2006-635: MOBILE ROBOTS CAPSTONE DESIGN COURSE

Fernando Rios-Gutierrez, University of Minnesota-Duluth
  He received his M.S. and Ph.D. degrees from Tulane University. Currently, he is an Assistant Professor at the Electrical and Computer Engineering department of the University of Minnesota Duluth. His main areas of interest include robotics, electronic instrumentation, learning techniques for robots, digital systems, and microprocessor applications.

Rocio Alba-Flores, University of Minnesota-Duluth
  She received her M.S. and Ph.D. degrees from Tulane University. Currently, she is an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Minnesota Duluth. Her interests include control, robotics, image processing, digital systems, and microprocessor applications.
Mobile Robots Capstone Design Course

Abstract

This work describes the educational experience gained during the "Design Workshop", a fourth year course in the undergraduate Electrical and Computer Engineering program at our University. The main topic of this course is concentrated on a team-based, semester-long project in which students design and build mobile robots for different applications.

1. Introduction

The number of electronic systems used in robotics, industrial automation, and other control systems continues to increase dramatically. These systems typically include subsystems with separate processors or controllers. The processors must communicate to coordinate their activities. For example, a typical autonomous navigation system (mobile robot) consists of an interconnected collection of processors connected to a real-time controller. As these systems become even more complex, the need for teamwork becomes even more critical.

A mobile robot is a system that contains mechanical and electronic parts that can be programmed to perform some specific functions, responding to sensory inputs under the control of an internal or external computer. The reasons to use mobile robots as the main topic for the Design Workshop is that in addition to involving the electrical and mechanical engineering disciplines, robotics deals with other sciences and humanities subjects, such as animal and human behavior imitation, learning techniques, and environment interactions. Robotic systems can relate to most processes in nature and human behavior. Because of this, their potential as educational tools for teaching and learning various subjects in technology and sciences is unlimited.

The design and implementation of an autonomous navigation vehicle requires a broad knowledge in areas traditionally not covered in a single discipline. These areas include electrical and computer engineering, computing sciences, mechanical engineering, and other engineering disciplines. As a result, it is very difficult to train students and engineers within a single discipline to effectively design and implement complex mobile robots. Thus, we felt that it was important to offer a senior design workshop to establish an interdisciplinary framework to teach the basics and offer a structured course for education in mobile robot design. One of the major goals of this new class is to expose students to industrial and commercial quality design, and bridge the gap between conceptual understanding and concrete implementations. After undergraduate students are able to apply abstract knowledge in concrete implementations, subsequent higher-level, theory-oriented courses have more relevance.

In this paper the authors present their experiences in using robotics in a one-semester capstone senior design workshop, with focus in interdisciplinary interactions and teamwork for the design and implementation of an autonomous mobile robot that is able to participate in the International Ground Vehicle Competition (IGVC).

The paper provides motivations and background information, describes the senior design workshop organization and the autonomous vehicle characteristics, the paper concludes with a summary and recommendations for future work.
2. Motivations and Background.

Traditional approaches to system design in engineering disciplines have focused primarily on hardware design, whereas computing sciences have focused primarily on software design. With the introduction of robotic systems, it became possible to provide students with hands-on laboratory experiences to construct interdisciplinary and more complex systems. As robotic systems have evolved in research and commercial applications, the number and complexity of these systems has also increased. A significant portion of the design process must now focus on the integration of hardware and software. However, most senior design courses still emphasize just on the software writing or the hardware construction parts. In order to address both software and hardware issues, it becomes essential to apply a team-based approach.

Applications of robotic systems usually involve a large number of various types of sensors and actuators connected to a real-time controller. The rapid increase of such applications requires in-depth research to correctly interface multiple sensors and actuators. These applications serve as excellent case studies to motivate students and teachers. Also, fast computation speed is a major barrier for many real-time sensing and control applications, especially for sensors requesting a large amount of computation, such as obstacle detection and image sensors.

A real-time controller is important in applications where multiple subsystems are developed, each containing a large number of sensors and actuators. Modern robotic systems are equipped with advanced, microprocessor-based embedded systems. For example, an intelligent navigation system may use several electronic control units and hundreds of sensors and actuators to monitor and improve its performance. With the rapid development of new technologies for precision navigation, more sensors and actuators with sophisticated control algorithms are added to the system. This requires more complex and reliable communications techniques. Many of these sensors may be linked with a real-time network to log sensory data and provide feedback for real-time control.

Robotic systems that are designed for real-time applications such as navigation systems and process automation are very expensive to develop. To be practical for a senior design class, the per-unit cost must be strictly controlled to fit within a typically constrained laboratory budget, since the cost of development of a platform for a mobile robot can become fairly expensive as the complexity of the sensors and control system are increased, it can reach thousands of dollars. In our case, early in the development process, this was a limitation that we had to work with. First, we try to establish which would be the best approach to follow in order to keep the cost of the robotic platform within the limits of the budget assigned to this class. Next, in order to reduce the implementation costs we decided to use as much devices and systems that we already had available in our labs, such as sensors, electronic devices, laptops, microcontroller cards and debugging, simulators, programming, and analysis software tools.

In this project, we integrated a Digital Image Processing program into a real-time control system in order to accomplish the fast image processing required to control the navigation of the robot. The image processing program developed for this project was able of processing images covering a sufficient width for the 5’ by 2’ mobile robot, at a processing speed of 2-5 image frames per second. This allowed a 4-5 mph ground velocity for the mobile robot. In order to accomplish other of the navigation requirements, we added and implemented a field-level Geographic Positioning System (GPS), which integrated multiple reference points, with real-time positioning signals received from a GPS device. Such a system can be used for many real-time
field applications, including yield mapping, and programmed navigation. Also, in our case, all sensors, controllers, and actuators were networked together using a real-time controller implemented with a microcontroller card.

The Senior Workshop class consisted of a group of 16 students that were separated into four independent design teams: Two six-student vehicle design teams, and two two-student support teams. Each vehicle team’s goal was to design, implement, and test an autonomous vehicle that was able to navigate in an unknown environment. The two support teams were in charge of designing the GPS and the power supply systems that were used for both vehicles. The membership selection to a particular subteam for each student was based mainly on their academic major and personal preference, or in case of conflict the advisors had the final decision. Overall, the design groups received supervision from three faculty advisors. One advisor supervised the design of the power supply and drive motors systems, the second advisor supervised the design of camera and image processing systems, and the third advisor supervised the design of sensors, navigation controller and GPS systems. The following block diagram illustrates the vehicle’s basic architecture.

![Vehicle Block Chart](image)

For each vehicle, three two-student design sub teams were formed. a) Vehicle design and modification team was in charge of the mechanical base modifications. The wheels, stability, motors, and other vehicle components were altered and/or fabricated by this team to ensure that the vehicle is capable of performing the desired tasks. b) The video capture processing and ultrasonic obstacle detection team was in charge of the video camera and sensors interfaces that were attached to the vehicle to provide environment information. c) The navigation team was in charge to combine the information provided by the cameras and sensors to control the vehicle with the given data, using various algorithms implemented in Matlab, to control the vehicle.

Each design team named a team leader. The team leaders were selected by their peers based on their leadership, and managerial skills. The team leader was the main communication channel
between the students and the faculty members supervising the workshop and the subteams leaders named for each vehicle.

The workshop was organized as a 16-week long project. The faculty members had weekly meetings with the students. In weeks 1-4 students received 3-hour lectures related to the theoretical background needed for the design of the different components, and 3-hour seminars related to the organization and development of the vehicles. During weeks 5-8 students dedicated full time to the development of their particular project. In week 9 each subteam presented preliminary oral and written reports in which they described the advances and problems they had in their projects. Weeks 10-15 were dedicated to debugging, testing, interconnecting and final assembly of the vehicle. In week 16 final oral and written reports of the completed vehicles were presented.

4. Workshop Educational Objectives:
This course fulfills the departmental requirement for a capstone senior design project, by providing a one-semester workshop alternative to the normal two-semester senior design option. The content of the course combines departmental expertise in digital system design, digital signal processing, power systems and control. Based on ABET’s educational outcomes, upon completion of the workshop students were able to:
- Complete a design project that is interdisciplinary in nature, integrating the knowledge obtained in previous ECE classes
- Accurately communicate his/her project results, both in written report format and in oral presentation format
- Understand how teams work and how to interact in a team setting. (Understand what is like to work in industry)
- Appreciate the role of engineering in society, so that students take into account environmental, economical, social and ethical issues that are important in the development of an engineering project.

5. Project Budget and Administration.
In Table 1, we present a full list of parts and cost for each vehicle. Originally the budget assigned to the workshop was $3000. The cost of the parts purchased for each vehicle was in the order of $1450, and as shown in the table this cost does not include the costs of parts that we already had available in the department such as the laptops and software packages. Taking these expenses into account the actual cost increases to about $3500 for each vehicle. One faculty member was in charge of administrating the budget. The main challenge in managing the budget was to find the appropriate parts to be used in the project that were in the price range that we could afford.

6. Workshop Assessment.
The projects were evaluated in several stages, in a gradual and continuous way. In the weekly meetings, each team presented the evolution of their projects and received orientation from the instructors. The objectives of these weekly meetings were also to have a close observation of the teams’ progress and assure that each team member contributed to the teamwork. During the ninth week, 30% of the final grade was assigned, after the students presented the preliminary written report and oral presentation of the results in their progress. Another 40% of the final grade was assigned to the students during week sixteen when they demonstrated that their
projects were working in accordance to the specifications. The last 30% of the final grade was assigned based on the final oral presentation, taking into account the quality and clarity of the presentation, and the completeness of the final written report.

Next we present a description of each of the components that were used to implement the autonomous navigation robot.

Body Design and Mechanics Modification: The autonomous vehicle base came from a Power Wheels Chevrolet Silverado pickup truck (shown in Figure 2). A majority of the original vehicle parts were used: batteries, battery chargers, wires and plugs, chassis, hood, motors, wheels. The final modified autonomous vehicle is shown in Figure 3.

It was originally decided that all components except the sensors, camera, GPS, computer, and load, would be mounted inside the robot. This provided the most environmental protection to some of the more sensitive equipment. The computer was mounted in the bed of the vehicle while the cameras were mounted above the vehicle, slightly over three feet off the ground. The GPS was mounted just behind the camera, and the emergency stop was mounted behind the GPS. The sensors were directly attached to the front and rear bumpers of the vehicle. To make the autonomous vehicle weatherproof, all sensitive components that did not need to be outside were placed inside the body. A single poly-sealed piece of particle board covered the essential components, and plastic hoods protected the camera, computer, and sensors.

Due to the Power Wheels’ simple design, the mechanical drive and traction needed to be modified. It was decided to use large rubber bands which wrap around the tires to provide traction on smoother surfaces. Unfortunately this did not provide enough traction in grass or gravel, so 20 soft golf spikes were inserted into the rear tires. With these changes the vehicle had better traction on most surfaces.

The original plastic chassis was not designed to support the load required by the vehicle with all its necessary components. Reinforcing studs were placed along the runners which aid in supporting the weight placed in the middle of the vehicle. With these modifications the vehicle
was able support roughly 200lbs before showing any signs of stress, and was very stable during turns.

**Motors and Motor Control:** Using the stock Power Wheels motors made design simpler, but the motors themselves needed to be mechanically stabilized. Because they were designed to reduce whiplash when children are riding on the vehicle, the motors themselves sat loosely on the chassis allowing for a less drastic acceleration. In order to fix this, wood was molded around the motors to keep them completely still.

The direction and speed of the motors were controlled by two signals going into the H-bridges. One signal was used to control the speed of the motors by a pulse width modulation signal coming off from one of the microcontroller ports. The other signal was a directional signal used to control the direction of the motor, either forward or reverse. The microcontroller received a 2-bit signal from the controller to indicate direction. Slow down and speed up procedures were implemented in order to safeguard the H-bridges from overheating.

**Steering System:** Quite possibly, the most complex mechanical task in modifying the autonomous vehicle was the steering. The original vehicle used Ackerman steering which is very similar to the steering in automobiles. The first step in developing a functional steering mechanism was to redesign the tie rods. The original tie rods were made of weak nylon and showed signs of wear before ever being used. Teflon, which has a tensile strength much greater than nylon, was used to craft the new tie rods. They were elongated to provide a much greater turning radius as well. The original tie rods provided a turning radius of only 25°, where the newly designed tie rods provide a turning radius of 43°.

The next steering challenge came when deciding how to actually move the wheels. It was decided to use bearing style eye joints, very similar to actual tie rod ends, to connect from the newly designed Teflon tie rods to a steering column which was pushed up through the top of the vehicle. With access to the steering on top of the vehicle, the challenge had been reduced drastically. A servo motor was mounted several inches away from this column and cogs were placed on both the servo armature and the steering column. A small chain was then used to connect the two cogs. A safety mechanism was implemented in order to shut the servo motor off when the wheels reached a certain position because the servo motor had enough force to tear the robot’s plastic chassis apart. The relay circuit developed prevented the servo from running if the steering makes contact with micro-switches mounted underneath the robot. Some details of the modifications performed to control the steering of the vehicle are shown in Figure 4.

**Digital Compass & Steering Control:** An important action of the steering control is to make sure that the vehicle is traveling in the desired direction. If the vehicle is not turning and veers slightly off the desired heading, the controller tells the steering drive to turn in order to put the vehicle in the correct heading. The main purpose of the digital compass was control the movement of the vehicle by telling the microcontroller the direction that the vehicle is currently moving. The compass used (CMPS03) must be calibrated to the magnetic north, so data obtained from the compass is correct.

The main information for the steering is obtained from the compass. The microcontroller takes data from the compass and sends it to the controller. Then, the controller analyzes the data and
tells the vehicle to go straight, turn the wheels to the left, or to the right. The navigation algorithm says how much to turn from the current heading. From this value, the controller calculates the desired heading by adding the turn angle to the current heading. Also, the turn angle indicates what direction the wheels need to turn.

![Figure 4: Modified Steering System](image)

**Wireless Drive, Remote Kill & Emergency Stop:** The autonomous vehicle is heavy, so a wireless driving system was implemented. A wireless controller was stripped out of a toy RC car. The robot’s power system has a latching relay which takes a pulse to stop and disconnect the power to the driving motors in case of an emergency. Since the robot will never be driven wirelessly and remote killed at the same time, it was decided to use the same circuit. An industrial emergency stop button was added to the robot. The button is a push to break style button. Pushing down the button cuts power to the drive motors and instantly stops the robot.

**The Power Supply Module:** Since the electrical components of the vehicle all require specific amounts of voltages, currents and power, a power supply module (power supply circuit board and the power supply box combined) was constructed to provide these power requirements. The input to the module came from a pair of 12-volt batteries. The H-bridges that drive the DC motors required 12 volts, the microcontroller board and each ultrasonic sensor required 5 volts, and the GPS required 3 volts. The video camera and the laptop were powered using their own battery. The Power Supply Module also received a control signal from the emergency stop button.

**Connection between Microcontroller and Computer:** The connection between the microcontroller, 68HC12 MiniDragon development board, and the computer was implemented using the computer’s parallel port. The parallel port was connected to one of the user programmable header on the microcontroller using a ribbon cable. In Matlab, the Data Acquisition Toolbox gives the user the ability to communicate with the parallel port. By connecting the computer to the microcontroller, the navigation algorithm can use the data that comes in from the sensors and the compass to help control the steering of the vehicle.

**8. Navigation Controller**
The navigation controller is the brain of the autonomous vehicle; it takes inputs from the obstacle sensors, video camera, compass and GPS in order to decide the movements of the vehicle. Next we give a description of the operation of this system.
Obstacle Sensors: The purpose of the ultrasonic sensors is to gain information about how close a physical object is from the vehicle. Also, by using multiple sensors and the parallax method, it is possible to determine how far an object is from the center of the robot. It was determined that three sensors in the front and one in the rear of the vehicle will be sufficient to help in the control of the navigation. The three sensors (SRF 04) in the front are the main source for detecting the location of close physical obstacles. In arranging the front sensors like this, the desire was to gain information not only about how far away each obstacle is, but also an approximation on the location of each obstacle. The first way this is accomplished is if there is an object detected on one or two of the sensors and not on the other(s). For example, if an object is detected on the left and center sensors, but not on the right, then the object is located to the left of the center of the vehicle. The next method uses parallax to find the horizontal offset from the center of the vehicle is obtained. If the vehicle ever needs to backtrack, the rear sensor, which is located on the center of the rear bumper, will give useful information about distances to objects located behind it.

Video Camera, Image Processing: Using Matlab, the team was able to implement an image processing algorithm to determine which pixels of an image obtained from the video camera had obstacles in them and which ones had course-lines. First, the program called a simple method that captured a picture from the camera to have it available for manipulation. Next, the image was processed to find out which pixels contained obstacles. The only obstacles present in the course are orange cones and barrels, some of them with white stripes on them (Figure 6). Color detection techniques were applied to filter out all but the orange from the image, which left only the cones and barrels in the picture. The next step was to filter out all but the white from the picture which left the outlines of the course-lines. Added together, these two images would give the pixels of all objects to be avoided. Next, the image was transformed to a binary one, where all obstacle pixels and course-lines were white, and all safe areas were marked black (Figure 7). This, along with the information provided by the compass and ultrasonic sensors was supplied to the navigation system.
The navigation algorithm looks at the data given to the controller by the various components to determine in which direction to move. The main component used to navigate the vehicle is the camera, the compass and sensor information are used to help in the control of the navigation. The navigation algorithm first gets the black and white image developed by image processing that shows the cones and lines as white spaces. The algorithm then looks forward to see if the area is clear. The vehicle is 32.5” wide, so there needs to be an area that big for the vehicle to pass through. The area for the vehicle to pass through will decrease towards the top end of the image because the camera sees a larger width, therefore making the width of the vehicle a smaller proportion of the image. Figure 8, shows the area of the image that is checked to see if the path is clear; the black pixels represent the area to be checked. The black area on the image is the area that needs to be checked if the vehicle is going to go straight. Every time a white pixel is seen in the image a counter is incremented. After all the pixels are checked, the algorithm looks at the number of white pixels in that area. If the number of white pixels is less than a threshold value (which is currently set to 50), the algorithm tells the vehicle to turn 0°. The threshold value is used because there will be excess white in the image that does not denote an obstacle or line.

9. Navigation Control Selection
In the IGVC robotic contest, there are two challenges in which the robot can compete. One is called the Autonomous challenge and the second is called the Navigation challenge. For the Autonomous challenge, the image processing is the main input for the algorithm. The first thing the algorithm does is to check the image to determine the desired turning angle. While the vehicle is turning, the algorithm checks the sensors to see if there is anything in the way of the vehicle. After the vehicle gets to the desired angle, the vehicle will get a new image and determine the best angle to turn at.

For the Navigation challenge portion of the competition, the GPS is the main input for the navigation. In this project the Garmin eTrex Legend unit was used to get this input. From the GPS, the navigation algorithm obtains the desired angle the vehicle needs to travel in to get to the waypoint. Using this angle as a desired heading, the navigation algorithm determines what angle the vehicle needs to turn to get at the desired angle according to the GPS. Using that angle as a starting point, the image is analyzed to see if that angle is clear. If the path is clear, the vehicle turns at that angle. If the path is not clear, the navigation algorithm looks for the closest angle to the angle given by the GPS that gives a clear path. Once that angle is found, the vehicle turns at that angle. While the vehicle is turning, the algorithm is checking the sensors to see if there are any objects in the way. The algorithm also constantly checks the current GPS location to see if the vehicle is at the desired waypoint. If the current GPS location does match, the algorithm sets the desired location as the next waypoint.

Combining the various detection components into this navigation system provides the autonomous vehicle with the information needed to properly navigate through the unknown environment. The system will examine the data and come to an intelligent decision of which direction to take. Once all components of the vehicle have been installed and are working individually, they must all be integrated into a single system that will control the overall operation of the vehicle. The navigation control program is written in Matlab and uses the information provided by each input to determine the autonomous vehicle’s direction.
10. **Global Positioning System**: The Global Positioning System (GPS) is a satellite-based navigation system made up of a network of 24 satellites placed into orbit by the U.S. Department of Defense. The accuracy of new handheld GPS receivers without a differential correction is about 15 meters on average. The error is due to a combination of several factors including: orbital errors, satellite clock errors, errors in the speed of the signal through the ionosphere and the troposphere, receiver noise, and multipath errors. This gives the receiver an erroneously long time measurement. One way to correct or eliminate some of these errors is to use what is called the Wide Area Augmentation System (WAAS)\(^9,10\). It uses 25 ground based stations positioned in the United States to monitor errors in satellite orbit, clock drift, and signal delays caused by the atmosphere and ionosphere. GPS receivers that use WAAS correction can obtain an accuracy of less than 3 meters, 95 percent of the time.

A controller implemented in Matlab, is used to control the communication between the PC and the GPS unit and to process the received data. The GPS program developed also implements a function to convert latitude and longitude to Universal Transverse Mercator (UTM)\(^11\). This allows the navigation program to work with distances and directions in X-Y coordinates. The UTM system applies the Mercator projection to mapping the Earth. The map provides low distortion at points near the tangent meridian (also called the central meridian). Choosing points farther away from the central meridian will result in higher distortion. To provide a low distortion map of the whole Earth, the UTM defines 60 different standard projections. Each projection is centered on a meridian that is 6 degrees from the next central meridian. The projections are called UTM zones. For example, Duluth, Minnesota is in zone 15 and Traverse City, Michigan is in zone 16. The central meridian of zone 15 is 93 degrees, and the central meridian of zone 16 is 87 degrees. Any point on Earth is always within 3 degrees of the central meridian of its zone.

11. **Conclusions**

Two autonomous vehicles were designed and built for the specific task of participating in the IGVC, and to fulfill the senior design requirement for the students in our department. The participation in this project gave students real life team work experience. They experienced the
application of theoretical information in different areas of knowledge to solve real life problems. This experience could later be used in their professional careers to solve similar problems in numerous other applications. The potential of real-world autonomous devices being able to control themselves is growing, and in some cases is very desirable.

Throughout the process of designing and building the autonomous vehicles, the teams encountered many problems and made some mistakes of their own and they had to be realized and acted on accordingly. The top challenge for students and faculty members was to manage the schedule of each subteam project, so that they were all ready to be put together by the end of the tenth week. It would be more desirable to have more time to construct the vehicles. In fact, the IGVC documentation suggests that this project be completed over the course of one year. Based on the time constraints that we had, along with a limited budget, the final result that we obtained were better than expected.

Overall, this project has integrated a variety of electrical, computer, and mechanical engineering techniques, along with computer science and mathematics. It also provided the students with a way to transfer theoretical knowledge into a practical application, been an invaluable final step in the electrical and computer engineering program. The final test of the project was the presentation of the autonomous navigation vehicle in the International Ground Vehicle Competition in Traverse City, MI. In this competition, the team managed to qualify for the finals. However, a computer failure on the day of the final competition prevented the team to get a better result.

References

# Parts List

<table>
<thead>
<tr>
<th>Qty</th>
<th>Item</th>
<th>Description</th>
<th>Price</th>
<th>Price to Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Vehicle Base</td>
<td>Power Wheels Chevrolet Silverado Pickup Truck</td>
<td>$300</td>
<td>$300</td>
</tr>
<tr>
<td>4</td>
<td>Wheel Bands</td>
<td>Rubber bands for wheel traction</td>
<td>$15</td>
<td>$15</td>
</tr>
<tr>
<td>1</td>
<td>Golf Spikes</td>
<td>Rear wheel traction</td>
<td>$10</td>
<td>$10</td>
</tr>
<tr>
<td>A/R</td>
<td>Wood</td>
<td>1x1s and particle board for base support</td>
<td>$15</td>
<td>$15</td>
</tr>
<tr>
<td>2</td>
<td>Tie Rods</td>
<td>Teflon fabricated tie rods</td>
<td>$5</td>
<td>$0</td>
</tr>
<tr>
<td>5</td>
<td>Steering Eye Joints</td>
<td>Bearing style eye joints</td>
<td>$2</td>
<td>$0</td>
</tr>
<tr>
<td>1</td>
<td>Pittman DC Motor</td>
<td>Steering motor</td>
<td>$25</td>
<td>$0</td>
</tr>
<tr>
<td>2</td>
<td>Cogs</td>
<td>From servo to steering joints, with chain</td>
<td>$2</td>
<td>$0</td>
</tr>
<tr>
<td>1</td>
<td>Relay Circuit</td>
<td>Power systems relay circuit</td>
<td>$3</td>
<td>$0</td>
</tr>
<tr>
<td>1</td>
<td>RC Remote</td>
<td>RC remote control and receiving circuitry</td>
<td>$20</td>
<td>$20</td>
</tr>
<tr>
<td>2</td>
<td>Inverting Amp</td>
<td>Inverting amplifier circuits, per direction</td>
<td>$5</td>
<td>$0</td>
</tr>
<tr>
<td>2</td>
<td>Full-Wave Rectifiers</td>
<td>Full-wave rectifier circuits, per direction</td>
<td>$10</td>
<td>$0</td>
</tr>
<tr>
<td></td>
<td>H-Bridges</td>
<td>Power systems H-bridges</td>
<td>$20</td>
<td>$20</td>
</tr>
<tr>
<td>1</td>
<td>Antenna</td>
<td>Antenna for long-distance wireless control</td>
<td>$1</td>
<td>$0</td>
</tr>
<tr>
<td>1</td>
<td>Stop Button</td>
<td>Industrial emergency stop button (red)</td>
<td>$10</td>
<td>$0</td>
</tr>
<tr>
<td>A/R</td>
<td>Plastic hoods</td>
<td>Weatherproofing plastic hoods</td>
<td>$10</td>
<td>$10</td>
</tr>
<tr>
<td>1</td>
<td>GPS</td>
<td>Garmin GPS eTrex unit with serial cable connection</td>
<td>$100</td>
<td>$100</td>
</tr>
<tr>
<td>1</td>
<td>Power System</td>
<td>Power systems + voltage regulators</td>
<td>$40</td>
<td>$40</td>
</tr>
<tr>
<td>4</td>
<td>Sensors</td>
<td>Polaroid 600 Sonar Ranging Sensor</td>
<td>$120</td>
<td>$120</td>
</tr>
<tr>
<td>1</td>
<td>Camera</td>
<td>Sony Digital Camera</td>
<td>$500</td>
<td>$500</td>
</tr>
<tr>
<td>1</td>
<td>Microcontroller</td>
<td>MiniDragon HC12 Development Board</td>
<td>$160</td>
<td>$160</td>
</tr>
<tr>
<td>1</td>
<td>Computer</td>
<td>Laptop with min: 2 USB, 1 Serial Port, 1 Parallel Port</td>
<td>$1500</td>
<td>$0</td>
</tr>
<tr>
<td>1</td>
<td>Matlab Program</td>
<td>Matlab 7.0.1 Program with Image and Data Acquisition Toolboxes</td>
<td>$500</td>
<td>$0</td>
</tr>
<tr>
<td>1</td>
<td>Dead Reckoning</td>
<td>Encoders with Hewlett Packard HEDS 5500</td>
<td>$100</td>
<td>$100</td>
</tr>
<tr>
<td>1</td>
<td>Digital Compass</td>
<td>CMPS03 Digital Compass</td>
<td>$42</td>
<td>$42</td>
</tr>
</tbody>
</table>

**Total Cost** | $3515

**Total Team Cost** | $1452

Table I.- Part List and Cost for the Autonomous Vehicle