace specialization from the University of Maryland Eastern Shore (UMES) in May 2011. In addition, he spends most of his free time serving as a volunteer firefighter in Pocomoke City, Md.

Dr. Payam Matin, University of Maryland, Eastern Shore

Payam Matin is currently an Assistant Professor in the Department of Engineering and Aviation Sciences at the University of Maryland, Eastern Shore (UMES). Matin has received his Ph.D. in mechanical engineering from Oakland University, Rochester, Mich., in May 2005. He has taught a number of courses in the areas of mechanical engineering and aerospace at UMES. Matin's research has been mostly in the areas of computational mechanics and experimental mechanics. Matin has published more than 20 peer-reviewed journal and conference papers. Matin worked in Auto-industry for Chrysler Corporation from 2005 to 2007.

Dr. Ali Eydgahi, Eastern Michigan University

Ali Eydgahi started his career in higher education as a faculty member at the Rensselaer Polytechnic Institute in 1985. Since then, he has been with the State University of New York, University of Maryland, Eastern Shore, and Eastern Michigan University. During 2006-2010, he was Chair of the Department of Engineering and Aviation Sciences, Founder and Director of the Center for 3-D Visualization and Virtual Reality Applications, and Technical Director of the NASA funded MIST Space Vehicle Mission Planning Laboratory at the University of Maryland, Eastern Shore. In Aug. 2010, he joined Eastern Michigan University as an Associate Dean in the College of Technology and currently is a Professor in the School of Engineering Technology. Eydgahi has served as Secretary/Treasurer and Program Chair of the ECE Division of ASEE, as a regional and chapter Chairman of ASEE, SME, and IEEE, as an ASEE Campus Representative, as a Faculty Advisor for National Society of Black Engineers Chapter, as a Counselor for IEEE Student Branch, and as a session Chair and a member of scientific and international committees for many international conferences. Eydgahi has been an active reviewer for a number of IEEE and ASEE and other reputedly international journals and conferences. He has published more than 100 papers in refereed international and national journals and conference proceedings such as ASEE and IEEE.

Designing an Autonomously Navigating Model Buggy

Abstract

Senior students in the engineering and technology programs are challenged to thoroughly apply their learned technological knowledge and skills toward design and implementation of a challenging engineering product in senior design or capstone courses. In this paper, a successfully implemented comprehensive design of an autonomously navigating 1/10th scale model buggy by a senior engineering student under supervision of two advisers is presented. The project utilizes a synergy of competencies gained from undergraduate academic and research experiences with insight to the efforts concerning the senior design project. The main goal of this project was to design and implement an autonomous system with the ability to navigate while utilizing GPS, a digital compass, and infrared (IR) sensors for obstacle avoidance. The system is designed in such way that can easily be replicable with a low cost platform while utilizing open source software and hardware. A number of tests were conducted to validate the performance of the model buggy. The student has gained significant experience in the development of this autonomous control system while applying knowledge learned during the undergraduate program of study.

1. Introduction

With the price of most Unmanned Aerial Systems (UAS) and Unmanned Ground Vehicles (UGV) upwards of tens or even hundreds of thousands, they often become unobtainable for most and too costly for widespread usage. There are still legal hurdles when it comes to flying UAS commercially [1], but the primary roadblocks to UGVs are only cost and awareness. With pending legislative changes by the Federal Aviation Administration (FAA) to allow wider use of UAS in the National Aerospace System (NAS), now seems the time for more low-cost alternatives for both UAS and UGVs. This project seeks to take an open source control system developed for an R/C glider and apply it to an autonomous ground vehicle.

Technology has come to the point where such a task is not insurmountable. Starting in 2009, SparkFun Electronics has held an Autonomous Vehicle Competition (AVC) where anyone can create an autonomous vehicle (ground or aerial) with the goal of being the fastest to travel around their building [2]. Each year new, solutions are found and platforms are pushed further in search of winning the top prize. The DIYDrones project [3] was started on a similar note; to create an autonomous aerial vehicle capable of following a GPS course using low-cost hardware. The field of low-cost autonomous vehicles is rapidly expanding and the ideas of hobbyists are becoming on par with commercial technology for a fraction of the price.

As a senior capstone course, the senior design project is designed to allow students to prove their ability as an engineer. This is a project that demonstrates a student's ability to engage in the practice of engineering as a profession while following a standard design approach to a real-world problem. By defining a project relevant to aerospace within a competitive and emerging commercial market, this is a valuable experience which adds to the unmanned systems field as a whole while fulfilling the objectives of a senior design course.

This paper presents outcomes of a senior design project that is focused on design and implementation of an autonomous ground vehicle. This vehicle is able to navigate using a pre-planned GPS path, avoid any obstacle that may be along that path, maintain a reasonably low cost, and wirelessly transmit telemetry based on a control system that utilizes an open source solution. In addition, this project includes the design and fabrication of a sleek, carbon fiber composite body, which is used in construction of many modern unmanned vehicles. The development of this vehicle utilizes the knowledge gained from many aspects of the undergraduate engineering degree while adding more capability to the open source unmanned system community. The project was performed by an undergraduate engineering student who was mentored and supervised by two faculty members.

2. Design

The proposed platform is a $1/10^{th}$ scale R/C Buggy that is sold as the Team Associated RC10 B-3[4]as shown in Figure 1. The motor, electronic speed controller (ESC), and steering are all commercial off the shelf (COTS) and are easily replaced. Parts are designed to be serviceable and easily replaced or upgraded so design modifications are possible without substantially modifying the platform.

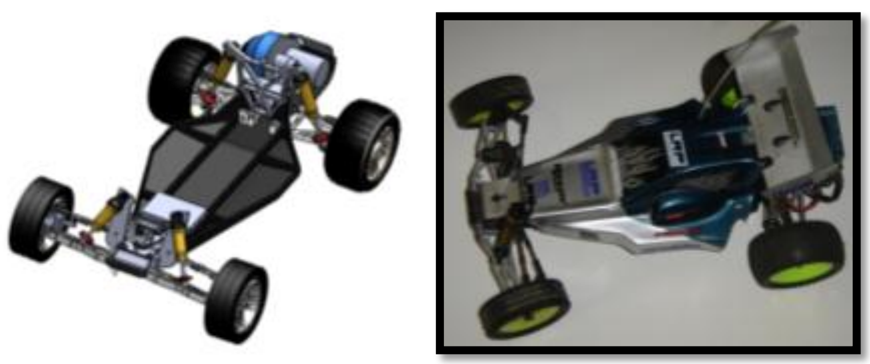

Figure 1: Team Associated RC10 B-3

Space comes at a premium and it is a design challenge to fit an entire autonomous control system on this platform. As opposed to a purpose designed chassis, custom sensor mounts have to be designed. In order to provide more durability, the plastic body is replaced with one fabricated out of a carbon fiber composite. This platform has the added benefit of a three-dimensional computer aided design (CAD) model that was created by the engineering department’s sophomore level CAD class as their final project. As remaining space is decreased, component placement is optimized by utilizing the CAD model.

Several constraints were considered in the design of the autonomous buggy. The following constraining factors guided how the overall system is designed:

- Utilize low-cost components. The overall cost of the project should be kept at or below a cost of $500, or be reproducible at that level. This is due to limited available funding and a goal of designing a UGV that others can reproduce easily.
- Utilize an open source UAV controller for its low cost and ease of modification. This controller must be modified so that it is able to navigate a ground vehicle by GPS, steer the wheels, and control the throttle.
- The vehicle platform must be large enough to carry all of the control components including sensors for obstacle avoidance. As such, extra interior space as well as external mounting points must be considered when selecting the vehicle.
- Use only one battery in order to reduce the system complexity and sources of failure. The system must be able to run off of a single battery that may vary in voltage depending on the platform used. A range of voltages may have to be accepted by the system that must be converted to power the components.
- Keep the system simple to reduce the circuit complexity. If selected components are able to accept the same voltage, the less power regulation and conversion has to be designed. Electrical components such as sensors and the controller should run on the same voltage to meet this constraint.
- Create an obstacle avoidance system. It should be able to give the vehicle ample time to respond and avoid with a range of five feet or greater. It cannot consume an excessive amount of power as this would decrease the performance of the vehicle, so a compact solution is desired. It also must be affordable and should not cost more than $100 to keep the overall system cost down. This system must fit on the selected platform.

A custom built chassis is considered as it is designed for the control system and has a specifically designed place for every component of the system. This allows for a custom control solution that matches the platform as designed. Sensor mounting is built in and the platform has space for every component.

A typical store bought platform was also considered as it may be very inexpensive for the size, coming in at less than $50. There is no platform design time needed and a very wide range of sizes are available. Even though motors and steering components are there, motor controllers need to be purchased or designed as most store bought R/C cars do not carry any kind of incremental control. As opposed to a custom chassis, custom sensor mounts needs to be designed and sensors may not have ideal placement. Most store bought R/C cars are also weakly constructed and not very durable. For all these reasons, this option is unsuitable for this project.

2.1 Parts Selection

For the motor and speed control, the Traxxas Velineon System was selected[5]. It includes a waterproof brushless motor and electronic speed control. It provides the capability for very high speeds as the control system matures. A slower motor could be selected, but it could potentially become a limiting factor. The durability of a waterproof system is also desirable.

The steering servo is not replaced as it functions properly and provides accurate control. If it were to be replaced, a servo with greater reliability and durability, including a waterproof housing could be selected. This expands the capability of the buggy and the kinds of missions it can withstand.

The buggy is powered by NiMH batteries[6]. They provide a higher energy density than NiCad batteries and have a longer usable lifetime than NiCads. NiCads are prone to a decreased life after multiple recharge cycles. The only more desirable batteries in terms of energy density are LiPo batteries, though they come at a significantly higher cost and require a specialized and more expensive charger.

The ArduPilot microcontroller board[7] was selected for multiple reasons. This board is designed and developed by hobbyists who sought a way to stabilize an R/C plane and allow it to navigate by GPS waypoints. It is a small and compact version of the Arduino microcontroller, which is very simple to program. It was designed as open source so the software and hardware are publicly documented and encouraged to be modified and improved. It has a built-in GPS connector and is designed for PWM connections typical of hobbyist R/C hardware. It has a built in dedicated R/C failsafe for manual override using an R/C radio that is run by a separate processor. Finally, it runs at five volts which is a common electronic voltage that is compatible with many other components.

The GPS unit selected is the USGlobalSat EM406A GPS unit[8]. The ArduPilot is designed to work with the EM406A. It is compatible with the connector on board and uses the standard NMEA protocol that is easy to integrate into any autopilot code.

The Honeywell HMC6353 digital compass[9] is selected to provide an accurate heading superior to that of the one provided by GPS. As a compass, it is selected for its low cost and the fact that it runs on a compatible five volts. It is only a two axis compass and does not perform ideally when tilted. However, for the cost and compatibility with the system, it is the best solution. At small geographic distances, the GPS heading is unreliable as it cannot sense the minute changes in direction. The compass provides an accurate heading at every moment.

For the obstacle avoidance system, infrared distance sensors are selected. The Sharp GP2Y0A710YK IR sensors[10] have the longest range available at a low cost as they can detect objects up to 18ft away. They come at a low cost at only $19.50 per sensor, and the next smallest range of infrared sensors is a sensor with a range of 5ft, which costs $14.95 a piece.

With a 1Hz control loop, speeds approaching 18fps (12.2mph) are possible with the 18ft sensors. In addition, the sensors run at 5v and as such are compatible with all of the other selected components. As these are the best IR sensors on the market, the next best option would be a laser scanner costing over $1,000.

For radio control, the Traxxas Link 2.4GHz Transmitter/Receiver was selected[11]. It includes one of the only four-channel pistol style transmitters on the market. One additional switch is used for switching between manual and autopilot modes, while the second switch is available to control a future expansion to the system such as a camera shutter or for sensor deployment.

For wireless telemetry transmission, the Digi XBee Pro 900 was selected[12]. XBees comes in only two frequency bands, 2.4GHz and 900MHz. The 900MHz module was chosen for greater range and less chance of interference with the 2.4GHz radio control system. The Pro 900 has up to a 1.8mi (3 km) range, and up to 6mi with a high-gain antenna. In addition, XBees are plug and play; no software needs to be written to begin transmitting a serial stream.

The LM7805[13] can accept a range of 7v to 25v nominal, and up to 35v maximum with a reliable output 5v. Typical hobbyist battery packs have outputs ranging from 7.2v to 9.6v while the microcontroller and other components run at 5v. The LM7805 is widely used and capable of handling the power regulation for all of the components.

2.2 Design Process

The design process is illustrated in Figure 2. The first system that needs to be completed is the control system so that the buggy is able to navigate. Following that, obstacle avoidance is integrated and tested on a planned mission. Finally, the carbon fiber body is fabricated for the completed vehicle. It should be noted that the student has been encouraged by the faculty advisors to develop the design process from the early stage of the project. The design process developed by the student has been evolved and modified during the course of the project as a result of a close interaction between the student and advisors.

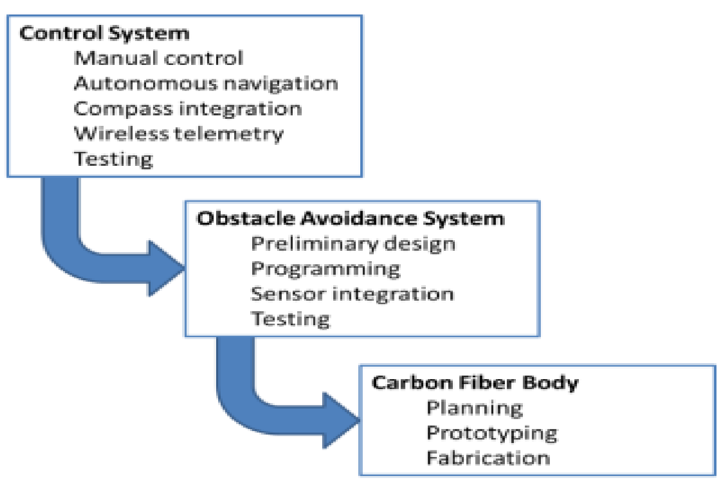

Figure 2: Work Sequence

2.3 Implementation

The ArduPilot is placed in the chassis frame, and the voltage regulation circuit board is designed to fit in the remaining space on that side. The receiver is placed at the location of the antenna mount, and the ESC is mounted against the rear shock tower.

The compass has been placed on a square dowel above the vehicle so that it is away from any electromagnetic interference generated by the car. The GPS is mounted on top of the body to give it a clear view of the sky to pick up satellite signals. The XBee is mounted on top of the motor as it is the only remaining place it would fit with the antenna attached.

Lastly, a mount has been designed on the front of the vehicle for the three IR sensors to sense obstacles. A flat platform was first designed, and then a mount attached to the vehicle body to place that platform above the front shock tower was designed. A full solid model was built for this design. The component placement diageam is presented in Figure 3.

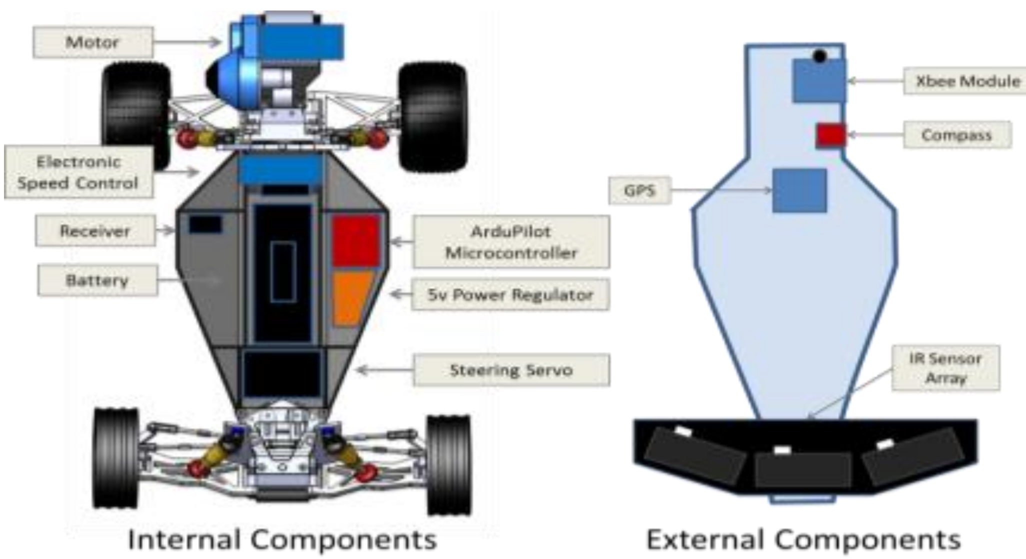

Figure 3: Component Placement Diagram

Figure 4 provides an overview of the wiring layout of the entire system. The compass utilizes analog lines 4 and 5 for its I2C communications, while the three IR sensors use lines 0, 1, and 2. Analog line 3 is available for an additional sensor. The GPS sends data over the RX line of the serial connection, while the XBee transmits data over the TX line. Two-way serial communication over the XBee is not possible due to the GPS using the only available RX line. There are two independent sets of five volt outputs on the regulator board, with one being dedicated for use by the IR sensors so that they are unaffected by voltage drops created by the rest of the system.

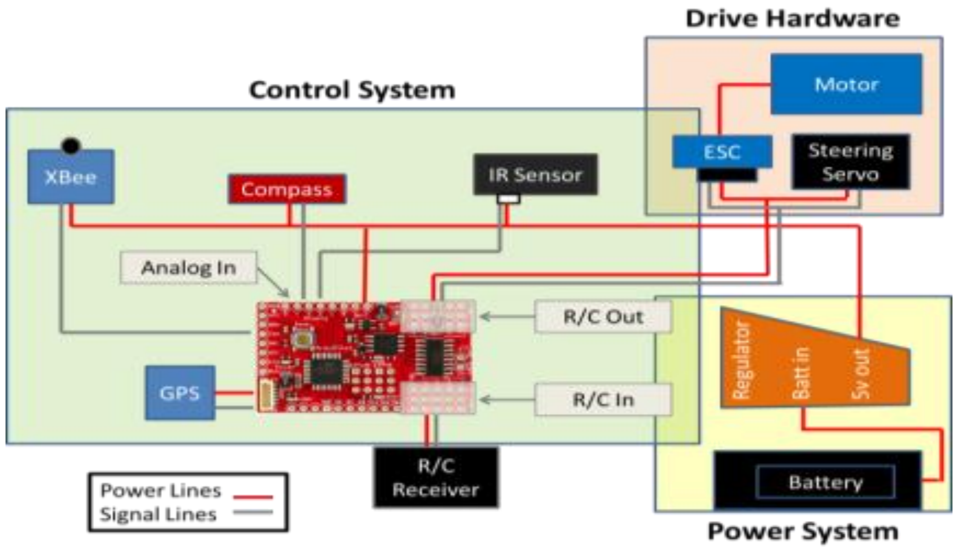

Figure 4: Wiring Diagram

A voltage regulation circuit is designed around the LM7805[13]. In Figure 5, the suggested fixed-output regulator from the datasheet is pictured. Two identical circuits are built based on this design that both provide five volts of power for the system. The circuit is soldered to a piece of printed circuit board cut to fit in the remaining space beside the microcontroller as depicted in figure 3.

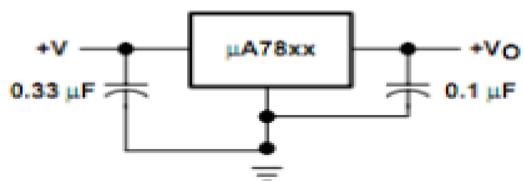

Figure 5: Voltage Regulation Circuit

A compass is selected in order to provide an accurate heading at any moment, including when the car is stationary. At low speeds this is superior to a GPS heading, which is calculated based on a changing position. The compass is placed on a square dowel approximately six inches above the car to be away from electromagnetic interference induced by current carrying wires and other components such as the electric motor. By being placed at the top of the dowel, the compass is visually verified for levelness prior to the start of any test.

The selected compass provides a heading with a tenth of a degree resolution using the I2C protocol. I2C works in a method similar to serial communications, only with addressable devices on an I2C bus that communicate over two analog lines. In the Arduino IDE[14], I2C communications are accomplished using the Wire protocol. This protocol facilitates communications and is used to initiate the data link and retrieves data from the compass. A function was created in order to handle communications with the compass and return a heading value to the control loop.

2.3.1 Obstacle Avoidance System

Multiple configurations are considered for sensor placement. The initial testing of the system considered obstacles at 10ft out as the vehicle was traveling at about a walking pace. The selection of three sensors was primarily based on the basic idea of a line-following algorithm. Instead of following a line, the buggy is looking for a clear path. Considering the buggy has a maximum turning radius of 30°, greater angles would be beyond the limit of the system. By reducing that angle to 20° and calculating the tangent of 20°, an avoidance of only 3.63ft is determined. This allows for a full three feet of avoidance at a distance of 10ft, and 1.8ft avoidance at 5ft away. This will be acceptable at avoiding small obstacles without a massive error in the course after only one correction while larger obstacles continue to be avoided on subsequent passes of the control loop.

The obstacle avoidance system is designed for the selected Sharp IR sensors[10]. First, the sensors were placed in their desired configuration in SolidWorks. Modeling was simplified as the vendor provided a solid model of the sensors. With the sensors in place, a platform was designed around their placement as seen in Figure 6. Once the platform is designed, an interface to mount the platform to the body of the car was necessary. It was designed to be tall enough to place the platform above the shock tower, as it is undesirable to have the sensors in front of the car in case of an accidental collision as the sensors would be damaged.

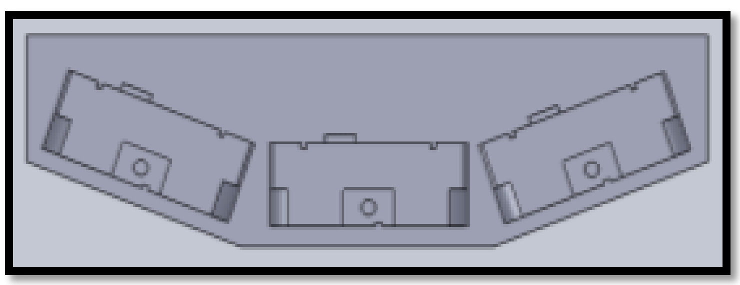

Figure 6: IR Sensor Platform

In order to program for the IR sensors, their output must be known. A linear chart of the voltage against the inverse of the distance is given in the datasheet. A few points were extrapolated from that line in order to get a line of best fit using Microsoft ExcelThis line of best fit is used in the program to get a useful distance reading based on the voltage sent to the microcontroller. This is used to write a function that reads the three sensors and decides what corrective evasive action to send to the control loop. The source code for the obstacle avoidance system is implemented in the Obstacle_Avoidance Tab.

As shown in Figure 7, a simple algorithm was developed for the obstacle avoidance system. The function constantly checks the forward sensor for an obstacle. If there is no obstacle, no correction is necessary. If there is an obstacle, it checks both the left and right sensors for their readings. If there are no obstacles, the vehicle defaults to the left side. If there are obstacles in front of all sensors, the vehicle chooses to travel towards the obstacle that is the farthest away, with the assumption that there is enough time to correct away from that obstacle on the next iteration of the control loop. Effective path planning avoids any major obstacles that could confuse this algorithm, while most unexpected obstacles such as a parked vehicle or pedestrian are avoided. It should be noted that no particular assumptions have been made about the size of the obstacle; the obstacle avoidance system only detects whether or not an object is there to reflect the infrared beam.

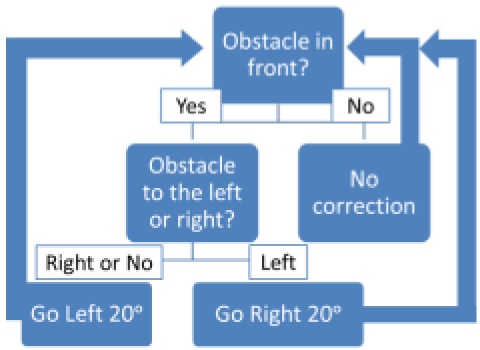

Figure 7: Obstacle Avoidance Program

For making the carbon fiber body, a test layup was done with inexpensive fiberglass in order to see how a similar fabric would lay out on the heavily contoured mold. The carbon fiber body was fabricated by utilizing a vacuum bagging process and a mold of the body. The vacuum applies equal pressure across the composite layup in order to allow it to conform to the molded shape. Three layers are used, two layers of carbon fiber with a third layer of fiberglass in between to add additional thickness for more rigidity. The final product is very durable and allows sensors to be mounted without fear of cracking or harming the body.

The completed system is pictured in Figure 8 as it was designed and prototyped.

Figure 8: Completed System

3 Testing

Path planning is conducted by utilizing Google Earth. Waypoints can be planned by using the "Add Placemark" feature from which latitude and longitude can be determined. When choosing waypoints, a 5m radius is considered and obvious obstacles are avoided. The obstacle avoidance system is primarily for unexpected obstacles as the chance of hitting obstacles is ideally reduced by accurately positioning waypoints during path planning.

The waypoint missions are uploaded as a part of the overall source code. They exist in the primary function tab of the ArduPilot program. Waypoints are entered as latitude and longitude points.

The results from two separate missions are presented in Figures 9 and 10. The first mission pictured in Figure 9 has been completed around a parking lot. The corner vertices represent the waypoints. The dots are the actual GPS coordinates that the buggy passed through. The dimensions of this image are approximately 60m by 40m. Figure 10 shows the second mission which is associated with the mission from the above steps. The dimensions of this image are approximately 20m by 45m. Using physical landmarks, it is determined that the car does not deviate more than approximately 3m from the course. The accuracy demonstrated was acceptable considering the low-cost equipment used. The GPS appeared to perform much better than what the manufacturer specified.

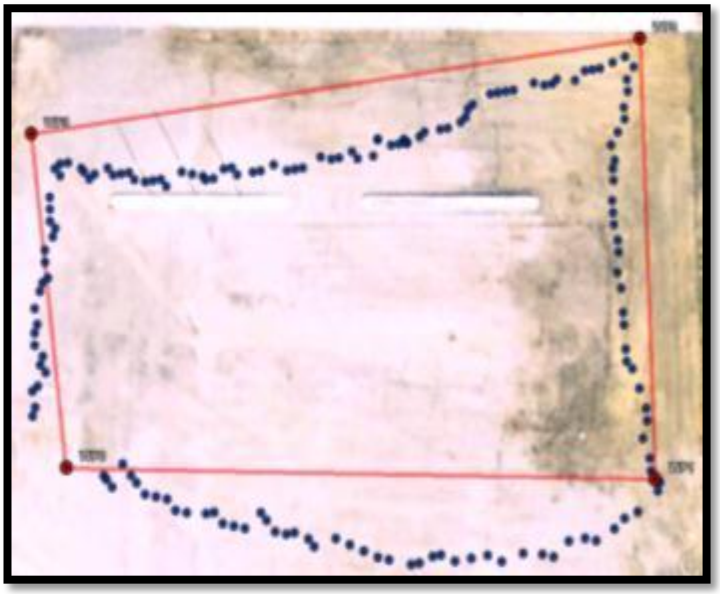

Figure 9: April 19th Run #4

Figure 10: May 11th Run #3

The steering is controlled by a PID controller. At the beginning of tuning the steering, the gains were 2.5, .01, and .15, respectively. The response is shown in Figure 11. The most obvious and first problem to be addressed is an overreacting proportional response, while the derivative gain needs to be tuned to effect the rate of change of the error.

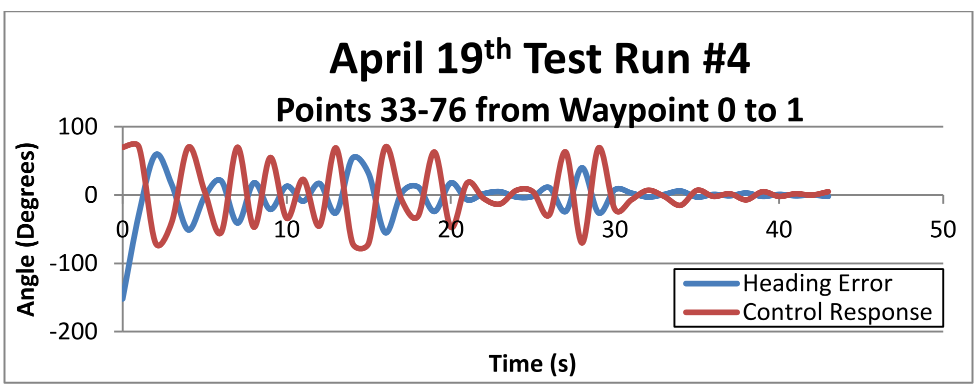

Figure 11: April 19th Run #4 Control System Data

In order to tune the control system, the proportional term needs to be reduced so that the car is not over-correcting to the error. The proportional gain controls a proportional response to the error. The derivative term helps control the rate of change of the response. It keeps sudden changes from happening too fast, slowing the rate of change. Once the proportional gain is turned down, the derivative gain is adjusted to help keep sudden changes in check. The integral term helps with the steady state error and may be adjusted slightly but is not the focus of getting this control system in order.

After multiple tests, the gains are tuned to be 1.3, .03, and .19, respectively. This produces a settling time of around five seconds and kept the error relatively low as shown in figure 12.

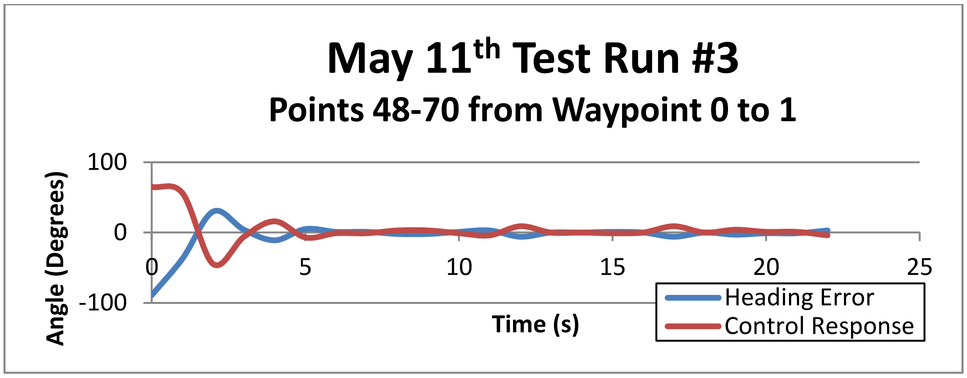

Figure 12: May 11th Run #3 Control System Data

The entire set of data behind these graphs is available for tuning the control system more precisely. Data such as the perceived GPS heading, GPS speed, distance to the next waypoint, and much more is transmitted by the system.

Although most paths are planned to avoid obstacles, errors in planning or unexpected objects can cause objects to enter the buggy's path. In some testing, there was a curb in the parking lot that was not in the planned path. Due to error in the navigation system, the buggy was traveling in the direction of this curb. Once the curb entered the path of the buggy, the buggy corrected and turned away from it. After a few seconds of travel, the buggy adjusted its path towards the waypoint. However because of the size of the curb, it was once again in the path. Again, the buggy successfully avoided a collision with the curb. Figure 13 shows the abnormalities in the path indicating the buggy correcting away from the curb. Red arrows indicate the relative direction of the correction while the concrete curb being avoided is highlighted in orange.

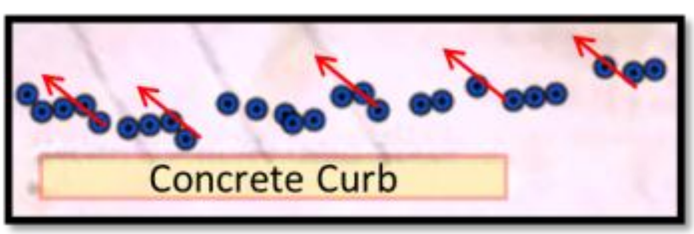

Figure 13: Path Corrections

3.1 Performance Analysis

The system met design constraints while utilizing low-cost components. The entire system can be reproduced for around $500. This project successfully modified an open source UAV controller for use on the ground. While the navigation code is not substantially modified, the obstacle avoidance and compass functions are major modifications to the original code.

The vehicle battery powers the entire vehicle and systems as desired while keeping the overall system simple. Drive components run directly from battery through the ESC while autonomous control components run from the 5v regulator which feeds from the battery. The selected platform is sufficient in size for all systems and still has room for more external systems.

The vehicle is capable of avoiding obstacles as far away as 18ft without significantly reducing the life of the battery. The obstacle avoidance components are low cost and come in well below the $100 target.

The navigation system consistently hits waypoints well below the 5m tolerance for success. The vehicle travels at a constant rate of approximately 1.5 ft/s with room for improvement and is able to settle on course to its next waypoint in around five seconds.

4 Cost Analysis

One of the primary objectives of this project was to deliver a low-cost solution. In every aspect, the authors have sought to provide the greatest capability and simplicity per component at the lowest price point. The complete tally is shown in Table 1.

Table 1: Cost Breakdown of the Project

Product Name	Vendor	Price	Qty	Subtotal
ArduPilot - Arduino UAV Controller	SparkFun Electronics	$24.95	1	$24.95
Traxxas Velineon Waterproof Brushless System	Tower Hobbies	$149.99	1	$149.99
20 Channel EM-406A SiRF III Receiver	SparkFun Electronics	$59.95	1	$59.95
FTDI Cable for Controller Programming	DIYDrones	$19.90	1	$19.90
Misc. Electronic Components	SparkFun Electronics	$30.00	1	$30.00
Sharp GP2Y0A710YK0F IR Sensor Package	Acroname Robotics	$19.90	3	$59.70
HMC6353 Compass Module	SparkFun Electronics	$34.95	1	$34.95
Wireless Telemetry Components	SparkFun Electronics	$140.70	1	$140.70
Misc. Carbon Fiber Layup Supplies	(ESR Donation)	$0.00	1	$0.00
15% Built-in Cost Buffer (Shipping, etc.)				$78.02
			Total	$598.16

The components needed for this project are estimated to fall in just below $600 for a total of $598.16 which includes a built in 15% cost buffer for extra costs such as shipping. This makes it a very affordable project that provides a working product with great capabilities. The use of COTS components allows the project to be duplicated relatively easily.

The donated carbon fiber supplies are estimated to cost approximately $140 on the market, and a comparable R/C buggy costs approximately $120 with an included two channel radio system. This brings the total cost to approximately $700, though the purchase of an additional motor and ESC (listed at $150 in Table 1) are avoided with certain workarounds.

Used buggies can be purchased online for approximately $50; In addition, a two channel radio can be used, at the loss of remote manual override. An on-board toggle switch or button can be added for safety if desired.

Carbon fiber provides durability, but the stock body could be used. The cost of carbon fiber was not an issue in this project as it was donated by a private company. A custom body could be fabricated out of a material such as aluminum or some other material at a reduced cost while offering improved durability as well.

Wireless telemetry is not necessary and could be eliminated to reduce cost; an on-board data logger can be integrated very easily for around $20. Wireless telemetry is very useful during debugging to provide live data on the status of the car, and could be useful while taking sensor readings if desired.

5. Student Learning Outcomes

The project was clearly defined in a few project kick-off meetings through brainstorming between the student and faculty advisors. The faculty advisors monitored the progress of the project through regular weekly meetings. The student presented his weekly accomplishments and challenges in a form of PowerPoint presentations to the senior design class audience. The advisors provided helpful comments during the presentations. The presentation always included the student's plan for the coming week, and the percent toward completion of the project. The advisors required written reports at different milestones. After the project was completed, the student presented the final presentation, and submitted the final written report. Before the written report was finalized, it was reviewed by one of the faculty. The report was progressively modified after three revisions before final submittal.

The project was conducted in the form of Senior Design Project I and II offered in two consecutive semesters. The senior design courses are 2-credit hour courses being offered in 16-week semesters. The faculty advisors mainly spend two hours a week in classroom reviewing designs and advising students. The advisors are available during their office hours if needed.

The followings provide the student learning outcomes of this project:

- The project exposed a student to design process of a real world problem with a low-cost approach.

- The student developed a logical 3-phase design approach to designing a functional autonomous buggy.
- The student gained hands-on experience working with computer aided design software, programming environments, mapping software, and control systems.
- The student improved his technical oral presentation skills as a result of regular weekly presentations.
- The students improved his technical writing skills as a result of working progressively on written reports.

6. Conclusions

Through this project, valuable experience was gained in the field of unmanned control systems. The process of designing the system and integrating multiple subsystems has been a great learning experience. The buggy performs as it is designed and can successfully navigate between waypoints with the ability to avoid obstacles. Overall, the cost was kept down to nearly $500 and a similar system with some concessions can be built for well under $500. The purpose of gaining experience across a wide area of engineering was accomplished over the course of this project.

References

1. “UAV FAA Regulatory Information." UAV MarketSpace. Web. 17 May 2011. http://www.uavm.com/uavregulatory.html
2. Seidle, Nathan. "Autonomous Vehicle Competition and a Bunch of New Products." SparkFun.com. SparkFun Electronics, 23 Dec. 2008. Web. 17 May 2011. http://www.sparkfun.com/news/215
3. Anderson, Chris. DIYDrones.com. DIYDrones. Web. 17 May 2011. http://www.diydrones.com
4. “Team Associated RC10 B3.” Team Associated. Web. 16 March 2012. http://www.teamassociated.com/cars_and_trucks/RC10B3/Team/
5. “Traxxas Velineon VXL-3s Electronic Speed Control” Traxxas. Web. 16 March 2012. http://traxxas.com/products/parts/escs/vxl3s
6. “Onyx NiMH Batteries.” Tower Hobbies. Web. 16 March 2012. http://www3.towerhobbies.com/cgi-bin/wti0001p?&I=LXXUN9
7. “ArduPilot - Arduino Compatible UAV Controller.” SparkFun Electronics. Web. 16 March 2012. http://www.sparkfun.com/products/8785
8. “20 Channel EM-406A SiRF III Receiver with Antenna.” SparkFun Electronics. Web. 16 March 2012. http://www.sparkfun.com/products/465
9. “Compass Module - HMC6352.” SparkFun Electronics. Web. 16 March 2012. http://www.sparkfun.com/products/7915

10. Sharp Corporation. "GP2Y0A710K0F Distance Measuring Sensor." Sharp Microelectronics of the Americas. 23 Mar. 2007. Web. 19 Apr. 2011. http://www.sharpsma.com/optoelectronics/sensors/distance-measuring-sensors/GP2Y0A710K0F
11. "TQ 2.4GHz High Output Traxxas Link." Traxxas. Web. 16 March 2012. http://traxxas.com/products/parts/transmitters/2238tq24ghztraxxaslink2ch
12. "XBee-PRO® 900." Digi International Inc. Web. 16 March 2012. http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/point-multipoint-rfmodules/xbee-pro-900#overview
13. Texas Instruments. "µA7800 SERIES POSITIVE-VOLTAGE REGULATORS" Texas Instruments Incorporated. May 2003. Web. 12 May 2011. http://www.sparkfun.com/datasheets/Components/LM7805.pdf
14. "Arduino Development Environment." Arduino. Web. 16 March 2012. http://arduino.cc/en/Guide/Environment