Autonomous Ground Vehicle

A Project Report
Submitted to the Faculty
of the
UNITED STATES COAST GUARD ACADEMY
in partial fulfillment of the requirements for the
Degree of Bachelor of Science
in
Electrical Engineering
by

_____
1/c Brian Seekatz

_____
1/c Jeff Matejka

_____
1/c Adam Reckley

Projects in Electrical and Computer Engineering

25 April 2008

Reviewed By:

_____
CAPT Richard Hartnett, Project Advisor

_____
LCDR Daniel K. Pickles, Class Coordinator

_____
Professor Keith C. Gross, Chief, Electrical & Computer Engineering Section

Abstract

This paper documents the process used for the concept, design, and implementation of the United States Coast Guard Academy's Autonomous Ground Vehicle (AGV), developed as an electrical engineering senior design project sponsored by C2CEN and NAVCEN. The vehicle is to compete in the annual Intelligent Ground Vehicle Competition (IGVC), held in Michigan in early June, a contest where students from many universities build robotic vehicles and test them in a number of obstacle courses. The project has numerous applications in all walks of life; however, autonomous navigation has large potential in the Coast Guard, because of the heavy emphasis placed on navigation for sea-going duties. The work being done during the 2007-2008 academic year is a continuation of the work done during the 2006-2007 academic year. The goals for the vehicle and the design solution to accomplish those goals are detailed in the paper.

Table of Contents

Introduction page 1

Background page 2

Objective page 5

System Design page 7

- Vehicle Frame page 7
- Motion Control page 9
- Sensor Array page 10
  - GPS page 10
  - SICK Laser page 11
  - Digital Camera page 12
  - Digital Compass page 18
  - Shaft Encoder page 18
- Decision Making Algorithm (DMA) page 19
  - System Integration page 19
  - Waypoint Navigation page 22
  - Obstacle Avoidance page 22
- Design Summary page 24

Results page 24

- Lane Detection Results page 24
- System Integration Results page 25
- Results Summary page 28

Conclusions page 30

References page 31

Bibliography page 31

Appendices page 32

Appendix A: Business Case page A-1

Appendix B: Project Management Plan (fall) page B-1

Appendix C: Project Management Plan (spring) page C-1

Appendix D: Functional Requirements Document page D-1

Appendix E: Support Plan page E-1

Appendix F: Design Specifications page F-1

Appendix G: Test Plan page G-1

Appendix H: Commented DMA MATLAB code page H-1

Introduction

As navigation technology advances in the 21st century, engineering projects, thought impossible 50 years ago, are becoming realities; one of which being the ability to navigate autonomously, or without input from a human user. The United States Coast Guard Academy's (USCGA), Electrical and Computer Engineering Section, sponsored by the Coast Guard Command and Control Engineering Center (C2CEN) and Navigation Center (NAVCEN), has endeavored to explore and develop this concept by designing an Autonomous Ground Vehicle (AGV) to compete in the annual Intelligent Ground Vehicle Competition (IGVC), held in Michigan in early June. This project is a prudent investment of the Academy's time and money because of its heavy emphasis on navigation, a skill that is absolutely essential to the successful completion of the Coast Guard's mission on the seas and in the skies. Furthermore, if the autonomous navigation technology can be used to help implement Coast Guard Unmanned Aerial Vehicles (UAV's), the high risk involved with dangerous search and rescue missions and other hazardous Coast Guard objectives can be minimized. An AGV capable of performing all required tasks of the IGVC will be created by combining the efforts of the AGV design team who began the project during the 2006-2007 academic year, three CGA undergraduate electrical engineering students and the graduate works of LT Everette and LT Taylor, URI.

The work required for the project was split up into two main phases that correlated, conveniently, to the two semesters of the academic year: Phase I (fall) efforts included modifying the 2006-07 AGV to better meet the IGVC requirements by improving the frame design and propulsion system. Phase II (spring) efforts included the sensory integration and decision making algorithm aspect of the design solution. At the conclusion of the academic year

the AGV is a complete unit, integrating all necessary components to successful participation at the IGVC. This includes the vehicle frame, motion control system, sensory array, and decision-making algorithm (DMA). In order to understand the design solution, which details the various components of the AGV and the integration process, the background of the AGV, its origins, importance to the Coast Guard, and preliminary terms and concepts are important to grasp.

Background

The AGV senior design project at the CGA was created as an opportunity to give students exposure to electrical, computer, and mechanical engineering to facilitate their engineering education. According to the IGVC website, the competition is "multidisciplinary, theory-based, hands-on, team implemented, outcome assessed, and based on product realization," [1]. These elements mix to create an excellent educational tool that will put students' understanding of many critical engineering concepts to practice. At the conclusion of the project students have an opportunity to test their design and learn from design teams of other universities at a nation-wide engineering competition, the IGVC, held at Oakland University in Rochester, Michigan. The event will mark the 16th year the competition has been held, and it is this competition that acted as the driving force for design requirements and development deadlines. For an in-depth cost/benefit analysis of the project see Appendix [A], the project business case.

Several basic terms and concepts must be explained in order to fully grasp the design solution. Autonomy in the context of the IGVC means the vehicle is capable of transiting to a given position while avoiding obstacles without input from a user, or in other words, no one is driving it. The vehicle employs a number of various sensors to provide information in decision making. They are as follows: 1) A Global Positioning System (GPS) unit provides position. 2)

A digital compass provides the vehicle's heading. 3) Digital cameras provide the lane detection capabilities and playback features. 4) A SICK® ranging laser provides ranges to obstacles for obstacle avoidance purposes. The vehicle's basic system consists of 2 rechargeable 12V batteries, which are used to power the SICK laser and the motors and controllers. The sensor information feeds into a laptop onboard the vehicle. This laptop is responsible for the vehicle's decision making capabilities. The laptop communicates to a controller, an intelligent piece of technology that uses transfer functions to control a response to the motor it governs. The controller uses pre-programmed functions called by the laptop to command the motor. The controller controls the motor by sending a series of very precise voltages to it.

The AGV is not a new project within the Coast Guard Academy or among many other colleges. The project began during the 2006-2007 academic year with 1/c Morello, 1/c Ulanoff, and 1/c Laucys working in conjunction with LT Everette, a graduate student at the University of Rhode Island (URI). This design team did a lot of the entry-level work and an admirable first attempt at the vehicle last year. At the conclusion of the 2006-07 academic year the design team had produced a vehicle capable of moving forward and reverse, and turning in a circle [4]. LT Everette had written programs to interpret incoming information from various sensors and make basic decisions as a result, but the code was never tested because the vehicle never achieved the operational state required for testing. Much of LT Everette's code has since been adapted to the new vehicle design which was developed because several potential problems were identified with the vehicle last year's team passed on.

The old frame design of the vehicle was unnecessarily large, making it less agile and unable to fit through a door. At first glance it is clear to see that most of the vehicle was empty space, shown below in Figure 1.

Figure 1: 2006-07 Vehicle

There are several reasons why this is a problem. First, a larger width makes the vehicle inherently more difficult to maneuver and more likely to collide with obstacles on the course. Secondly, the vehicle could not fit through a standard size door. This characteristic was identified as a functional requirement early on with the possibility of using the vehicle for an indoor autonomous navigation project in mind.

The drive system of last year's vehicle was also targeted as a major problem. The vehicle is driven by the motors much in the same way that a tank is driven, with two independent treads on either side. The vehicle's front and rear wheels are connected by a chain wrapped around gears on both of them as shown below in Figure 2.

Figure 2: 2006-07 Drive System

This design requires that the vehicle drive forward on one side and drive in reverse on the other in order to turn in place, in the same way a tank turns. The nature of the tank-style turn makes it very difficult to identify a specific pivot point and greatly reduces maneuverability in general.

The final problem identified with the previous design team's vehicle was the Robotics First motor and controller system. The CGA is involved with other robotics competitions and as a result maintains an inventory of robotic motors and controllers to be used in design projects of other majors. The previous design used the Robotics First motor controller setup because of its availability and history of use at the CGA. However, the Robotics First controller is difficult to modify due to its lack of documentation, and the motors are, as a result, limited in their functionality. In other words, the design capacity was greatly reduced due to the inflexibility of the motors. In order to compete, the vehicle must overcome these problems and achieve several objectives.

Objective

The AGV was ultimately created in order to compete in the IGVC, which has many specific requirements for the vehicle itself and then functional requirements that the vehicle must meet in order to pass the rigors of the various obstacle courses. The design team's objective is to create a vehicle that meets IGVC specifications and can successfully and quickly navigate the autonomous challenge and navigation challenge courses.

According to the official IGVC rules, the vehicle must be propelled by direct mechanical contact to the ground such as wheels, tracks, or pods. The vehicle's dimensions are constrained by the following: 3'-7' long, 2'-5' wide, and no greater than 6' high. Vehicle power must be generated onboard with a maximum vehicle speed is 5 mph, and have both a mechanical and a wireless emergency stop switch. Finally, the vehicle is required to carry a 20lb payload [2].

The vehicle must successfully complete the autonomous challenge according to the IGVC rules. Thus, the, "vehicle must negotiate around an outdoor obstacle course under a prescribed time while staying within the five mph speed limit, and avoiding obstacles on the track" [2]. Judges will rank the entries that complete the course based on shortest time taken, and those that do not complete the course on longest adjusted distance traveled. There will be at least six feet of clearance on either side of obstacles in the center of the course and six feet of clearance on the non-boundary side of obstacles placed up against the boundary. Obstacles include 5-gallon pails, construction drums, cones, pedestals, and barricades [2].

The vehicle must also complete the navigation challenge in which it must autonomously travel from a starting point to within 2m of a number of target destinations and return to home base given only the coordinates of the targets in latitude and longitude. This navigation challenge course also has many obstacles placed on it, including construction barrels, barricades, fences, and others. The course is judged on the shortest time taken to travel to all necessary waypoints.

The objective of the design team was to build a vehicle capable of successfully transiting the IGVC courses. This goal was achieved in two primary phases: 1) the frame and drive system, and 2) the "intelligent" portion of the design: sensor integration and decision making capabilities. The two phases were completed on schedule by the end of the fall and spring semesters respectively. For a detailed schedule of the design and implementation process, see Appendices [B] and [C], the project management plans for both the fall and spring semesters. The method of carrying out these phases is outlined in the design solution.

System Design

In order to develop a design solution that satisfactorily met all of the IGVC requirements which are included in detail in Appendix [D], the formal requirements document, a "bottom-up" approach to the design was used. A bottom-up design was most practical because it focused on the frame of the vehicle first, and then from there moved into the more high-tech portions of the vehicle. It is impossible to add sensory equipment until after a frame has been built to accommodate the sensory equipment. Because the autonomy of the vehicle removes a human from the navigation equation, the vehicle must have components that closely parallel various parts of the human body. The four components of the design include: 1) the vehicle frame, or the skeleton, 2) motion control, or the muscular system, 3) sensor processing, or sight, and 4) the Decision-Making Algorithm (DMA), or the brain. As mentioned above, the design components of the project move from low-tech to high-tech, which allows the project to be completed in stages. By following the design solution and support processes included in the project support plan, Appendix [E], all 4 stages of the design were completed and next year's team has a solid foundation from which to start.

Vehicle Frame

The first stage in the vehicle design was the vehicle frame. The vehicle frame design was a change from the vehicle constructed by last year's group because instead of a relatively wide, four-wheel, "tank-driven" design we switched to a narrower, three-wheel, front wheel driven design. Last year's design was considered to be "tank-driven" because it had two front wheels and two rear wheels connected to each other on each side by a bicycle chain, much like wheels on both sides of a tank are connected by treads. While this design does synchronize wheel velocity on both sides and provide high stability, it creates trouble when trying to turn precisely

on a given pivot point. In autonomous navigation, it is critical for the vehicle to know its position at all times, and after an inconsistent turn, the vehicle will likely assume its position incorrectly. The reason for this inaccurate assumption is that the pivot point is difficult to calculate or changes for different turns. To correct this problem, the team switched to a design with two front, motor-driven wheels, and a rear, free-rotating, free-wheeling caster wheel modeled after vehicles used in previous IGVC's. The independent front wheel drive system can be seen below in Figure 3.

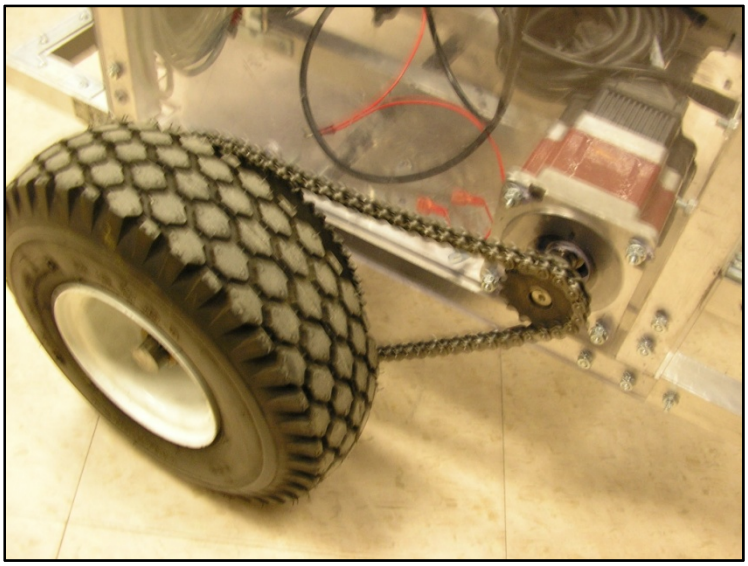

Figure 3: Front Wheel Drive System

The free-wheeling, free rotating caster that gives the vehicle its high maneuverability is shown below in Figure 4. The wheel has a large radius and a tire to help handle bumps/divots and absorb shock in the IGVC course. If the vehicle bounces around a lot, it will assume its position incorrectly and be less efficient.

Figure 4: Rear Caster Wheel

This design made it much easier to identify and create a specific pivot point, but has also reduced vehicle stability minimally. During testing the vehicle showed no problems due to a lack of stability. The narrower frame design allows the vehicle to fit through a standard size door and opens up the possibility of attempting indoor navigation projects in the future.

Motion Control System

After the completion of the vehicle frame, came the second phase of the project, the motion control system. The Robotics First motor/controller system used by last year's team was abandoned for two main reasons. 1) There was no supporting documentation concerning PC interaction with the controller or MATLAB interaction with the code that was uploaded to the controller. 2) The Robotics First motor/controller was very limited in functionality and programmability. QuickSilver Controls was chosen because it is currently being used in several successful IGVC groups from various colleges. After purchasing the motor/controller and software from QuickSilver and beginning to become familiar with the system, it is obvious why winning teams are using this system. The QuickControl software is very powerful and performs

the entire computer control system work of transfer functions and response. The exhaustive command reference and user manuals included in the software package allow for controller programs that accurately manipulate quantities such as velocity, number of rotations, direction, and time duration for various step and ramp functions. Initially, the motion control phase of the project looked to be one of the more difficult steps, but after receiving the QuickControl software the team was able to generate an arsenal of motion functions that accurately perform turns, forward and reverse movements, and much more. With the current motion control setup the design should never be constrained by a lack of motion control.

Sensor Array

Thirdly, the design solution requires sight, or a sensor array. The sensors that are processed and integrated to create sight are: GPS, a SICK® ranging laser, digital cameras a digital compass, and a shaft encoder. All sensors must feed into the decision making algorithm in such a way that the correct decision can be made from the information on hand.

GPS

For outdoor purposes, like the IGVC, GPS provides constant position information and validation. This data allows the vehicle to know where it is, where it was, and where it wants to go—all extremely important information when attempting to navigate autonomously. Without GPS there is no way for the vehicle to recognize that it has reached its goal, or to "validate" its position. A Javad Maxor-GGDT GPS receiver, shown below in Figure 5, was shipped in by NAVCEN to aid the project.

Figure 5: Javad GPS Receiver/Antenna

The GPS receiver has 20 channels of dual frequency GPS and GLONASS and is horizontally accurate to within 15mm + 1.5ppm x D. The GPS receiver/antenna is the single most expensive piece of equipment on the vehicle (~$12,000) and more than exceeds performance requirements for the IGVC problem.

SICK Laser

The SICK® ranging laser is used for obstacle detection and is shown below in Figure 6. Thus, while GPS provides a specific position "objective," the laser aids the vehicle in achieving that objective as it transits. LT Everette's graduate work is used to interpret the SICK® laser information. LT Everette's study last year focused on "sensor integration into the MATLAB computing environment" and, "storage methods for all sensor data, in order to allow for post processing functions," two skills that are critical to autonomous navigation using the SICK® laser [3].

Figure 6: SICK Laser

LT Everette's code is currently being used in the vehicle DMA to both read, process, and render a decision based on SICK laser information. LT Taylor also developed a MATLAB program that can map any objects within the immediate area of the vehicle. The program then saves each of the objects mapped, effectively "remembering" where they are located. If LT Taylor's software is integrated, the robot will be able to use the mapped objects to gain an approximation of its own location.

Digital Camera

In order for the autonomous ground vehicle to be capable of competing in the IGVC, the vehicle must be capable of completing the autonomous challenge according to the IGVC rules. Since the autonomous challenge requires the vehicle to navigate a path to a destination, one crucial capability that the vehicle needed was lane detection. Developing a system for lane detection required a set of functions to be created that could take a snapshot using a camera mounted on the front of the vehicle and process the image to extract the straight lines in order to find a safe range of bearings.

The first step in designing the lane detection capability was to create code that could extract the straight lines from an image. Extracting the straight lines from an image was completed using previously written code from LT Everette. LT Everette's code utilized the edge detection and houghlines MATLAB functions to extract the straight lines from an image. The initial code written for the purpose of locating straight lines in an image was much too sensitive. When the code was initially tested, it extracted straight lines from not only the legitimate lines in the image, but also of any patches of light or shadow in the image. However, by using "sobel" edge detection and increasing the minimum length of detected lines, the code was successfully altered so that it only extracted the lines of interest from the image.

Once code was developed that could extract the correct lines from an image, the next step of the process was to develop a method for determining a range of bearings that the vehicle could travel to remain inside the designated lane. In order for the method to be inclusive, it needed to take into account the various line configurations with which the vehicle could be presented. The first line configuration that was considered was the case where there were two lines visible on the image. An example of a line configuration with two lines visible can be seen below in Figure 6.

Figure 6: Image of Line Configuration with Two Lines Visible

The method developed for processing this type of line configuration involved finding the midpoints of both of the lines extracted. In order to understand the process of locating the midpoints, it is crucial to first understand the way that the results are returned from the houghlines function. When the houghlines function finds a line in an image, it returns the x-y coordinate for the start and end of the line. Using the x-y coordinates of the starting and ending location of the line, the midpoint of the line in the x direction and the midpoint in the y direction can be found with equations 1 and 2, respectively.

$$Midpoint_X = \frac{x_e - x_s}{2} \quad \text{(eq.1)}$$

$$Midpoint_Y = \frac{y_e - y_s}{2} \quad \text{(eq. 2)}$$

Once the midpoint of each of the lines was calculated, the x distance and y distance from the center of the image to the midpoints was found. Using the x-distance and y-distance from the center of the image to the midpoints, the angle between each midpoint and the center of the vehicle was calculated using equation 3.

$$\theta = \tan^{-1}\frac{x_{c-m}}{y_{c-m}}; \quad \text{(eq. 3)}$$

where $x_{c\text{-}m}$ and $y_{c\text{-}m}$ represent the x and y distance from the midpoint to the center of the image.

Finally, by adding the two angles calculated between the midpoints and the center of the image, a total angle of headings that would allow the vehicle to remain inside the lane was calculated. In figure 4 below, the extracted lines, midpoints, and the calculated angles can be seen for the line configuration shown in Figure 7.

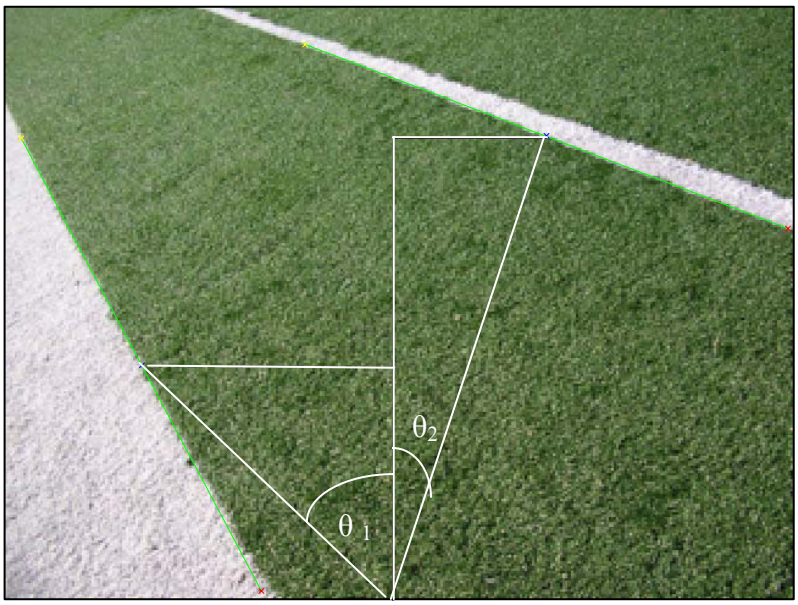

Figure 7: Image of Line Configuration with Two Lines Visible
with Extracted Straight Lines, Midpoints and Angle of Safe Travel

After a method for processing an image with a line configuration containing two lines visible was completed, the next line configuration that was considered was the possibility that only one line was visible on an image. One example of this type of line configuration can be seen below in Figure 8.

Figure 8: Image of Line Configuration with One Line Visible

The method for processing an image with a line configuration of one line visible is similar to the method used to manage a line configuration of two visible lines. However, with only one line visible on the image, it is impossible to find an inclusive range of bearings that are safe for travel. Therefore, the method for one line configuration finds a cutoff bearing rather than a full range. Rather than finding the midpoint of the line detected, the method for a configuration of one visible line uses one of the endpoints of the line, whichever endpoint is further away from the vehicle which corresponds to the closer to the top of the image. Using this point instead of a midpoint, the x-distance and y-distance are once again found. Once again, using equation 3, the angle between the center of the vehicle and the endpoint can be found. Conversely to the previous method, the angle calculated is not the safe bearings of travel. In the example shown in Figure 8, the actual safe bearings would be any bearing that is to the left of the calculated angle. In Figure 9, the extracted lines and calculated angle can been seen, as well as the safe bearings that are to the left of the line connecting the position of the vehicle with the endpoint of the line. One additional measure that needed to be taken in order to properly process the image was the situation where both sides of the same line were detected, as can be seen in Figure 9. In order to handle this situation, code was added that would simply ignore one of the lines if their midpoints were within certain range of each other.

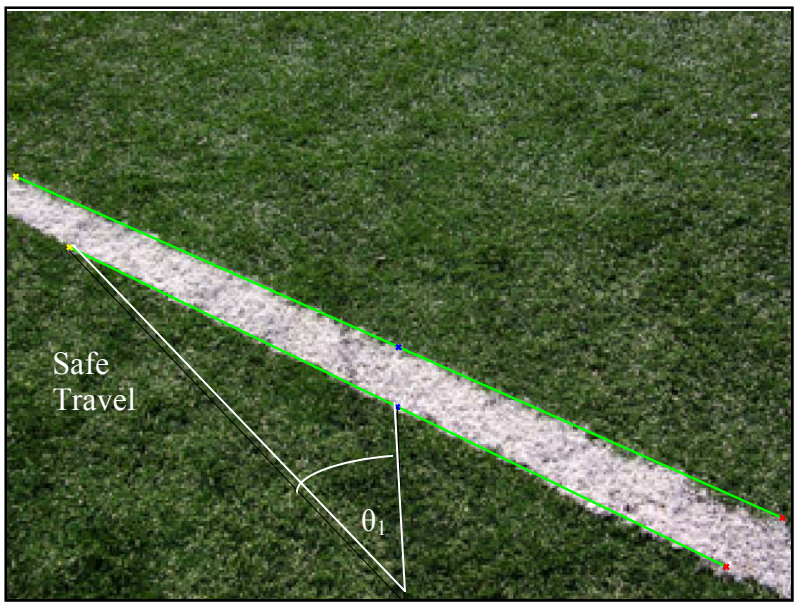

Figure 9: Image of Line Configuration with One Line Visible
with Extracted Straight Lines, Midpoints and Area of Safe Travel

With methods for both line configurations developed, the next step in building a lane detection system was to design technique for acquiring an image without user interaction. When the vehicle was in operation, the lane detection function must be capable of being called repeatedly, which led to a problem with the acquisition of the image. Initially when the function was written, a video input object was created at the beginning of the function and used to capture an image. However, when the function was called more than once, the function tried to create another video input object identical to the one already created. In order to correct this error, another function was created whose sole responsibility was to create the video input object, which was then passed to the main program. Once the video input object was passed to the main program, when the function for image processing was called, it could be passed the video input object in addition to the current heading of the vehicle and the minimum length desired. Passing the current heading and the desired minimum length gave the function greater flexibility to adapt its calculated safe bearings relative to the vehicle to actual bearings depending on the vehicle's orientation. In addition, by passing the minimum length to the vehicle through the main

program, the user has the freedom to adjust the sensitivity of the houghlines function depending on the quality of the images achieved in the vehicle's current environment. Once both of the functions were completed, the vehicle had a lane detection capability that could be used repeatedly to allow the vehicle to remain inside a designated lane.

Digital Compass

The primary purpose of the digital compass is to provide a constant check that the vehicle is heading in the correct direction. When transiting to a given waypoint the vehicle moves by calculating the vector, or direction and range, that will take the vehicle to the waypoint. If the vehicle for whatever reason travels outside a maximum allowable deviation from desired course, the vehicle will stop, recalculate its ideal vector, and correct its course. The model being used is the Honeywell HMR3000 three axis compass which has a 0.5^{o} accuracy and a fluidic tilt sensor for $+/-45^{o}$ compensation. The compass specifications have met all problem needs and make the compass a reliable sensor; however, as with all magnetic compasses, the compass becomes extremely inaccurate when used within the metal frame of a building due to magnetic interference.

Shaft Encoder

The shaft encoder, which is a function built into the QuickSilver motors, is not currently being used in the design solution but it could be used to calculate distance traveled by the individual wheels. Currently GPS provides accurate position validation but if an indoor problem was attempted, the shaft encoder would be critical. It allows effective control of the distance the vehicle travels based on the number of wheel rotations. There are some inherent problems however, with relying too heavily on shaft rotations: if the wheels slip at all, the distance traveled will be incorrect and the vehicle will assume its position incorrectly. Despite this potential

obstacle, the shaft encoder would very likely be the primary means of determining position for any indoor projects because GPS does not work indoors due to its low signal strength.

Decision Making Algorithm (DMA)

The primary purpose of the DMA is to integrate all sensory information and render a decision based on it. The flow of information is depicted well below in Figure 10:

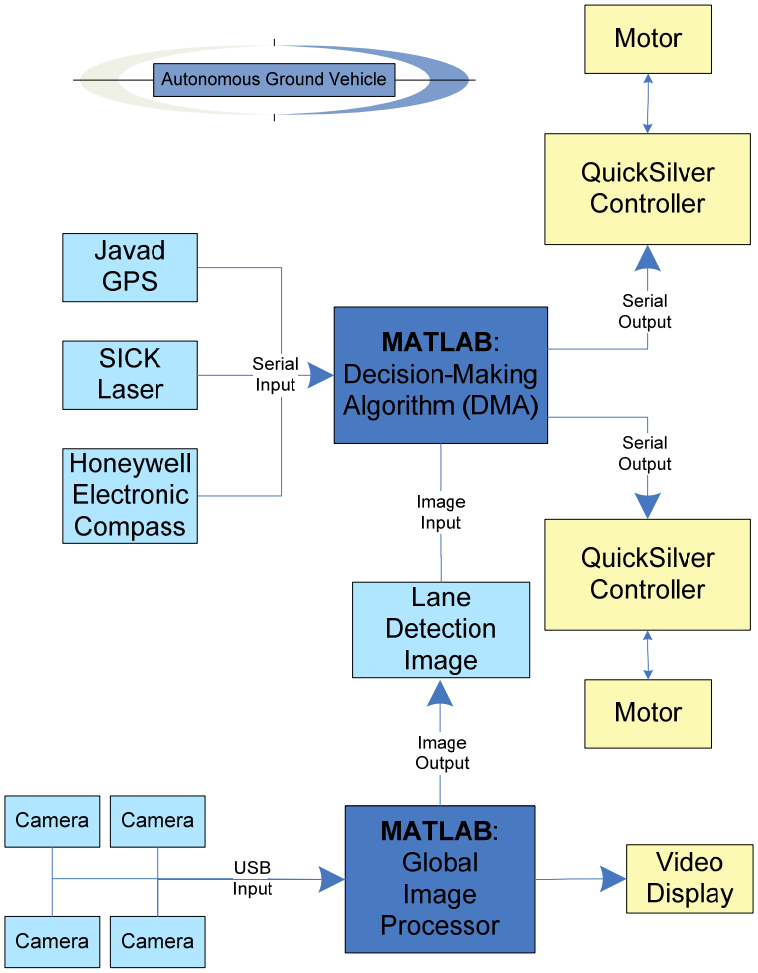

Figure 10: Design Model

System Integration

The magnetic compass, Javad GPS, SICK laser, and the motors operate using the serial communications port. The serial communications port is similar to file reading and writing, but

instead it uses a specific port on the computer. Many of the components of the vehicle require two way communications to either retrieve data from them or send commands to them. The read and write commands for the GPS, and the motors uses characters and actual words or numbers while the Laser uses a hexadecimal system.

The magnetic compass is setup so that when it is plugged in it will start writing data to the computer, which can be retrieved using the serial port read commands in MATLAB. To read the data from the compass it is initialized using a baud rate 19200 bits per second. For the current laptop setup with the serial hub, it uses comport 6 (com6). The compass transmits a few details in its data string such as the degrees magnetic, the pitch, and the roll. The current vehicle get_heading.m function, the only retrieves the degrees magnetic, but the pitch and roll could be retrieved as well and used in the future.

The Javad GPS receiver requires some two way communication so that it begins transmitting a string out just like the compass. However, if it is hard coded right after the serial port is connected, it will start transmitting the data immediately. Once it begins transmitting the data, the get position function will perform the string manipulation to retrieve the latitude and longitude. The GPS is setup using the Port A on the GPS module with the antenna attached. Currently, the GPS is configured to transmit at 115200 bits per second, which is plenty for the data to be retrieved almost instantly. With the current vehicle setup on the laptop, the GPS is plugged into the comport 7 (com7). The initialization command for the GPS is "em,/dev/ser/a,nmea/GGA:1", which means that it will send a NMEA string periodically every second using port A. The manual for the GPS contains other useful information about the GPS and other commands, but this command works very well for continuous position reports when requested. Currently, the get position command will read the data being transmitted, parse the

latitude and longitude from the data stream, and output it as a numerical value in decimal form, which can be used with our other functions.

The SICK laser requires the user to use some two way communications, but the data from the laser is slightly different than the two other sensors. Currently, the AGV uses the code LT Everett wrote the previous year. The start_sick.m file setups up the laser using the correct hexadecimal commands at a baud rate of 9600 bits per second. The current laptop setup has the laser on comport 9 (com9). Anytime the user wants to scan for data, the get_scan.m function will work, but it takes about .8 seconds pause minimum before the data is ready. The data from the laser is given in millimeters up to 8 meters. The other functions to use with the laser are the obsheading.m file which uses a north up approach to determine a safe bearing from the laser data. Currently, the vehicle uses 0 degrees to indicate straight in front of the vehicle, and the result will one of these two cases: (1) the safest bearing given a minimum range and the approximate number of degrees that are required for clearance for the vehicle or (2) the vehicle will give a 90 degrees to the right result to indicate that it is a dead end with not enough clearance.

The motors operate independently and require the user to write commands to them. Each motor is setup using 57600 bits per second. Motor 1 is the on the left side of the vehicle on comport 4 (com4) (next to batteries) and motor 2 is on the right side of the vehicle on comport 5 (com5) (by the motor controllers). There are hundreds of commands in the command manual, which can just simply run, return a value such as the click counter, or run and return a value. The manual explains all of these commands it great detail, but there are a few simple commands that are vital to the vehicles movements. The best solution to adding another command will be to copy one of the other commands such as the vmi.m file and change the variables. Currently,

there is a mode to drive a certain distance as a specific velocity and acceleration (evm.m), to drive continuously at any specified velocity and acceleration (vmi.m), and a mode to stop the motion by resetting the motor command (stopmotor.m). Combining these functions, the vehicle also has a turn mode to turn either by moving only one wheel forward (radiusturn.m) or to move forward on one wheel and reverse on another (rotationturn.m). To turn to the left, it is a negative number of degrees and the right is a positive number of degrees. For example, if the vehicle needs to turn 30 degrees to the left would be "rotationturn(-30, motor1, motor2)" while 30 degrees to the right would be "rotationturn(30, motor1, motor2)".

Waypoint Navigation

To navigate through waypoints, the vehicle must first be set up with the waypoint matrix. Initialize the GPS, setup a variable for the waypoint matrix which will contain both the latitude and longitude in the form from the get_position.m file, and add various locations into the matrix. The waypoint_navigation.m file will take a position fix at the current location of the vehicle, rotate towards the waypoint using the azimuth output of the dist_to_wpt.m file, and then drive forward until the current position and the waypoint vary by more than 15 degrees or if the vehicle is within three feet of the waypoint. It will then move on to the next waypoint that the user requested. Obstacle avoidance can be layered on top of this function so that it checks the laser data for the clearest path then if it is safe it does not stop to turn away and enter the obstacle avoidance loops.

Obstacle Avoidance

Obstacle avoidance is currently performed using the minimum distance to obstacles and adjusting the wheel speed based on those distances at close ranges. It will check the laser constantly although the update time is rather slow, then it will determine if it is headed on a safe

path only stopping if it is in danger. The basic steps and ideas behind obstacle avoidance can be simplified down to a couple steps. In the flowchart below, Figure 11, it takes a scan, checks the clearance and obstacle heading in front of the vehicle, and then makes a decision based on the result of that function.

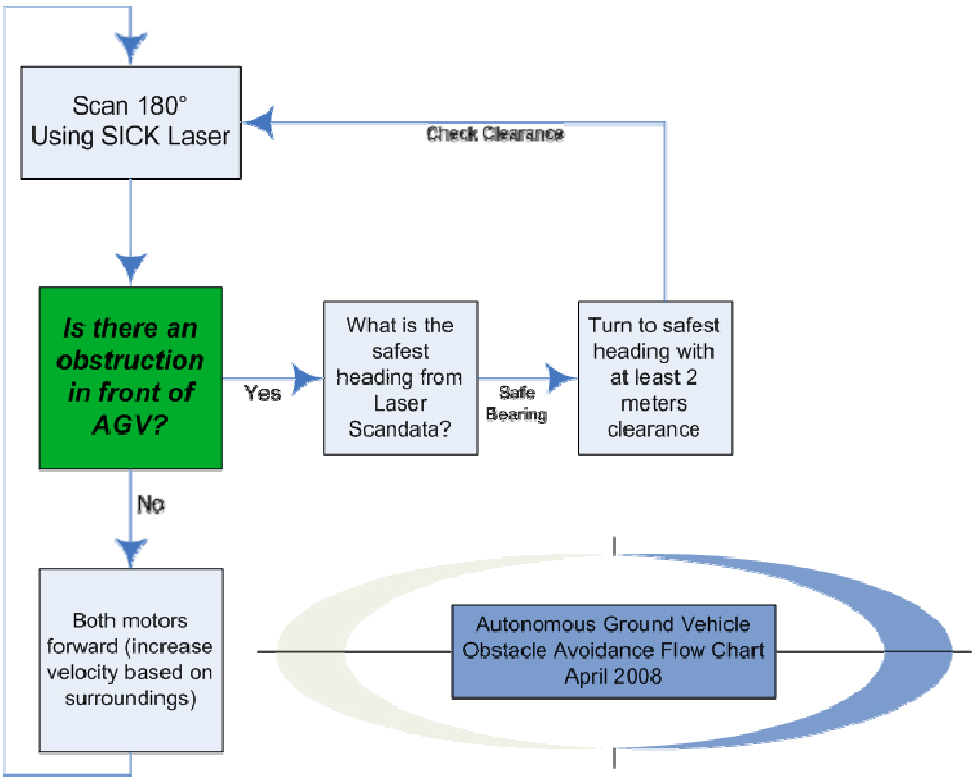

Figure 11: Basic structure of the decision-making algorithm for obstacle

For example, if the algorithm ran "obsheading(scan1, 0, 2500, 20)" then it will check directly in front of the vehicle and make sure that the minimum distances in a range of 10 degrees left and right of 0 degrees are no less than 2500. If the result is a number greater than about ±4 degrees, that is the new safe bearing, which correlates to a degrees to use in the turn functions then loops back to the scan to check the clearances. On the other hand, if it returns a value between ±4 degrees, the vehicle will drive straight and loop back to scan while it is moving. Currently the

best solution for our design of indoor obstacle avoidance and navigation is the script in the obsmovement.m file. For a closer look at commented MATLAB code used in the DMA see Appendix [H].

Design Summary

All components have been integrated together on the vehicle and implemented as described. For a more in-depth discussion of the design decisions made see Appendix [F], the project design specifications.

Results

After carrying out the test plan included in Appendix [G], much was learned about the functional capabilities and efficiency of the vehicle's decision making algorithms. First, the lane detection component of the vehicle design was tested.

Lane Detection Results

After the lane detection segment was designed and implemented, a battery of tests were run upon it to determine its actual capability and constraints. The tests were run in accordance with the lane detection test described in the test plan. In order for the testing to be comprehensive, the function must be tested with every type of possible line configuration. Therefore, the function was passed images that contained both two lines and one line, with the orientation of the lines varying to include lines with positive, negative, zero, and infinite slope. All of the tests conducted with lines whose slope was positive or negative yielded the correct range of safe bearings, which was verified with manual calculations.

In the process of testing the lane detection capability, a number of constraints on the capability were discovered. First, the lane detection capability has no information about the beginning and ending locations, so if the vehicle gets turned so that it is facing backward, the

function will continue to process images and return results that are in the opposite direction the vehicle wants to navigate. Therefore, the decision-making algorithm must be written in such a way that if the vehicle gets turned around on the course, it will not look to the lane detection function order to try to regain the correct direction. Secondly, two of the line configurations that were tested caused errors in the lane detection capability. When there was only one line in an image, and that line was either exactly vertical or exactly horizontal, the function could not decide on which side of the line were the safe bearings. Although this may seem like a large problem, in reality it is acceptable while still maintaining a reliable system. In order for the vertical or horizontal lines to cause an error, they must be exact. Even a tiny deviation from true horizontal or true vertical and the lane detection capability works exactly as specified. Due to the fact that the likelihood of achieving perfectly horizontal or vertical lines is miniscule, the constraint caused by this error is negligible.

System Integration Results

Using the MATLAB script mentioned earlier in the design section, the AGV is navigating through GPS waypoints using the compass to point it in the correct direction. The vehicle will switch waypoints at about 3 feet from the latitude and longitude of the waypoint. In the rigorous tests that were completed of the GPS navigation mode, the vehicle achieved about 1 foot accuracy at most of the waypoints and seemed to over correct too much to turn towards the waypoint. Most of the time the vehicle will slowly pass the waypoint, stop, turn towards the point, then move forward, and repeat this process a few times until it is within the range to move on to the next waypoint. It can be setup to slow down as the range decreases to the target. The GPS was accurate and precise in that the vehicle repeatedly maneuvered to the same exact mark on the testing field where a waypoint was positioned. There was about a 10 to 20 degree turning

error since the GPS antenna was positioned in the rear of the vehicle and the magnetic compass was positioned in the center of the vehicle.

Testing of the obstacle avoidance was completed numerous times. Figure 12 below shows that when the vehicle faced a corner, the algorithm that determined the safe heading chose the least amount of turning degrees from the intended heading of 0 degrees. The obstacle heading function was setup to look for a minimum of 2500 millimeters at 0 degrees with a 20

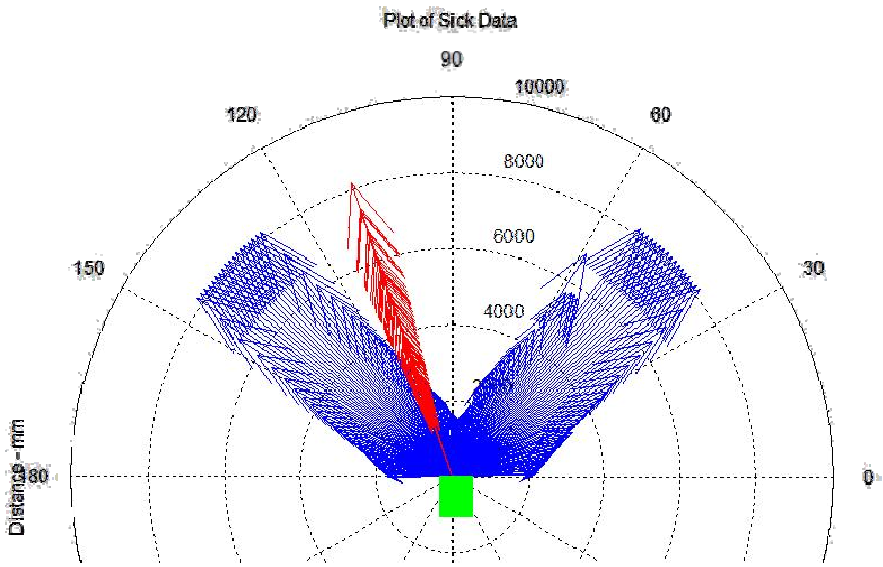

Figure 12: SICK Scan facing the corner with two hallways on each side with limited clearance. Red arrow indicates intended safe direction.

degree swath. If these numbers are dynamically adjusted so that it followed the wall, or an edge like a lane, then the vehicle would make less turns away from the obstacles in its path.

Figure 13 shows that even with a large clearing, the vehicle picks the path that meets the requested ranges. Again, to make the algorithm "smarter," these ranges need to be dynamically adjusted so that it picks the best path and plans out where it will be in the future based on the scanned data. Currently, another solution implemented is to create a range for the vehicle to meet. For example, the AGV knows that it has to make a 10 degree turn to stay in the safest

area, if one of the wheels is slowed or sped up, then the number of degrees reduces slowly and it corrects without stopping the vehicle. This was tested successfully using various speed modes, minimum distances from 70 to 0 degrees for the left wheel and 180 to 110 degrees for the right and multiplying the speed of the wheel by a factor of the angle.

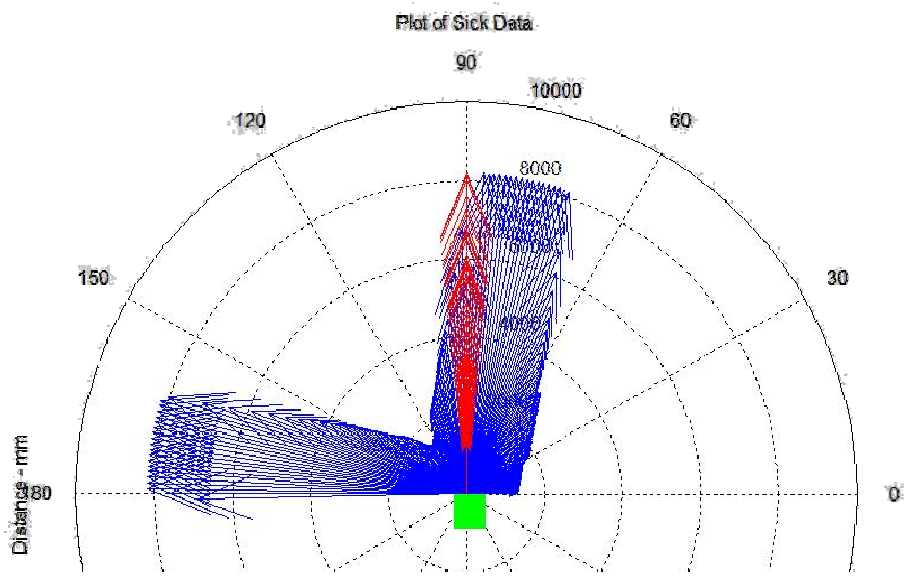

Figure 13: SICK Scan looking down a hallway with a lot of clearance. Red arrow indicates intended safe direction.

Figure 14 shows that the vehicle is going to be expecting a dead end, however, the obstacle heading function does not account for this future incident. Once the error is noticed, the vehicle will need to reevaluate the situation once it reaches the minimum range of 2.5 meters. It then keeps turning 90 degrees to the right until it finds the next best path. The vehicle needs to remember paths from before as it is moving along to overcome this mistake and act "smarter."

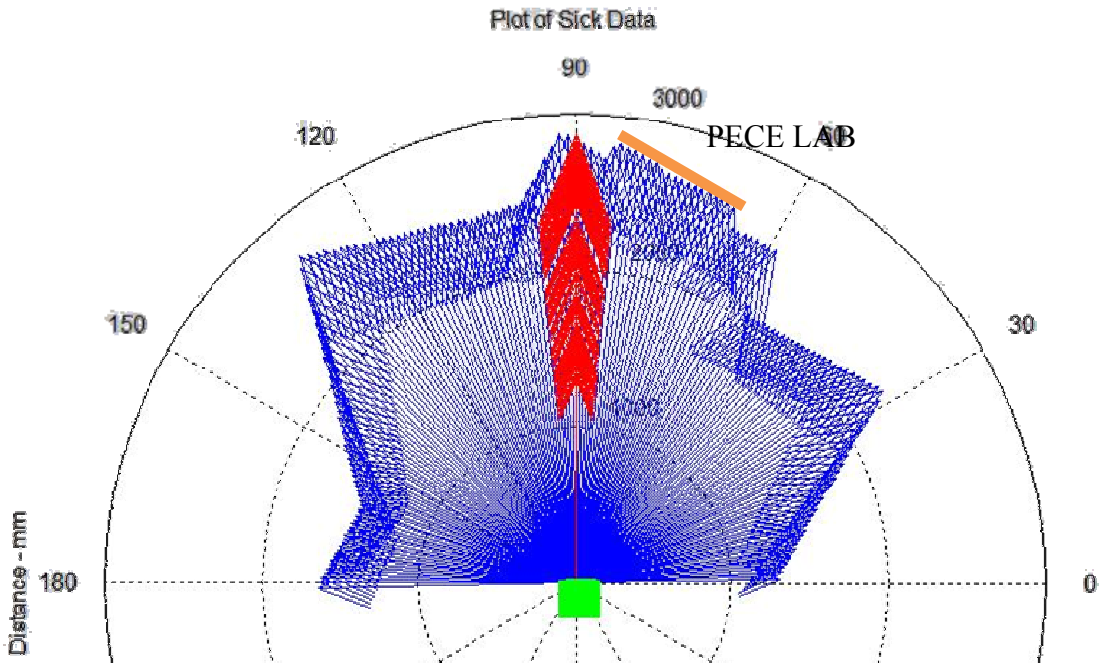

Figure 14: SICK Scan facing Mac 214 showing more than 2.5 meters clearance. Red arrow indicates intended safe direction.

Results Summary

At the beginning of the 2007-2008 academic year the design team was presented with a vehicle that was crudely put together and could only move forward, reverse, and spin in a circle with very little control. Since then the vehicle has undergone immense changes both externally with its frame, motion control system, and sensor array, but also internally with the manner in which the vehicle is programmed and controlled. This top to bottom overhaul has yielded excellent results. The completed vehicle is shown below if Figure 15.

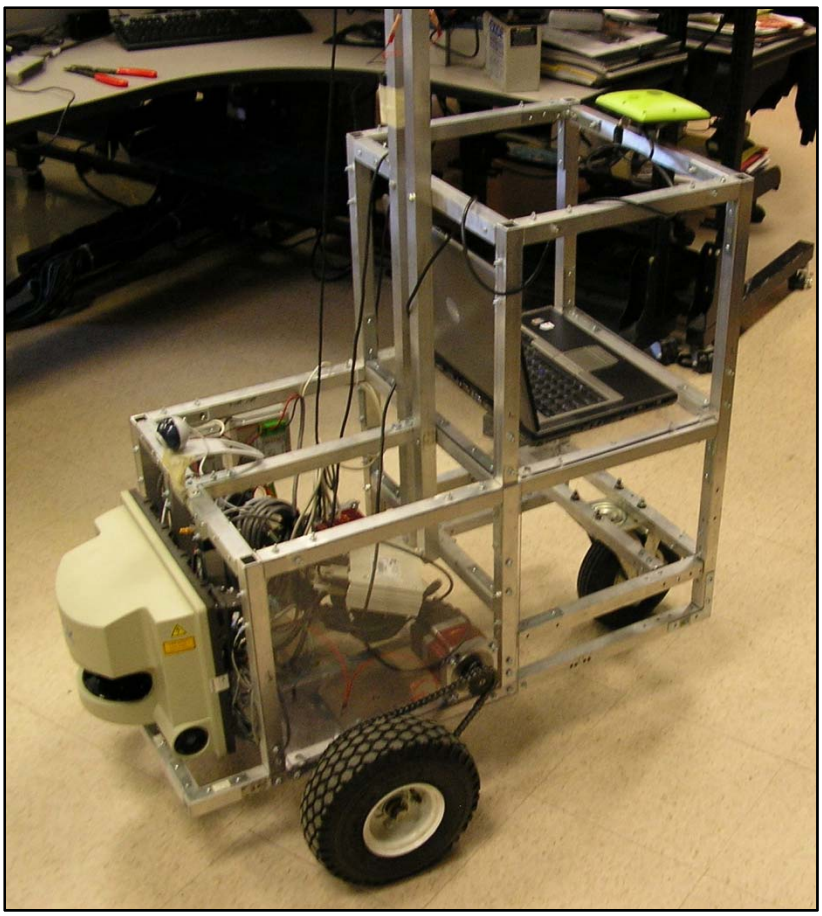

Figure 15: Completed AGV

For the IGVC autonomous challenge the vehicle must be able to navigate around obstacles while staying within the defined boundaries. The vehicle is currently able to drive forward continuously, successfully avoid obstacles, and also detect boundaries and avoid them. For the IGVC navigation challenge the vehicle must be able to navigate through a waypoint matrix while avoiding obstacles. The vehicle is currently capable of performing these tasks as well. The vehicle cannot yet do all of them at the same time efficiently but it has every component it needs to do this. The only remaining step before complete integration is to continue to develop the DMA code and test it. The design team just needed more time to achieve ready status for the IGVC and there is absolutely no reason the following design team

should not compete next year. If the current design team had one more month of focused work the vehicle would be competition ready.

So while the vehicle is not where it needs to be for the IGVC, when it is compared to the starting point, a vehicle essentially incapable of performing the most basic tasks, the results are clear and abundant.

Conclusions

Autonomous navigation is an important concept that has the potential to minimize risk involved with dangerous Coast Guard search and rescue. The AGV developed this year will continue to hone these necessary skills. After implementing the design solution described above, the AGV is ready to compete in the IGVC and successfully perform all required tasks. Information from the SICK® laser, GPS, compass, and digital cameras is processed and integrated into the MATLAB decision making algorithm, which then makes a navigation decision based on the desired position and the current perceived "safe" routes. While the vehicle does have all necessary capabilities to compete in the IGVC, the decision was made in February that even though success was possible, it was not probable, and USCGA would not compete in the 2008 IGVC. When the vehicle does make the journey to Michigan in 2009 with the excellent foundation set in 2008, the vehicle will represent the United States Coast Guard Academy well in the IGVC. It will mark an important milestone for USCGA engineering and speak volumes to the ingenuity, competency, and dedication of USCGA engineers.

References

[1] "The Purpose of IGVC." IGVC—Intelligent Ground Vehicle Competition. 01 SEP 2007. IGVC. 21 SEP 2007 <http://www.igvc.org/deploy/rules.htm>.

[2] "Official Competition Details, Rules and Format." IGVC—Intelligent Ground Vehicle Competition. 01 SEP 2006. IGVC. 21 SEP 2007 <http://www.igvc.org/deploy/rules.htm>.

[3] Everette, Dan. "Design, Construction, and Analysis of an Autonomous Ground Vehicle." (2007): 9.

[4] Morello, Nathan, Ulanoff, Adrian, Laucys, D. "Autonomous Ground Vehicle." United States Coast Guard Academy (2007).

Bibliography

[1] Gombar, Brett, Andrew Bacha. "Johnny-5" Virginia Tech (2005): 1-15.

[2] Joerger, Mathieu, Boris Pervan. Autonomous Ground Vehicle Navigation Using Integrated GPS and Laser-Scanner Measurements. San Diego: Position, Location, and Navigation Symposium, 2006.

[3] Ishikawa, Ken, Kazuyuki Kobayashi, and Kajiro Watanabe. "A Lane Detection Method for Intelligent Ground Vehicle Competition." SICE 2003 Annual Conference. 1 (2003): 1086-1089.

[4] Ito, Masayoshi, Kazuyuki, and Kajiro Watanabe. "Study on the Method to Control the Autonomous Vehicle by Using DGPS." Proceedings of the 41st SICE Annual Conference. 4 (2002): 2382-2384.

Appendices

[A] Business Case

[B] Project Management Plan (fall)

[C] Project Management Plan (spring)

[D] Requirements Document

[E] Support Plan

[F] Design Specifications

[G] Test Plan

[H] Commented DMA MATLAB Code

APPENDIX A: BUSINESS CASE

Needs Statement

Last year, a team of first class cadets began work on the Autonomous Ground Vehicle project. During the time that they worked on the project, they developed many of the components that could be used to make an autonomous ground vehicle operate. In order to continue on the foundation that last year's team built, our team must further develop the autonomous ground vehicle to the point where the vehicle is ready to participate in the Intelligent Ground Vehicle Competition (IGVC) in June. It is the IGVC that drives the needs for this project. In order to have the vehicle ready for competition, our team must develop a moving platform that can carry, integrate, and interpret various sensory components. The vehicle design from the previous year was an excellent start to solve this problem; however, there is much that it lacks and the needs of the competition have not been met. Therefore, a new platform design is absolutely necessary.

The new platform must be simple, because a new group of students will be working with it every year and should not be slowed down by an excessively complex design. It must also be adaptable and reusable, as it will very likely be used for widely varying tasks and competitions. The design must be repeatable, so that future students could duplicate, or upgrade our design without tearing the entire vehicle apart. The design must be operational, dependable, and meet the specific requirements of the IGVC; if the movement of the platform is consistent, it will take less time to deal with the basic workings of the vehicle and allow more time for programming the movement and integrating the sensors. All of these needs will increase our potential to succeed in future competitions.

Objectives Statement

The objective of this project is to design and develop an autonomous ground vehicle that can successfully compete in the Intelligent Ground Vehicle Competition in June and serve as a platform for competitions in years to come. Perhaps the most important objective involved with this project is to build a flexible, adaptable, reusable vehicle. In order to accomplish the objective, the platform will be redesigned to be a front-wheel drive vehicle with two rear free-rotating, free-wheeling wheels, and all of its system components included inside the vehicle body. The vehicle will fit through a standard size doorway. Also, the redesigned vehicle must allow for easy access to its components should the need arrive for troubleshooting.

Cost Benefit Analysis

As the Coast Guard begins to update its technology through different initiatives such as deepwater, unmanned vehicle operation is becoming a key development. The purpose of the project is to develop an unmanned, autonomous ground vehicle that can navigate without human aid once given a destination as well as avoid any obstacles that may present themselves while the vehicle is on its way from the starting point finish. Currently, the Coast Guard is developing similar technology for its planned unmanned aircraft. Although there is a difference between the development of a ground based vehicle and an air based vehicle, much of the technology that will allow for navigation and obstacle avoidance is similar. Once Coast Guard unmanned

aircraft are developed, the risk to human life will be greatly diminished during dangerous missions. In order to acquire this capability, the Coast Guard must spend time and resources developing this technology. After the initial costs are overcome, the Coast Guard will have a system that can go out on more dangerous missions without increasing the risk to human life. Overall, the benefits of such a capability far outweigh the initial costs that it would take to acquire such an important capability. Since the autonomous ground vehicle project deals with autonomous navigation and obstacle avoidance, it could clearly benefit the Coast Guard that is developing similar technology.

APPENDIX C: PROJECT MANAGEMENT PLAN (FALL)

Intelligent Ground Vehicle

PROJECT PLAN

09 OCT 2007

3. INTRODUCTION

The purpose of this project management plan is to describe the timeline for work to be accomplished on the intelligent ground vehicle during the spring semester.

3.4 Project Description

The Intelligent Ground Vehicle project is to continue the work done in the 2006-2007 AY and develop an autonomous vehicle that would be able to compete in the annual Intelligent Ground Vehicle Competition. This involves using sensors to detect and avoid obstacles as well as follow a path designated by two lines.

1.2 Project Background

As the Coast Guard begins to update its technology through different initiatives such as deepwater, unmanned vehicle operation is becoming a key development. The purpose of the project is to develop an unmanned, autonomous ground vehicle that can navigate without human aid once given a destination as well as avoid any obstacles that may present themselves while the vehicle is on its way from the starting point finish. Currently, the Coast Guard is developing similar technology for its planned unmanned aircraft. Although there is a difference between the development of a ground based vehicle and an air based vehicle, much of the technology that will allow for navigation and obstacle avoidance is similar. Once Coast Guard unmanned aircraft are developed, the risk involved with dangerous missions will be greatly diminished.

1.3 References

16th Annual Intelligent Ground Vehicle Competition Official Rules.

2. ROLES AND RESPONSIBILITIES

- Sponsors: C2CEN, NAVCEN
- Project Advisors: CAPT Hartnett, Prof. Swazsek, LT Taylor
- Project Team Members: 1/c Seekatz, 1/c Matejka, 1/c Reckley

3. PROJECT DESCRIPTION, SCHEDULE, AND RESOURCES

The work for the project is broken up into three main concentrations: 1) motion control, 2) sensor array with a focus on image processing and lane detection, and 3) the Decision Making Algorithm (DMA). In general these concentrations can be developed in parallel; however, in order to begin advanced testing the three concentrations must be united. Good concentration

coordination is critical for this project so the work breakdown structure includes the concentration unification early in the semester.

The project currently has practically all required resources to implement the design solution at this point. The majority of the work left to do this semester is code that must be written in MATLAB or the QuickControl Software.

3.1 Project Work Breakdown Structure (WBS)

Spring Semester Breakdown – See attached.

3.2 Resource Estimates

N/A.

3.3 Schedule

This section presents the project schedule. Some items are included below.

- Feb 15: Test 1: Compass
- Feb 27: Test 2: GPS
- Feb 29: Deadline for intent to compete in the IGVC.
- Mar 16: Bring concentration work together. All basic components complete.
- Mar 25: Test 3: SICK Laser
- May 15: Spring Project Presentation.
- May 15: Spring Paper/Project Notebooks due.
- May 30: IGVC

3.4 Communication Plan

N/A.

APPENDIX C: PROJECT MANAGEMENT PLAN (SPRING)

Intelligent Ground Vehicle

PROJECT PLAN

08 FEB 2008

1. INTRODUCTION

The purpose of this project management plan is to describe the timeline for work to be accomplished on the intelligent ground vehicle during the spring semester.

1.1 Project Description

The Intelligent Ground Vehicle project is to continue the work done in the 2006-2007 AY and develop an autonomous vehicle that would be able to compete in the annual Intelligent Ground Vehicle Competition. This involves using sensors to detect and avoid obstacles as well as follow a path designated by two lines.

1.2 Project Background

As the Coast Guard begins to update its technology through different initiatives such as deepwater, unmanned vehicle operation is becoming a key development. The purpose of the project is to develop an unmanned, autonomous ground vehicle that can navigate without human aid once given a destination as well as avoid any obstacles that may present themselves while the vehicle is on its way from the starting point finish. Currently, the Coast Guard is developing similar technology for its planned unmanned aircraft. Although there is a difference between the development of a ground based vehicle and an air based vehicle, much of the technology that will allow for navigation and obstacle avoidance is similar. Once Coast Guard unmanned aircraft are developed, the risk involved with dangerous missions will be greatly diminished.

1.3 References

16th Annual Intelligent Ground Vehicle Competition Official Rules.

2. ROLES AND RESPONSIBILITIES

- Sponsors: C2CEN, NAVCEN
- Project Advisors: CAPT Hartnett, Prof. Swazsek, LT Taylor
- Project Team Members: 1/c Seekatz, 1/c Matejka, 1/c Reckley

3. PROJECT DESCRIPTION, SCHEDULE, AND RESOURCES

The work for the project is broken up into three main concentrations: 1) motion control, 2) sensor array with a focus on image processing and lane detection, and 3) the Decision Making Algorithm (DMA). In general these concentrations can be developed in parallel; however, in order to begin advanced testing the three concentrations must be united. Good concentration

coordination is critical for this project so the work breakdown structure includes the concentration unification early in the semester.

The project currently has practically all required resources to implement the design solution at this point. The majority of the work left to do this semester is code that must be written in MATLAB or the QuickControl Software.

3.1 Project Work Breakdown Structure (WBS)

Spring Semester Breakdown - See attached.

3.2 Resource Estimates

N/A.

3.3 Schedule

This section presents the project schedule. Some items are included below.

- Feb 15: Test 1: Compass
- Feb 27: Test 2: GPS
- Feb 29: Deadline for intent to compete in the IGVC.
- Mar 16: Bring concentration work together. All basic components complete.
- Mar 25: Test 3: SICK Laser
- May 15: Spring Project Presentation.
- May 15: Spring Paper/Project Notebooks due.
- May 30: IGVC

3.4 Communication Plan

N/A.

APPENDIX D: FUNCTIONAL REQUIREMENTS DOCUMENT

Autonomous Ground Vehicle

Functional Requirements Document

19 October 2007

1. INTRODUCTION

1.1 Project Description

The Intelligent Ground Vehicle project is to continue the work done in the 2006-2007 AY and develop an autonomous vehicle that would be able to compete in the annual Intelligent Ground Vehicle Competition. This involves using sensors to detect and avoid obstacles as well as follow a path designated by two lines.

1.1.1 Background

As the Coast Guard begins to update its technology through different initiatives such as deepwater, unmanned vehicle operation is becoming a key development. The purpose of the project is to develop an unmanned, autonomous ground vehicle that can navigate without human aid once given a destination as well as avoid any obstacles that may present themselves while the vehicle is on its way from the starting point finish. Currently, the Coast Guard is developing similar technology for its planned unmanned aircraft. Although there is a difference between the development of a ground based vehicle and an air based vehicle, much of the technology that will allow for navigation and obstacle avoidance is similar. Once Coast Guard unmanned aircraft are developed, the risk to human life will be greatly diminished during dangerous missions.

1.1.2 Purpose

The purpose of completing this project is to increase understanding of sensor information processing and decision-making based on the integration of various sensors. This application of electrical engineering is extremely useful in not only the Coast Guard, but also society at large.

This project is also a very public one, and has much potential to reflect positively on the U.S. Coast Guard Academy in public affairs and recruiting. By performing well at the Intelligent Ground Vehicle Competition, either this year or in the future, we continue to bring honor to the Coast Guard and the Academy's electrical engineering program.

1.2 Assumptions and Constraints

For the purpose of this project the following assumptions have been made: 1) that we will be able to acquire Quicksilver Control motors and drivers and that our sensor integration and decision making software can be implemented on the equipment, and 2) that LT Taylor will finish his code governing the operation of the "SICK" ranging laser.

The following constraints have been placed on the project by the rules of the Intelligent Ground Vehicle Competition: 1) must be a ground vehicle, 2) length: min 3', max 7', 3) width: min 2',

max 5', 4) height: max 6', 5) vehicle power must be generated onboard, 6) max speed 5mph, 7) the vehicle must have both a mechanical and wireless emergency stops, and 8) each vehicle will be required to carry a 20lb payload.

1.3 Documents Reference

A) The 16th Annual Intelligent Ground Vehicle Competition (IGVC), Official Rules.

2. SPONSOR NEEDS / OPERATIONAL REQUIREMENTS

The sponsor needs and operational requirements are outlined as the guidelines to the Intelligent Ground Vehicle Competition. To better fulfill the requirements the vehicle must score more points in the various events at the competition. There are four challenges of the competition: 1) Autonomous, 2) Design, 3) Navigation, and the optional 4) JAUS. By performing better at the competition we represent the Coast Guard and the Academy positively and demonstrate a more comprehensive knowledge of sensory integration. All functional requirements have the goal of a higher competition score in mind.

3. REQUIREMENTS TRACEABILITY MATRIX

Level	Need	Functional Requirement	Justification
1.		**Functionality**	
1.1	1.	Hardware restricted speed of no more than 5mph.	Safety for observers if vehicle loses control. Competition requirements dictate 5mph max speed.
1.2	2.	No bigger than 6' high and 30" wide.	In order to maintain adaptability and mobility and allow the vehicle to travel indoors, this size requirement is necessary to fit through doors.
1.3	3.	Emergency stop at least 2 feet high on rear of vehicle.	Safety of observers if the vehicle does not respond to objects appropriately.
1.4	4.	Remote Control Emergency stop.	Users can shut down the vehicle without getting close enough to get injured if the vehicle is out of control.
1.5	5.	Vehicle power must be generated onboard using batteries and motors.	If the propulsion system wasn't independent, the ground vehicle wouldn't be autonomous.

1.6	6.	Vehicle design and implementation must be adaptable and well documented for future teams.	To allow future teams to step in and contribute to the progress of the project expanding its capabilities.
1.7	7.	Vehicle components must be protected against the weather	Vital vehicle components must be protected against weather due to the fact that can be used outdoors.
2.		**Performance**	
2.1	8.	Software must recognize white lines from a camera and direct the vehicle to navigate between them.	For the competition, the course will consist of a marked path. The path will have white lines that lead to the end of the course.
2.2	9.	Sensors and software must detect and avoid obstacles in the vehicle's path with minimal deviation from the intended course.	For autonomous operation, the vehicle must be able to avoid objects in its path. The course at the competition will have cones and other possible obstructions within the white lines.
2.3	10.	When the vehicle is going forward, the vehicle will not be allowed to deviate more than + or – 5 degrees left or right of track.	Motors will not run at exactly the same speed so small corrections may be necessary to keep the vehicle traveling straight. Wheels could also slide in place causing the vehicle to turn when it shouldn't.
2.4	11.	Each vehicle will be required to carry a 20-pound payload	The motors and structure must be capable of holding and moving a weight in order to simulate performing a task. It is also a competition requirement.

APPENDIX E: SUPPORT PLAN

Autonomous Ground Vehicle
DESIGN/SUPPORT PLAN
20 NOV 2007

1/c Brian Seekatz, 1/c Adam Reckley, 1/c Jeff Matejka

Purpose:

The purpose of completing this project is to increase understanding of sensor information processing and decision-making based on the integration of various sensors. This application of electrical engineering is extremely useful in not only the Coast Guard, but also society at large. This project is also a very public one, and has much potential to reflect positively on the U.S. Coast Guard Academy in public affairs and recruiting. By performing well at the Intelligent Ground Vehicle Competition, either this year or in the future, we continue to bring honor to the Coast Guard and the Academy's electrical engineering program.

Design Solution:

The design solution required for this project consists of four main parts, or phases. Because the autonomy of the vehicle removes a human from the navigation equation, the vehicle must have components that closely parallel various parts of a human body. The four phases of design include: 1) the vehicle frame, or the skeleton, 2) motion control, or the muscular system, 3) sensor processing, or sight, and 4) the decision making algorithm, the brain.

The vehicle frame design was a change from the vehicle constructed by last year's group because instead of a relatively wide, four-wheel, "tank-driven" design we switched to a narrower, three-wheel, front wheel driven design.

Last year's design was considered to be "tank-driven" because it had two front wheels and two rear wheels connected to each other on each side by a bicycle chain, much like wheels on both sides of a tank are connected by treads. While this design does synchronize wheel velocity on both sides and provide high stability, it creates trouble when trying to turn on a precisely on a given pivot point. In autonomous navigation, it is critical for the vehicle to know its position at any given time, and after turning the vehicle will assume it is in one given position, but if the pivot point is difficult to calculate or changes for different turns, the vehicle will assume its position incorrectly. To correct this problem, we are switching to a design with two front, motor-driven wheels, and a rear, free-rotating, free-wheeling caster wheel; this design is modeled after working designs from previous Intelligent Ground Vehicle Competitions (IGVC). This design will make it much easier to identify/create a specific pivot point. This design does have the potential to reduce vehicle stability; however, this will not be an issue because the vehicle will still be adequately stable to navigate the relatively flat IGVC course.

The narrower frame design allows the vehicle to fit through a standard size door and opens up the possibility of attempting indoor navigation projects in the future. Last year's vehicle had excessive space for sensors, computers, etc. and there was no reason to make the frame that large.

Next, the design solution requires a solid, reliable motion control system and interface, and this design is also a change from last year's vehicle because we switched from the Robotics First control set-up and "tank-driven" drive train to a QuickSilver Controls motor/controller system and the front wheel drive train. The Robotics First motor/controller system was abandoned for several reasons: 1) there was no supporting documentation concerning PC interaction with the controller, and 2) there was no supporting documentation concerning MATLAB interaction with the code that was uploaded to the controller. We decided on the system provided by QuickSilver Controls because it is currently being used in several successful IGVC groups from various colleges. After purchasing the motor/controller and software from QuickSilver and beginning to familiarize ourselves with the system, it is obvious why winning teams are using this set-up. The QuickControl software is very powerful and does the entire computer control system work for us. With exhaustive command reference and user manuals, the software allows us to program the controller with code to make the motor do practically anything with high accuracy. Initially, the motion control phase of the project looked to be one of the more difficult steps, but after receiving the QuickControl software we are all breathing a lot easier.

Third, the design solution requires sight, or a sensor array. The sensors we will be processing and integrating are: GPS, a SICK ranging laser, digital cameras, an electronic compass, and a shaft encoder. Certain sensors will be used for certain purposes more than others but all must feed into the decision making algorithm in such a way that the correct decision based on the information at hand can be made. For outdoor purposes, like the IGVC, GPS provides constant position information. This allows the vehicle to know where it is, where it was, and where it wants to go, all extremely important information when attempting to navigate autonomously. The SICK ranging laser will be used for obstacle detection. So while GPS provides a specific position "objective", the laser aids the vehicle in achieving that objective as it transits. In order to interpret the SICK laser information we will be using the graduate work of LT Taylor, who is currently studying at University of Rhode Island. The digital cameras will be used primarily for lane detection, which will be a requirement for the IGVC. In order to interpret the information from the cameras we must synchronize the feed from all of them and develop a transform to use the images to detect a lane if it is present. The electronic compass will help with determining if the vehicle is headed in the correct direction, by setting an "objective course" that will lead the vehicle to the desired position directly. When an obstacle is encountered the vehicle course will be modified slightly to avoid the obstacle and then return again to the "objective course". The compass will provide a constant check on the accuracy of the vehicle's course. The shaft encoder will be used to calculate distance traveled by the individual wheels and serves as a check for the other sensors. There are some problems with relying too heavily on shaft rotations as a means of determining distance traveled. If the wheels slip at all, the distance traveled will be incorrect; however, for any projects indoors, this may be the primary means of determining position because GPS will not be an option.

Finally, our design solution will use MATLAB to create a decision making algorithm that will take in all the available information and make navigation decisions accordingly to reach the desired position. MATLAB will interact with the QuickSilver motor/controller system via seven digital I/O pins. Using these input/output capabilities, we can write dozens of basic movement functions using the QuickControl software and then use the MATLAB decision making algorithm to call the various functions by sending the appropriate 7 bit code to the controller. Essentially, the "eyes" see the situation and pass information to the "brain", which then reacts by telling the "muscles" to perform a given action, which, in turn, will move the "skeleton" wherever it needs to go. The algorithm will involve prioritizing obstacle avoidances, along with avoiding obstacles in the "right" way. For example, if an obstacle is avoided by turning around and driving the opposite direction, then little progress will be made towards the primary objective of reaching a given position. Instead obstacles should be avoided in such a way as to continue making progress towards the primary objective.

Support Considerations:

Economic:

The autonomous ground vehicle project has received all of its funding thus far from the United States Coast Guard Academy Electrical Engineering Department. The largest expense that required funding thus far was the purchase of the new motors. The next purchase that will require funding will be a number of web cameras in order to create the aerial view of the robot to track its progress as it moves. Although all of the funding for the project has come from the USCGA Electrical Engineering Department, the project has been supported by C2CEN. C2CEN has provided a new Global Positioning System (GPS) unit specifically for use on the robot.

After meeting with project advisors CAPT Hartnett and Prof Swazsek, the team's projected budget was set at $8000. Since the project is anticipated to take multiple years to complete adequately, the budgeted money does not need to be spent and can be rolled over into next year for next year's team, should they find a use for the money. The autonomous ground vehicle project is the only project of its kind currently being developed at the Coast Guard Academy, so the project does not need to compete for funding against a similar project, and funding should not be a problem, as long as it can be sustained that the project is continuing to benefit the Coast Guard and its missions.

Manufacturability:

With the design completed, our team will physically build the design, with software contributions from LT Everette and LT Taylor. In accordance with our design, there will be three systems that will need to be constructed. The first system that needs to be built is the drive-train system. The drive-train will consist of the wheels, batteries, and recently purchased motors, will provide the power and propulsion that will move the vehicle. The second system that needs to be built is the navigation system. The navigation system will consist of an electronic compass, GPS unit, and SICK Laser Rangefinder. The navigation system, when completed, will provide the vehicle will direction and distances that the vehicle needs to move in order to reach a destination. The final system that needs to be built is the observation system. The observation system will consist of a number of cameras. The cameras will be positioned so as to give an

aerial view of the robot when adjusted and combined into a single image through MATLAB. In addition, the observation system will provide an image from the front view of the vehicle in order to show the limits of the path that the vehicle needs to follow.

The largest concern that is associated with building the design is the interoperability of all of the systems. Each of the systems must complete a specific task in order to make the overall vehicle a success. For instance, the observation system must provide a clear image of both lines that designate the path the robot must follow. Using the information provided by the observation system, the navigation system must then calculate a track to get to a final destination while remaining inside the designated path. Based on the information from the navigation system, the drive-train system must then accurately move specified distances and directions in order to follow the path and avoid any objects in the way of the vehicle.

Sustainability and Reliability:

The largest requirement for spare parts would be the drive-train system. The drive-train system is where all of the moving parts of the vehicle are located. These moving parts will most likely to be the first parts that go bad. In addition, the drive-train system contains the batteries, which provide the power for the entire vehicle. The batteries may lose their ability to hold a charge over time and thus the batteries may need to be replaced. Besides the parts of the drive-train system, the rest of the vehicle should not need much maintenance, since the rest of the parts are electronics, which with proper care should last a reasonably long time. As long as one spare battery for each type of battery on the vehicle is kept as well as an extra tire and chain, the vehicle should be able to operate a reasonably long time without a necessary maintenance.

Due to the fact that the majority of the maintenance will need to be done on the drive-train system, spare parts should be stored at the local unit. The spare parts should be kept and maintained at the local unit, because the parts of the drive-train are crucial to the robot being able to move, and without them the vehicle would be useless. In addition, the parts of the drive-train system, excluding the motors, are low-tech parts that could be fixed by someone without a lot of knowledge about how the rest of the vehicle operates.

Life-Cycle Costs:

As stated earlier, the life-cycle of the design will largely depend upon the batteries and wheels. Over the course of the life of the vehicle, the batteries will need to be replaced whenever the batteries become unable to hold their charge. In addition, the wheels must be replaced whenever the treads on the tires cause the vehicle to lose traction, or if the tire of the vehicle gets punctured and goes flat. The rest of the parts on the vehicle should outlast the parts of the drive-train system, barring any unforeseen events. Over the course of the life of the vehicle, the only support that may be needed is cleaning of the lenses of the cameras, as well as small calibrations to the electronic compass, GPS unit, and laser rangefinder, to ensure that those electronic parts are still working at a precision that will result and accurate navigation by the vehicle. The only special training that would be needed for someone working on the vehicle would be how to account for any errors during the calibration of the vehicle. As stated earlier, the spare parts that should be kept at a local unit include batteries for the vehicle as well as a spare wheel and chain to ensure the vehicle has mobility.

As far as upgrades are concerned, the vehicle should not require any upgrades to maintain its original level of ability. However, if the vehicle is needed to be more precise or follow a narrower path, some of the equipment could possibly need to be upgraded. The GPS and motors would be the most obvious upgrades for better performance. The newer GPS unit would be able to find a position more accurately than the old model and new motors would be able to control the distance that the robot moves more accurately. In addition, the laser rangefinder could be upgraded as well in order to reduce the size and make the robot more aesthetically pleasing.

At first, the majority of the funding would be provided by either C2CEN or the Electrical Engineering Department at the Coast Guard Academy. After the vehicle was completed and put into operation, the cost of maintenance and support would become the responsibility of the local unit, since the parts most likely to require maintenance are easily accessible items.

External Considerations:

Environmental:

The autonomous ground vehicle design will have little to no impact on the environment. The vehicle is driven using rechargeable batteries and releases no harmful emissions into the atmosphere. At the end of the vehicle's serviceable life, the sensors, if still operable, should be removed and reused, the motors and controllers can be reused for educational and testing purposes, and the frame can recycled. There are no special environmental factors that must be considered during the design, test, production or use of our solution. Our solution complies with federal and state regulations because it does not employ any special or potentially harmful technology. All sensors meet regulations and do not pose a threat to the environment.

Health and Safety:

The autonomous ground vehicle will have little or no impact on the user's or the public's health. Much like its impact on the environment, the vehicle does not employ any potentially dangerous equipment and poses no threat to humans. There is no special training required to operate the vehicle; due to its autonomy, it really should not even be "operated" by anyone, except for a remote emergency shut-off switch.

Ethical, Social & Political:

The only social consideration in the design was the potential for an indoor project that could be used to improve the electrical engineering department by designing a vehicle that routed messages or some other simple office task.

Software Considerations:

Privacy:

No private user data will be collected at any time for this project and therefore, no privacy considerations were made in the design.

Security

The vehicle will not be required to store any “secure” data. However, if the vehicle were to be used in a situation where the position it is transiting to is classified, the vehicle’s software would need to be modified to encrypt its objective.

APPENDIX F: DESIGN SPECIFICATIONS

Autonomous Ground Vehicle

Design Specification

04 APR 08

Purpose:

The purpose of this project is to develop an Autonomous Ground Vehicle (AGV) eligible to compete in the Intelligent Ground Vehicle Competition (IGVC) held annually in Rochester, Michigan. The vehicle must compete in two challenges: autonomous and navigation. The autonomous challenge focuses primarily on the vehicle's ability to avoid obstacles and visual boundaries on the ground while navigating from a starting point to an end point. The navigation challenge tests the vehicle's ability to accurately navigate to GPS waypoints while avoid obstacles. The design for this project is the proposed solution to the problems raised by the IGVC that stay within the bounds of IGVC rules.

Design Solution:

The design solution for the AGV is an assembly of several sophisticated systems that as a whole can perform all necessary tasks. The design can be broken down into the following primary areas of design: vehicle structure or frame, motion control, sensor array, and Decision Making Algorithm (DMA). The flow of information between these systems is best given by the following flow chart in diagram 1:

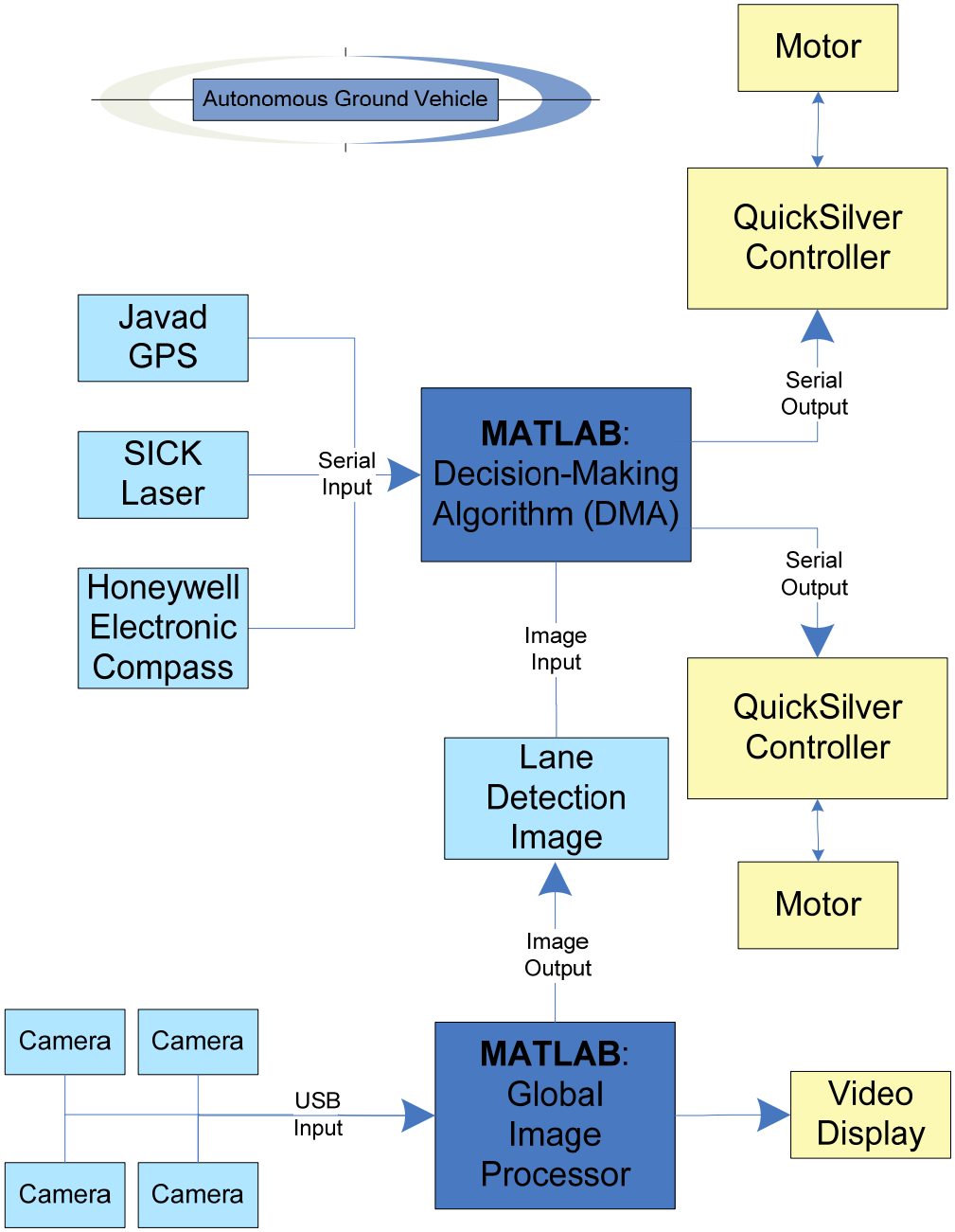

Diagram 1: Information Flow Chart

Frame

The frame is a box-like structure similar to the 2006-2007 design made from aluminum tubing and brackets. It rides on two front wheels and a rear free-rotating, free-wheeling caster wheel. The frame measures approximately 30” across, 48” from front to back, and 66” off the ground.

Motion Control

Two motor/controller systems manufactured by QuickSilver Controls are used to drive the vehicle using gear chain and sprockets to connect the motors to the two front wheels. The motors run on two 12V DC batteries.

Sensor Array

The sensor array consists of GPS, SICK laser, electronic compass, and a camera to provide all necessary sensory functions required by the IGVC.

The IGVC navigation challenge requires that the vehicle travel to a series of positions given in a waypoint matrix and travel within 2m of the desired latitude and longitude. This aspect of the competition requires that the vehicle has some method of validating its position and determining whether or not it has traveled within 2m of the desired lat/long. GPS is the solution to this problem and a Javad Maxor GGDT GPS receiver/antenna was the specific system chosen to provide position validation.

The IGVC autonomous and navigation challenges require that the vehicle detect and avoid obstacles in its path. This aspect requires that the vehicle has some method of detecting obstructions and their position relative to the vehicle for avoidance. The SICK ranging laser is the solution to this problem because it provides accurate bearing and ranges to any objects inside the 180 degrees view in front of the laser.

The IGVC autonomous and navigation challenges indirectly require that the vehicle be able to accurately steer on a desired heading. This requires that the vehicle has some method of validating its current heading. The Honeywell electronic compass is the solution to this problem because it provides accurate heading information for checking current heading against desired heading.

The IGVC autonomous challenge requires that the vehicle detect and avoid boundaries on the obstacle course defined by white lines marked on the ground. This aspect requires that the vehicle has some method of detecting and avoiding visual boundaries on the ground. A digital camera with MATLAB image processing algorithm is the solution to this problem because it can detect the white lines and their position relative to the vehicle for avoidance.

Decision-Making Algorithm (DMA)

The DMA is the most complicated component of the vehicle design because it performs the not-so-simple task of tying all the loose ends together and rendering a navigation decision based on it. The DMA is implemented in MATLAB and reads in all sensory data via serial connections and outputs motion control commands to the controllers also via serial connections. The flow of information during obstacle detection and avoidance in the DMA is depicted below in diagram 2:

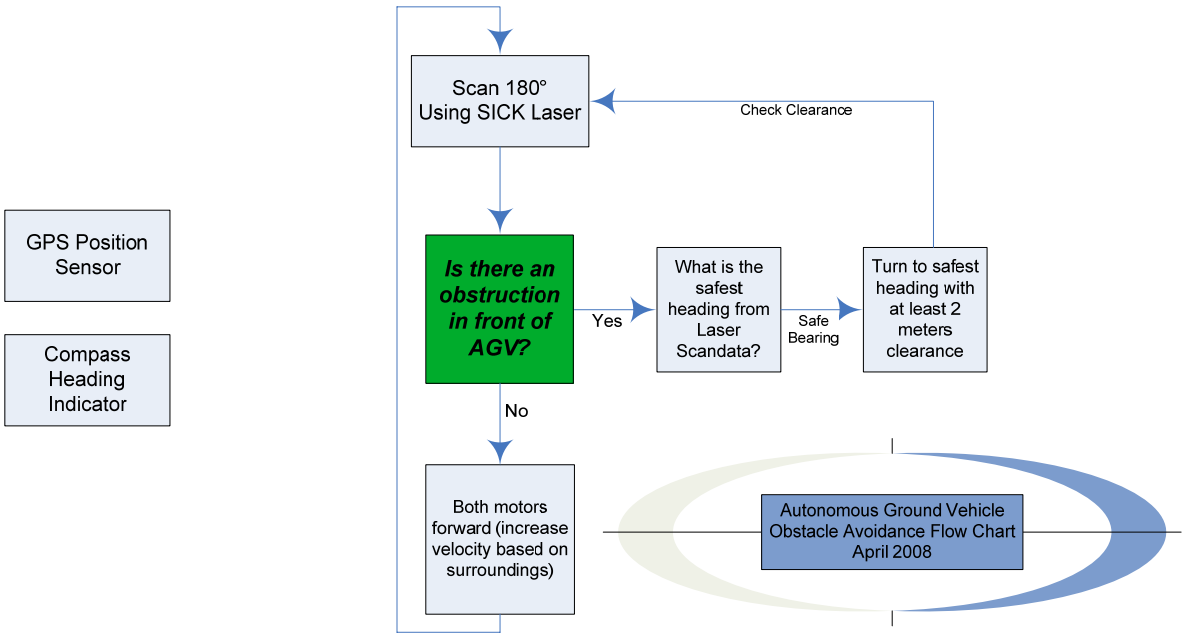

Diagram 2: Obstacle Avoidance

Design Decisions:

The design decisions for the AGV can also be broken down into the four primary areas of design: vehicle structure or frame, motion control, sensor array, and Decision Making Algorithm (DMA).

Frame

The vehicle frame is perhaps the simplest aspect of the vehicle design and is of the least consequence is terms of solving the problem. The 2007-2008 AGV design team received a vehicle frame from the previous design team made of aluminum tubing held together using brackets and screws. The frame designed by the previous team met IGVC standards and was capable of providing a suitable platform for the motion control, sensor array, and DMA systems. On the other hand, the vehicle could not fit through a standard size door which was identified as a requirement by project advisors. Therefore, the frame design had to be modified.

The aluminum tubing is strong and easy to cut and drill, making it easy to modify the old frame design without scrapping it entirely. It was determined that in order to fit the vehicle through a standard size door, it was necessary to cut four inches of each side of each horizontal frame bar running from side to side. It was not possible to simply cut eight inches off of one side of the frame bars because there were screw holes drilled in the front cross bars that were precisely measured to center the SICK laser on the vehicle. So to avoid re-drilling the holes four inches were taken off of either side.

The old frame design involved four wheels attached to the vehicle using two free-spinning axels attached to the vehicle using small aluminum plates with holes bored in them big enough for the axel to slide through. The front axle also had to be resized to fit the vehicle frame size. As for the rear wheels, it was decided that the use of four wheels greatly reduced maneuverability and that a single free-

spinning, free-wheeling caster wheel in the rear would maximize mobility. A small, 5 inch industrial strength caster wheel was installed temporarily to allow motion control testing to proceed near the conclusion of the fall semester. This wheel was replaced by a larger caster wheel with an inflatable tire on it. The larger caster wheel with the tire helped to better solve the project problem because the vehicle will be transiting outdoors on potentially bumpy, unlevel terrain. The larger wheel will prevent the vehicle from getting stuck in ruts or other terrain anomalies while the tire will help to minimize bouncing and increase stability. Increased stability aids the vehicle by making ordered movements more accurate and also by making compass readings more accurate.

The current vehicle design with two front wheels and a rear caster wheel is a design being implemented in several successful vehicles currently competing in the IGVC. The improvement over the previous design is clear; the vehicle can turn in place with ease and performs very accurate heading based turns. The design team feels very good about the design decisions made this year concerning the vehicle's frame.

Motion Control

The motion control design on the AGV could possibly represent the best engineering decision made by the 07-08 design team. The heart of the motion control design is in the motor/controller system being used, made by QuickSilver Controls. The old design implemented a Robotics First motor controller system that is commonly used by the Mechanical Engineering department in their Robotics First competitions. It is reasonable to think that this motor/controller system was chosen by the previous team due primarily to its accessibility. However, no documentation whatsoever could be found for the controllers which effectively renders them as a black box that cannot be edited. This is unacceptable for the design solution. The vehicle must perform a wide spectrum of movements in the challenges and an inability to interface with the controller or edit it will cause failure.

For this reason, the design team began researching new motor controller systems. The best source of information when making an important design decision has always been the design papers of other schools that are winning the IGVC. Several highly successful schools are using the motor/controller system developed by QuickSilver controls. Their package comes with a motor, controller, and all necessary cables to connect them together and to a viable power source, testing software and an exhaustive user manual and command reference. A single package was ordered to test the motor's capabilities and to see whether or not it would fit the requirements of the AGV motor/controller system.

The QuickSilver motor/controller system went above and beyond the required capabilities. It was easy to command the motors using a large variety of parameters such as distance (number of clicks), velocity, acceleration, time, and more. The QuickControl software was easy to use and the user manual was comprehensive and easy to pick up and start reading. The manual contained examples that made implementing basic movements on our vehicle effortless. The reference manual includes a list of all available motion functions and descriptions of each.

Originally the design team decided to interface with the motors using 7 digital input/output bits going from the controller to a digital interface box that converts the connection to a serial port. This solution was viable and performed the necessary functions. However, as the design team became more familiar with the controllers this method of communicating with the controllers was abandoned in exchange for much easier method of sending controller language command lines directly to the

controllers via a straight serial connection. The current motion control set-up performs all necessary functions efficiently and accurately.

Sensor Array

The sensor array in its basic set-up has changed very little from last year to this year. This is because the required sensory information for the IGVC does not change from year to year. The vehicle must travel from waypoint to waypoint (GPS compass) while avoiding obstacles (SICK laser) and boundaries (lane detection). All of these sensors are in place on the vehicle; however, what has changed from last year is some of the actual hardware. The SICK laser and the Honeywell electronic compass are the same units used on last year's vehicle and meet all needs of the problem for the purposes they serve. The GPS receiver and antenna, on the other hand, have been switched out.

The previous GPS receiver and antenna were made by Trimble and boasted several meters of accuracy which may or may not have been accurate enough for success at IGVC. At a point in vehicle development where no one on the design team was looking ahead towards GPS accuracy, a Javad GPS receiver was shipped in by NAVCEN. The new GPS receiver has 20 channels of dual frequency GPS and GLONASS and is horizontally accurate to within 15mm + 1.5ppm x D. The GPS receiver/antenna is the single most expensive piece of equipment on the vehicle (~$12,000) and more than exceeds performance requirements for the IGVC problem.

The previous year's design team knew that lane detection and avoidance would be a requirement but their vehicle never made it far enough for them to look at the problem seriously. This year lane detection capabilities are based off of a single camera mounted on the front of the vehicle using a Hough transform to detect lines in still frame shots. The camera demonstrates the necessary capabilities in basic testing but has not yet been mounted on the vehicle or used for lane detection and avoidance.

Decision-Making Algorithm (DMA)

The vehicle takes a very rudimentary approach to intelligent navigation. The vehicle gathers information about its current situation and its desired position and then calculates a vector (direction and distance) to travel. While traveling on its calculated vector the vehicle continues to scan for obstacles and continues to recalculate its ideal vector. When the vehicle's current heading falls outside an acceptable range of the desired heading, it will stop, recalculate the ideal vector and continue steering on it. When an obstacle is detected the vehicle will stop and recalculate a new ideal vector that avoids the obstacle.

All design decisions made concerning the DMA were safe decisions made for the purpose of maximizing flexibility and the possibility of success at the cost of efficiency. As the AGV project continues to develop, there are and will be countless areas for improvement to make the vehicle more competitive; everything done so far was done for the sole purpose of making it work, not necessarily well. Once the problem is solved, it will be up to the teams of following years to improve upon our methods and develop better solutions.

Requirements:

As described above, the vehicle was design to meet the requirements posed by the IGVC. The vehicle frame is within size requirements and is strong enough to hold the required payload. The motion control system is powerful enough to push the vehicle faster than the IGVC maximum allowable speed and has enough control functions to allow for efficient and precise control of the vehicle in all situations. The sensor array provides all information necessary to make appropriate navigation decisions based on desired position, desired heading, obstacle detection, and boundary detection. The DMA is able to retrieve incoming sensor data, process it, render a decision, and output the decision to the motion control system efficiently. The vehicle performs all necessary tasks to succeed at the IGVC.

APPENDIX G: TEST PLAN

Autonomous Ground Vehicle
Test Plan
March 21, 2008

Synopsis: *Provide an overview of what you are testing and what type of test you are conducting. Testing types include Acceptance, Integration or Unit tests. Refer to your textbook for definitions of these tests. For the purposes of this exercise – you are developing the procedures for an Acceptance Test.*

Personnel:

1. *1/c Adam Reckley – Camera Expert – working on all camera testing*
2. *1/c Brian Seekatz – Autonomous Expert – working on all autonomous testing*
3. *1/c Jeff Matejka – Motion Control Expert – working on all motion control and sensor testing*

Procedure: *Provide an overview of how to set-up the system. This should include a recommended test location, any approvals required before testing can begin and location of off-site or additional resources associated with the test. You should also estimate how long you expect the entire test plan to take (from a few hours to several days or even months).*

Equipment Required:

1. *Vehicle with motors*
2. *Laptop*
3. *MATLAB Scripts*
4. *Compass, GPS, Laser Sensors*
5. *2- 12V Batteries*

Initial Set-up:

The vehicle has all sensors attached and power cables must be connected to start. To connect the laptop to vehicle sensors, use the USB cable from serial port box. MATLAB scripts are on laptop on desktop in the "AGV Control" folder.

Sensor	Required Range Value
GPS	1 meter

Test # 1 – *Motion Control – Turning*

Objective: Ensure vehicle turns to a certain degree.

Requirements: Vehicle must turn to a specific heading within 15 degrees for navigation challenge.

Test Overview: This test will determine if the vehicle can turn to a specific heading or turn a specific number of degrees left or right.

Test Procedure:

1. Turn vehicle on.
2. Start MATLAB with ControlBot.m script.
3. Initialize the motors and compass.
4. Use the radius turn mode (radius) with various degrees left (- degrees) and right (+ degrees).
5. Determine how accurate the turn is
6. Use the compass turn mode (c) with various headings from the compass.
7. Determine how accurate the turn is

Results:

Test # 2 – *Motion Control – Forward and Reverse*

Objective: Ensure vehicle moves a specific distance forward and reverse.

Requirements: Vehicle must drive a specific distance forward to a point within 1 foot.

Test Overview: This test will determine if the vehicle can drive forward and reverse to a specific distance away from the vehicle.

Test Procedure:

1. Turn vehicle on.
2. Start MATLAB with ControlBot.m script.
3. Initialize the motors.
4. Use the forward mode (f, ff, or sf) with various distances forward and reverse in feet.
5. Determine how accurate the vehicle traveled and how many degrees it traveled left and right of the straight line.

Results:

Test # 3 – *Compass Heading Accuracy*

Objective: Ensure the compass is accurate.

Requirements: Compass will indicate an accurate heading.

Test Overview: This test will determine if the compass is accurate indoors and outdoors or under what condition it is safe to use within 5 degrees.

Test Procedure:

1. Turn compass on.
2. Start MATLAB with ControlBot.m script.
3. Initialize the compass.
4. Use the compass monitor mode (comp).
5. Determine how accurate the compass data is push vehicle through different areas including buildings and open fields.

Results:

Test # 4 – *SICK Laser Accuracy*

Objective: Ensure the Laser is accurate.

Requirements: Laser will receive accurate data.

Test Overview: This test will determine if the Laser is accurate indoors and outdoors or under what condition it is safe to use within 1 foot for objects 8 meters away.

Test Procedure:

1. Turn SICK Laser on.
2. Start MATLAB using the laser script.
3. Initialize the Laser.
4. Use the Laser monitor mode (LMS_test.m).
5. Determine how accurate the laser data is by pushing vehicle through different areas including buildings and open fields and measuring distances to objects.

Results:

Test # 5 – *GPS Accuracy*

Objective: Ensure the GPS is accurate.

Requirements: GPS will receive accurate data.

Test Overview: This test will determine if the GPS is accurate indoors and outdoors or under what condition it is safe to use within 1 meter accuracy.

Test Procedure:

1. Turn GPS on.
2. Start MATLAB using the ControlBot.m script.
3. Initialize the GPS.
4. Use the GPS monitor mode (gps).
5. Determine how accurate the GPS data is by pushing vehicle through different areas including buildings and open fields and recording the positions and compare to repeated tests.

Results:

Test # 6 – *GPS-Based Navigation Test*

Objective: This test investigates the vehicle's ability to accurately navigate from some initial position to a given desired position using only an electronic compass and a GPS receiver.

Requirements: The requirement for success at the IGVC is that the vehicle transit to within 2m of the desired latitude and longitude.

Test Overview: Initial conditions required for test: 1)pre-determined lat/lon for desired position and a 2m circle drawn around the position, 2) pre-determined lat/lon for initial position, 3) measured azimuth and distance from initial position to desired position, and 4) start vehicle with vehicle's GPS antenna placed directly over initial position. The test will consist of the vehicle autonomously calculating azimuth and distance from the initial position to the desired position. The vehicle should then turn to the appropriate heading and then travel the appropriate distance.

Test Procedure:

1. Ensure initial conditions are met.
2. Initialize motors, compass, and GPS.
3. Initiate GPS navigation by selecting appropriate function from ControlBot.m script in MATLAB.
4. Add desired position as waypoint.
5. Determine if vehicle (specifically the GPS antenna) is inside the 2m, radius around the desired position.

Results:

Test # 1 – *Lane Detection*

Objective: *The purpose of this test is to determine whether or not the lane detection software provides an accurate range of bearings for the safe travel of the vehicle. The lane detection software must be able to detect lines and provide a range of safe bearings under multiple conditions and line placement.*

Requirements: *Ensure that the camera and function finds an accurate safe range of bearings that corresponds to the heading information and hand-calculated angle.*

Test Overview: *During the test, the vehicle will be positioned so that when an image is taken from the front of the vehicle, it will capture images with a single horizontal line, a single vertical line, two vertical lines and two diagonal lines. By using all of the lane possibilities, the test will ensure that no matter what view of the lane the vehicle has, it will be able to determine the range of safe bearings that the vehicle can transit. The test will be conducted on the football field in order to utilize the accurate and straight lines. The test should result in a safe range of bearings for the two vertical and diagonal lines that is directly in front of the vehicle, while the test for the two single lines should result in a safe range of bearings on the far left and right of the vehicle with directly in front of the vehicle being unsafe bearings.*

Test Procedure:

1. *Position the vehicle on the football field such that the front camera on the vehicle has only a view of a single vertical line.*
2. *Run the MATLAB function to initialize the camera and set up the video input object for the camera.*
3. *Run the MATLAB function for finding the safe range of bearings, titled "SafeRanges(heading,LaneCamera."*
4. *Record the data that is sent to the MATLAB command window and record the heading information (See Data Sheet).*
5. *Save the image taken with the vehicle for later analysis.*
6. *Re-position the vehicle so that the vehicle has a single horizontal line and repeat steps 2-5.*
7. *Re-position the vehicle so that the vehicle has an image of two vertical lines and repeat steps 2-5.*
8. *Re-position the vehicle so that the vehicle has an image of two diagonal lines and repeat steps 2-5*
9. *Using the heading information and the images taken from the vehicle, manually calculate the safe range of bearings by finding the midpoints on each line found in the image. Using trigonometry, find the angle between the midpoints of the two lines. This angle is the safe range of bearings for the vehicle.*

Results:

Data Collection:

Data Sheet for Test #1: Lane Detection

Position	Heading	Lowest Range	Highest Range
Single Vertical			
Single Horizontal			
Double Vertical			
Double Diagonal			

APPENDIX H: COMMENTED DMA MATLAB CODE

```
function evm(Distance1, Acceleration, Velocity, WhichMotor)
% use if you know the distance you want to travel
%
% distance in feet
%
% acceleration in feet per sec2 (1-3 is safe), will deccelerate at same
%           value
%
% velocity in fps
%
% whichmotor is the variable for init_motor function
%  ie motor1 = init_motor('com4'), whichmotor = motor1


dist = ceil(Distance1 * 24000) ; %24000 = 1 vehicle foot
acc = ceil(Acceleration * 5798) ;
vel = ceil(Velocity * 48552000) ;


fprintf(WhichMotor, ['@20 11 25 ' num2str(dist) char(13)]);
fprintf(WhichMotor, ['@20 11 26 ' num2str(acc) char(13)]);
fprintf(WhichMotor, ['@20 11 27 ' num2str(vel) char(13)]);

 fprintf(WhichMotor, ['@20 233 25 0 0' char(13)]);
```

```
function heading = get_heading(serialobj, count)
% For use with the 2007-2008 AGV Honeywell Compass (Black)
%
% Setup:
%       comp=init_compass('com6');
%
% This function returns heading outputed from the compass.
% Example:
%   temp = get_heading(comp, 3);
%     ^                  ^   ^
%     |                  |   |
%     |                  |    - number of headings to average
%     |                  |
%     |                   - serial component after initialized
%     |
%      - current heading from compass
%
%
%

clear hdg;

for n = 1:count
    flushinput(serialobj);
    [data,null] = fscanf(serialobj, '%s');
    [null A null null null null null] = strread(data, '%s %f %s %s %s %s %s',
'delimiter', ',');
    hdg(1,n) = A;
    clear A;
end

heading = mean(hdg);
```

```
function [Latitude, Longitude] = get_position_mod(gps_port);
% For use with the 2007-2008 AGV Javad GPS (green model)
%
% Setup:
%       gps = init_gps('com7');
%
% This function returns the latitude and longitude determined from the GPS.
% Example:
%   [lat lon] = get_position(gps);
%               lat - will contain the latitude
%               lon - will contain the longitude

flushinput(gps_port);

[data,null] = fscanf(gps_port, '%s');

result = strread(data,'%s',-1,'delimiter',',');
% string parsing ;)
lat = char(result(3));
lon = char(result(5));
LatitudeDM = [str2num(lat(1:2)) str2num(lat(3:10))];
LatitudeD = LatitudeDM(1) + LatitudeDM(2)/60;
LatD = char(result(4)) ;
if strcmp(LatD,'S')
    Latitude = LatitudeD * -1;
else
    Latitude = LatitudeD;
end
LongitudeDM = [str2num(lon(1:3)) str2num(lon(4:10))];
LongitudeD = LongitudeDM(1) + LongitudeDM(2)/60;
LonD = char(result(6)) ;
if strcmp(LonD,'W')
    Longitude = LongitudeD * -1;
else
    Longitude = LongitudeD;
end

clear data null

end
```

```
function [scandata time] = get_scan(sick_port)
% For use with the 2007-2008 AGV SICK Laser
%
% Setup:
%       sick1=start_sick('com9',9600);
%
% This function returns scandata matrix of degrees 1-181.
% Example:
%   scan1 = get_scan(sick1);
%           scan1 = the current scan of the area 1-181 degrees in front of
%                   vehicle.
%
%
%
%
%
% This function dumps the current serial buffer to a variable, parses
% a scan output, converts measurement data, and outputs a vector of
% 181 ranges.

tic;
clear scandata;
clear msg;
% Read the contents of the serial buffer
delay(sick_port,0);
if sick_port.BytesAvailable
    msg = fread(sick_port,sick_port.BytesAvailable,'uint8');
else
    msg = [];
end
flushinput(sick_port);

%msg = test;

msg = msg';
bytecount = 369;
for n = 6:(2500-3);
    if (msg(1,n) == 2) && (msg(1,n+1) == 128) && (msg(1,n+2) == 110)
        if (n+bytecount) <= (2500-3);
            scan(1,:) = msg(1,(n+7):(n+7+bytecount));
            break;
        else
            scandata = [];
            break;
        end
    end
end

degree = 1;
for n = 1:2:361
        scandata(1,degree) = scan(1,n) + (scan(1,n+1) * 256);
    degree = degree+1;
end
scandata = scandata';
time = toc;
function motor_port = init_motor(comm_port)
```

```
% For use with the 2007-2008 AGV Quicksilver Motors
%
% Setup:
%           motor1=init_motor('com4');
%           motor2=init_motor('com5');
%
% This function returns the serial device structure for the motors.
% Example:
%       to interact with the motors something like:
%           fprintf(motor1, ['@20 4' char(13)]);
%
%              motor1 = the motor you wish to stop
%
%                 @20 = a group ID for the motors (both are configured
%                       for 20...
%
%                   4 = the command from the Quicksilver command manual
%
%            char(13) = a carriage return (REQUIRED for all
%                       commands to motors)

motor_port = serial(comm_port, 'BaudRate', 57600,'terminator','CR/LF');
fopen(motor_port);

end


% fclose(motor)
% fclose(motor1)
```

```
At_end = 0 ;
first_dead_end = 1 ;
while (At_end == 0)
    try
        scan1=get_scan(sick1);
    catch
    end
    pause(.8)
    disp(['You currently have something ' num2str(min(scan1(75:105))/1000) '
meters away.']);

    if ( min(scan1(60:120)) <= 3000 )
        try
            scan1=get_scan(sick1);
        catch
        end
        pause(.8)
        bearing_difference = -obsheading(scan1, 0, 2500,25)
%         if ( bearing_difference <= -50 | bearing_difference >= 50 )
%             if ( first_dead_end == 1 )
%                 vmi(1,-(min(scan1(1:70))/2300) ,motor1);
%                 vmi(1,-(min(scan1(110:180))/2300) ,motor2);
%                 pause(1)
%                 stopmotor(motor1);
%                 stopmotor(motor2);
%                 first_dead_end = 1 ;
%             end
%             %get another scan
%             try
%                 scan1=get_scan(sick1);
%             catch
%             end
%             pause(.8)
%             bearing_difference = -obsheading(scan1, 0, 3000, 26)
%             bearing_difference_left = -obsheading(scan1, -45, 4000, 26)
%             bearing_difference_right = -obsheading(scan1, 45, 4000, 26)
%             if ( bearing_difference_left <= -45 ) | (
bearing_difference_right >= 45 )
%                 disp('deadendmode')
%                 bearing_difference = -obsheading(scan1, 0, 3500, 15)
%                 bearing_difference_left = -obsheading(scan1, -45, 3500, 15)
%                 bearing_difference_right = -obsheading(scan1, 45, 3500, 15)
%                 if ( bearing_difference_left <= -45 )
%                     stopmotor(motor1);
%                     stopmotor(motor2);
%                     pause(3)
%                     rotationturn(bearing_difference_left,motor1,motor2)
%                     pause(1)
%                 elseif ( bearing_difference_right >= 45 )
%                     stopmotor(motor1);
%                     stopmotor(motor2);
%                     pause(3)
%                     rotationturn(bearing_difference_left,motor1,motor2)
%                     pause(1)
%                 else
%                     stopmotor(motor1);
%                     stopmotor(motor2);
```

```
%                       pause(3)
%                       rotationturn(bearing_difference,motor1,motor2)
%                       pause(1)
%                   end
%               else
%                   if ( bearing_difference_left == 90 )
%                       stopmotor(motor1);
%                       stopmotor(motor2);
%                       pause(3)
%                       rotationturn(bearing_difference_right,motor1,motor2)
%                       pause(1)
%                   else
%                       stopmotor(motor1);
%                       stopmotor(motor2);
%                       pause(3)
%                       rotationturn(bearing_difference_left,motor1,motor2)
%                       pause(1)
%                   end
%               end
%           end
        if (bearing_difference < -15 || bearing_difference > 15)
            stopmotor(motor1);
            stopmotor(motor2);
            pause(3)
            rotationturn(bearing_difference,motor1,motor2)
            pause(1)
%           elseif (min(scan1(55:125)) <= 1500 | bearing_difference < -1 |
bearing_difference > 1 )
%               disp('Dynamic Mode')
%               vmi(1,(min(scan1(1:25))/2300)*.2+(min(scan1(25:70))/2300)*.8
,motor1);
%
vmi(1,(min(scan1(155:180))/2300)*.2+(min(scan1(110:155))/2300)*.8 ,motor2);
%               first_dead_end = 0 ;
        elseif ( bearing_difference <= -1 )
            disp('Dynamic Mode')
            vmi(1,(min(scan1(45:135))/2300)*.88*cosd(6*(bearing_difference))
,motor1);
            vmi(1,(min(scan1(45:135))/2300) ,motor2);
            first_dead_end = 0 ;
        elseif ( bearing_difference >= 1 )
            disp('Dynamic Mode')
            vmi(1,(min(scan1(45:135))/2300) ,motor1);
            vmi(1,(min(scan1(45:135))/2300)*.88*cosd(6*(bearing_difference))
,motor2);
            first_dead_end = 0 ;
        else
            vmi(1,min(scan1(45:135))/2300 ,motor1);
            vmi(1,min(scan1(45:135))/2300 ,motor2);
            first_dead_end = 0 ;
        end
    else
        vmi(1,min(scan1(45:135))/2300 ,motor1);
        vmi(1,min(scan1(45:135))/2300 ,motor2);
        first_dead_end = 0 ;
    end
end
```

```
function radiusturn( degrees, motr1, motr2 )
% + degree is to the right
% - degree is to the left
% 26.5inches = radius of vehicle
Circ = 14.33 ;   %pi * 2 * 27.5/12 ;
% pause(1)

if (degrees > 0)
     dist = Circ * (degrees/360);
     motor_drive = 1 ;
elseif (degrees < 0)
     dist = Circ * -1 * (degrees/360) ;
     motor_drive = 2 ;
end

    if motor_drive == 1
        evm(dist,1,3,motr1)
    elseif motor_drive == 2
        evm(dist,1,3,motr2)
    end

    pause(ceil(dist / 3+2))
```

```
function rotationturn( degrees, motr1, motr2 )
% + degree is to the right
% - degree is to the left
% 26.5inches = radius of vehicle
Circ = 14.33 / 2 ;   %pi * 2 * 27.5/12 ;


if (degrees > 0)
     dist = Circ * (degrees/360);
     motor_drive = 1 ;
elseif (degrees < 0)
     dist = Circ * -1 * (degrees/360) ;
     motor_drive = 2 ;
end

    if motor_drive == 1
        evm(dist,1,1.5,motr1)
        evm(-dist,1,1.5,motr2)
    elseif motor_drive == 2
        evm(dist,1,1.5,motr2)
        evm(-dist,1,1.5,motr1)
    end

    pause(ceil(dist / 3 + 2))
```

```
function sick_port = start_sick(commport,baud)
% For use with the 2007-2008 AGV SICK Laser
% UPDATE April 2008: You might have to restart MATLAB after you use this
% command once because the laser may not send until the second time you
% start it.
%
% Use:
%       sick1 = start_sick('com9',9600);




% ---- LT Everette's Code below ----

% This function starts the SICK laser and places it into continous
% scanning mode.

% Construct telegram to request an acquisition
startmsg = [02 00 02 00 32 36 52 08];

sick_port = serial(commport,'BaudRate', baud);
set(sick_port,'InputBufferSize',2500);
fopen(sick_port);

% Flush the serial buffer on the sick port
if sick_port.BytesAvailable
    fread(sick_port,len,'uint8');
end

% Write start telegram to Sick laser to start continous scanning
fwrite(sick_port,startmsg,'uint8','async');
delay(sick_port,0);

end
```

```
disp(wpt_mtx)
    starting_wpt = input('what wpt do you want to start at?', 's');
    end_wpt = input('what wpt do you want to end at?', 's');
    current_wpt = str2num(starting_wpt) ;
    At_wpt = 0 ;
    while (At_wpt == 0)
                % get compass heading & GPS
        try
            temp = get_heading(comp,5);
            [lat lon] = get_position(gps);
        catch
        end
        [rng az] =
dist_to_wpt(wpt_mtx(current_wpt,1),wpt_mtx(current_wpt,2),lat,lon);
        disp(['You are currently ' num2str(temp) ' degrees magnetic. Waypoint
is ' num2str(az) ' degrees true.']);
        azcor = az + 14.3 ;
        disp(['Waypoint is ' num2str(azcor) ' degrees magnetic.']);
        bearing_difference = turn_vehicle(temp,azcor); % check sign
        % radius turn to heading
        if (bearing_difference < -15 || bearing_difference > 15)
            stopmotor(motor1);
            stopmotor(motor2);
            pause(3)
            radiusturn(bearing_difference,motor1,motor2)
            pause(2)
        end
        % drive the distance
        if (rng <= 5)
              stopmotor(motor1);
              stopmotor(motor2);
              current_wpt = current_wpt + 1 ;
              if ( current_wpt == end_wpt )
                    At_wpt = 1 ;
              end
        else
%             At_wpt = 0 ;
             vmi(1,3.5,motor1);
             vmi(1,3.5,motor2);
        end
        disp([num2str(rng) 'ft, ' num2str(az) 'degrees to wpt #'
num2str(current_wpt)])
    end
```