

13th Annual Intelligent Ground Vehicle Competition- June 2005



Design Report for:

M A R V I N
O U B O I N A
B T B S T V
I O O U E I
L N T A L G
E O I L L A
M C I T
O U A G O
U S I E R
S D N
E T
D

University of Minnesota Duluth
Department of Electrical and Computer Engineering

TEAM MEMBERS:

John Fredrickson
Brian Linder
Aashish Parmar
Pao Thao
Jon Penning
Tong Moua



FACULTY ADVISORS:

- Dr. Fernando Rios-Gutierrez
- Dr. Rocio Alba-Flores
- Prof. Scott Norr

TABLE OF CONTENTS

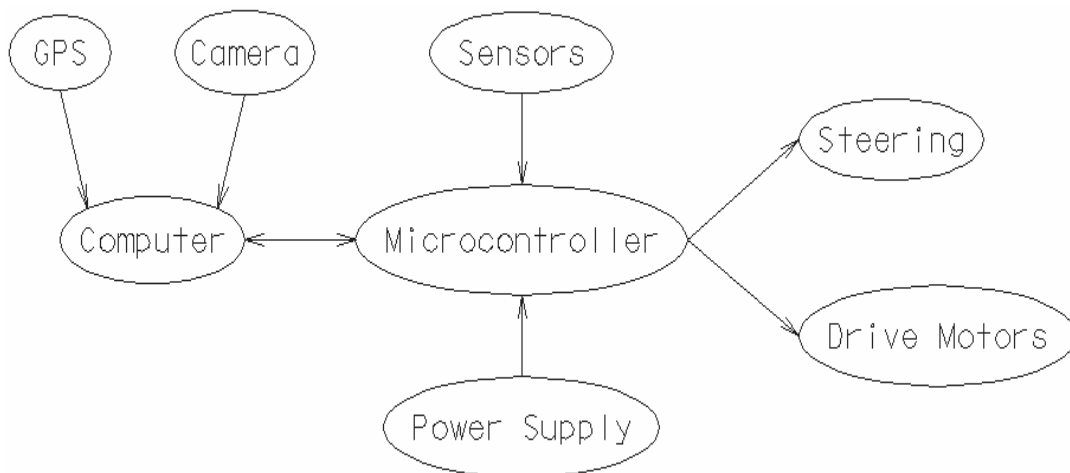
1. INTRODUCTION.....	3
2. TEAM ORGANIZATION	3
3. HARDWARE DESIGN	4
3.1 Robot Structure.....	4
3.2 Steering/Drive System.....	4
3.3 Wheel Design and Motor Control.....	5
3.4 Component Mounting.....	5
4. ELECTRICAL SYSTEM.....	6
4.1 Wireless Drive and Remote Kill.....	6
4.2 RS232 standard voltage and GPS	6
4.3 Vehicle Sensors and Cameras.....	7
4.4 Power Supply System.....	7
5. SOFTWARE AND CONTROL DESIGN.....	8
5.1 Digital Compass.....	8
5.2 Rear Wheel Control.....	9
5.3 Microcontroller and Laptop Interface.....	9
5.4 Steering Control.....	10
5.5 Vehicle Control and Image Processing.....	11
5.6 GPS Navigation.....	13
5.7 Vehicle Technical Characteristics.....	13
6. COSTS FOR THE ROBOT	14
7. CONCLUSION	15

1. INTRODUCTION

The University of Minnesota, Duluth, is entering into its first year at the 2005 Intelligent Ground Vehicle Competition (IGVC). MARVIN (Mobile Autonomous Robotic Vision-Aided Intelligent Navigator) is the result of senior design workshop project, independent study work and research by a team of eleven electrical and computer engineering students. This Intelligent Autonomous Vehicle is able to maneuver in an unknown environment using various methods of self-navigation control. A computer vision component, an obstacle detection component, and live global positioning system (GPS) data have been integrated to develop the robot's navigation system.

2. TEAM ORGANIZATION

The senior design workshop team was divided into four independent design sub-teams. Each sub-team had different responsibilities and was allowed to focus more on their interests and apply their knowledge and skills designing MARVIN. Consistency with the whole design process was highly observed. The sub-teams were structured using the following flowchart of MARVIN's design:



The sub-teams were organized as follows:

Members	Responsibilities	Major
John Fredrickson Siep Joshi	In charge of the vehicle base design	Electrical and Computer Engineering
Brian Linder Robyn Zolen	The cameras and sensors	Electrical and Computer Engineering
Aashish Parmar Paul Crotteau Prab Cheema	Power supply system	Electrical and Computer Engineering
Jon Penning Tong Moua	Navigation system	Electrical and Computer Engineering
Tim Delf Jason Sprenger	GPS system to aid navigation	Electrical and Computer Engineering

3. HARDWARE DESIGN

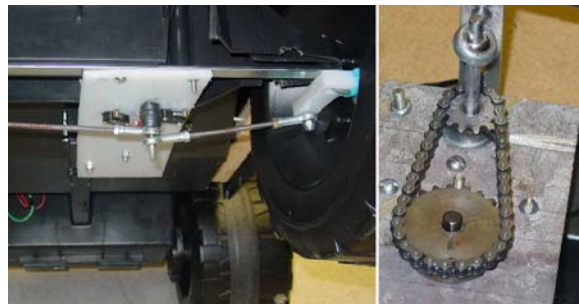
3.1 Robot Structure

MARVIN began as a Power Wheels Chevrolet Silverado pickup truck. A majority of original vehicle parts were used: batteries, battery chargers, wires and plugs, chassis, hood, motors, wheels, and the rear axle. With all necessary components, the Power Wheels plastic chassis is not designed to support the load required. Wooden studs were placed along the runners which aid in supporting the weight placed in the middle of the vehicle. The vehicle can now support roughly 20lbs before showing any signs of stress.

3.2 Steering/Drive System

The robot was designed using Ackerman steering which is very similar to the steering in automobiles. Teflon, which has a tensile strength much greater than nylon, was used to craft the new tie rods replacing the weaker original ones.

They are elongated to provide a much greater turning radius as well. The stock tie rods provided a turning radius of only 25°, where the newly designed tie rods provide a turning radius of 43°. To actually move the



wheels, bearing style eye joints were used to connect from the newly designed Teflon tie rods to a steering column. A servo motor was mounted several inches away from this column and two cogs are connected using a chain. A relay circuit was designed to prevent the servo from running if the steering makes contact with micro-switches mounted underneath the robot.

3.3 Wheel Design and motors

The stock Power Wheels motors made design simpler, but the motors themselves needed to be mechanically stabilized. In order to fix this, wood was molded around the motors to keep them completely still. These motors have 12VDC windings and draw approximately 3Amps at maximum load. They provide adequate power to ensure good ramp climbing ability. The maximum speed of the robot is approximately 4.3 miles per hour thus meeting one of the important IGVC criterion of not going above a 5MPH speed.

3.4 Component Mounting

It was originally concluded that all components except the sensors, camera, GPS, computer, and load, would be mounted inside the robot. This provides the most environmental protection to some of the more sensitive equipment. The computer is mounted in the bed of the vehicle while the cameras are mounted above the vehicle, slightly over three feet off the ground. The Garmin GPS eTrex Legend Unit is mounted just behind the camera, and the emergency stop is mounted behind the GPS. The sensors are directly attached to the front and rear bumpers of the vehicle.

It was determined that three sensors in the front and one in the rear of the vehicle will be sufficient for a number of purposes. The three sensors in the front will be the main sensors for detecting the location of physical obstacles. There will be one center sensor aimed forward and one sensor on the front left and one on the front right pointing slightly to their respective sides. 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.

4. ELECTRICAL SYSTEM

4.1 Wireless Drive and Remote Kill

MARVIN is heavy, so to get on and off the field he uses wireless driving. A wireless controller was stripped out of a toy RC car. The controller takes in 5V and outputs $\pm 5V$ depending on direction: left/right or forward/backward. Using a simple inverting amplifier and a full wave rectifier, a directional signal and PWM signal, respectively, are sent to the H-bridges mounted to each drive motor and the steering servo. With these signals, the H-bridge outputs the appropriate voltage to drive the motors.

4.2 RS232 standard voltage and GPS

The GPS unit chosen was the Garmin eTrex Legend. This decision was based on the unit's low cost and reliability. It is waterproof, is also able to track 12 satellites in parallel and has WAAS capability. More importantly, this unit could be easily interfaced with the Matlab program, providing direct data to the vehicle's navigation system. Matlab is used to control the communication between the Laptop and the GPS unit and to process the received data. However, before the Matlab program can be explained, a description of the communication protocols is needed.



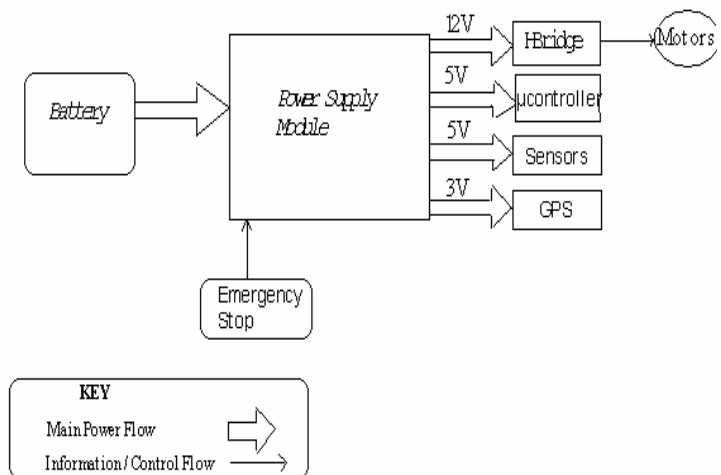
The physical protocol layer used by Garmin is based on the RS-232 standard. However, it does not meet the RS-232 standard for voltage levels. The device only transmits positive voltages whereas the RS-232 requires positive and negative values. In addition, the voltage swing between mark and space does not meet the strict RS-232 standard. However, the voltage characteristics are compatible with most hosts, and it was found that these voltage characteristics were still compatible with the PC used. The application protocol layer is Garmin's own proprietary protocol. All data is transferred in byte-oriented packets with little-endian byte ordering.

4.3 Vehicle Sensors and Cameras

To allow time for calculation, stopping, and turning, the Polaroid 600 Sonar Ranging Sensor was chosen as the main ultrasonic sensor for the vehicle. These sensors receive their operating power from the power supply module (described below). The Creative Instant Web-Cam gets its power from the Universal Serial Bus port on the Laptop. The laptop gets its power from its own battery.

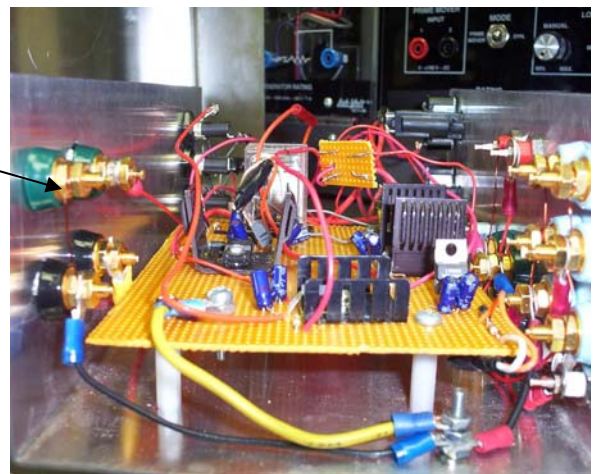
4.4 Power Supply System

Power supplies are a critical component when it comes to designing electronic devices, especially automatic vehicles. Therefore, it is very important not only to be able to build a power supply that could supply sufficient energy to run the components of the vehicle efficiently, but also to prevent dangerous power surges.



The input to the module comes from a 12-volt battery. As shown in Figure 4.1, the voltage must be divided up to satisfy the voltage requirements of each component. The H-Bridge that drives the DC motors and the miniDragon 68HC12 microcontroller board require 12 volts, the ultrasonic sensor require 5 volts, and the Garmin Global Positioning System (GPS) requires 3 volts. All voltage requirements are direct current voltages.

The Web-Cam and the laptop were powered using the laptop battery and the USB port on the laptop respectively. The Power Supply Module also receives a signal from the emergency stop button. The power supply box is a casing on which the main power supply circuit board is mounted. The Power Supply Module's size and dimensions are 10x6x5 inches.



A relay is an electrically controlled mechanical device that opens and closes electrical contacts when a voltage (or current) is applied to a coil. A relay provides isolation of control signals from switched signals. OMRON 12V DC LY1 relay is connected to the two H-Bridges and is used as the manual wireless emergency stop.

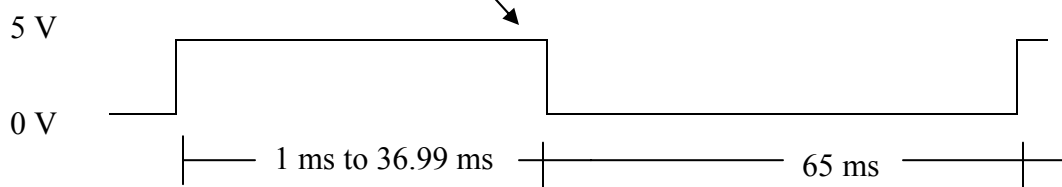
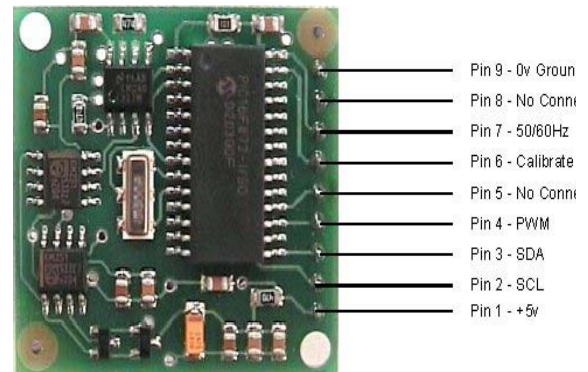


5. SOFTWARE CONTROL AND DESIGN

5.1 Digital Compass

The main purpose of the digital compass is to help control the movement of the vehicle by telling the microcontroller the direction that the vehicle is currently moving. The compass that is used is the CMPS03 which uses the Philips KMZ51 magnetic field sensor to detect the earth's magnetic field. The compass must be calibrated so accurate the data is given. The compass needs to be calibrated to magnetic north, so the data that is read off of the compass is the correct data. The data that comes off of the compass is easy to read. If the compass is pointing at true north (0°), the PWM signal coming off of pin 4 will give pulse with a width of 1 millisecond.

The pulse width will increase until the compass is pointing near to true north (359.9°), which returns a pulse width of 36.99 milliseconds. There is always a 65 ms period in between the high pulses where the signal goes low. The PWM signal can be seen below



The compass uses a 16-bit timer to generate the pulse width which gives a one microsecond resolution (0.01°), but it is recommended to measure the pulse width with a 10 microsecond resolution (0.1°). The pulse width is captured by the microcontroller using the input capture registers, much like how the ultrasonic sensors are used. The timer is captured on the high-to-low and low-to-high transitions. The low-to-high transition is subtracted from the high-to-low transition to give the number of clock cycles

during the pulse width. The difference is then divided by 75 to give a result that is between 10 and 369. Finally, this result is subtracted by 10 (to account for the 1 millisecond offset) to give the result in degrees from 0 to 359.

5.2 Rear Wheel Control

The direction and speed of the motors are controlled by 2 signals going into the H-bridge. The H-bridge converts the signals into the desired voltage to control the motors. One signal is used to control the speed of the motors by a pulse width modulation signal. The other signal is a directional signal used to control the direction of the motor, either forward or reverse. This signal is +5V if the direction is forward and 0V if the direction is reverse. Both the signals come from the 68HC12 microcontroller (miniDragon) ports. The microcontroller receives a 2-bit signal from Matlab to indicate direction. The signals for rear wheel control are summarized as follows.

Drive Direction	Input (from Matlab)		Output (to H-Bridge)	
	PH3	PH2	PP1 (direction)	PP0 (PWM)
Stopped	0	0	Low or High	Low
Forward	0	1	High	High
Reverse	1	0	Low	High

To stop (or slow down) the vehicle, the PWM signal is slowly decremented until the width is zero and to increase the speed, vice versa is done. The slow down and speed up procedures are implemented in order to protect the H-bridges from voltage spikes.

5.3 Microcontroller and Laptop Interface

The connection between the microcontroller and the computer is done using the parallel port. The parallel port is connected to the user programmable header on the microcontroller by using a ribbon cable. In Matlab, the Data Acquisition Toolbox gives a user the ability to communicate with the parallel port. Pins 2-9 on the parallel port can be set either to input or output. Pins 2-9 on the vehicle are set to output. Pins 7 through 9 are used for the select bits, pins 5 and 6 are used to control the rear drive, and pins 3 and

4 are used to control the steering. To get data into the computer, the vehicle uses pins 10-17 for input. In order to send data to the computer, the select bits are used to tell the microcontroller which data to send to the computer. The select bits are connected to the microcontroller on port B pins 5 – 7. The microcontroller periodically checks the select bits to see if it needs to update the data. Since the microcontroller runs slower than the computer, the computer needs to delay in order to read the right data coming off of the microcontroller. 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. The connections between the computer and microcontroller are summarized as follows:

	Microcontroller to Computer								Computer to Microcontroller						
Parallel Port	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3
Microcontroller	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PB7	PB6	PB5	PH3	PH2	PH1	PH0

5.4 Steering Control

The steering is controlled by a servo motor connected to an H-Bridge. The H-bridge receives a PWM and directional signal from the microcontroller. The microcontroller obtains data from the servo-motor's encoder, which is a Hewlett Packard HEDS 5500. The encoder pulses every time it hits a block on its encoder disk and there are 1024 blocks per revolution of the encoder disk.

The main part of the steering comes from the compass. The microcontroller obtains data from the compass and sends it to Matlab. Then, Matlab analyzes the data and tells the microcontroller to go straight, turn the wheels to the left, or turn the wheels to the right. Matlab tells the vehicle to turn the wheels to either turn left, right or stay straight by sending a 2-bit signal to the microcontroller.

The steering signals are summarized below:

Steering Direction	Input (from Matlab)		Output (to H-Bridge)	
	PH1	PH0	PP3 (direction)	PP2 (PWM)
Straight	0	0	Low or High	Low
Turn Right	0	1	Low	High
Turn Left	1	0	High	High

In order to control how far the vehicle turns, Matlab uses the data from the compass to tell the vehicle to turn. The navigation algorithm in Matlab says how much to turn from the current heading. From this value, Matlab calculates the desired heading by adding the turn angle to the current heading and converts it to a value between 0 and 359. Also, the turn angle indicates what direction the wheels need to turn. If the turn angle is greater than 0: turn right; and if the turn angle is less than 0: turn left. For the turns, the vehicle continues turning until the current heading minus the desired heading gets to within 15% of the turn angle. At this point the vehicle needs to straighten out the front wheels, so Matlab sends a signal to turn the wheels in the opposite direction. Another portion of the steering control is to make sure the vehicle is traveling in the desired direction. If the vehicle is not turning and veers slightly off the desired heading, Matlab tells the microcontroller to turn in order to put the vehicle in the correct heading.

5.5 Vehicle Control and Image Processing

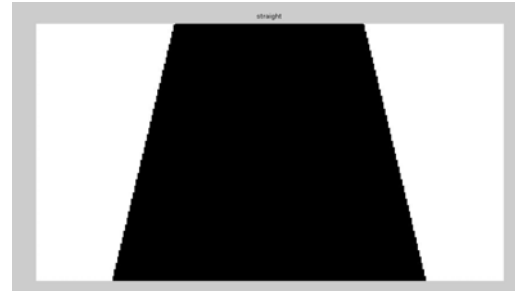
The primary navigation method in MARVIN is achieved by the use of a laptop and a USB WebCam. The navigation algorithm looks at the data given to Matlab by the various components to determine which direction to move. First, Matlab gets the black and white image developed by image processing that shows the cones and lines as white spaces.



The algorithm first 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.

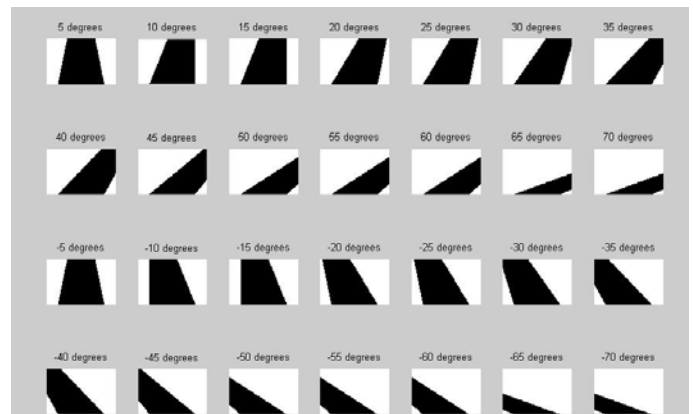
The figure on the right shows the area of the image that is checked to see if the path is clear; the black pixels represent the area check. The figure just shows the area of the image that will be checked if the image is looked at.



The black area on the image is the area that needs to be check 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 check, the algorithm looks at the number of white pixels in the area that is check. 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.

After the algorithm checks straight ahead and finds that the path is clear, it checks the same image, but the area that is looked at is an area that turns to the right at a slight angle (currently set at 5°). If the white space in that region is less than the threshold, the vehicle turns at that slight angle (5°).

Next, if the path is not safe, the algorithm checks to the left at the slight angle. If the path is still not safe, the algorithm continues to check to the right and left at incrementing angles until it sees a safe path. The black area on each image is the area that is checked to see if the path



is clear. The black area represents the path that will be taken if the vehicle is going to turn at the selected angle. The algorithm checks up to positive and negative 70° . If a safe path is still not seen, the vehicle will stop, and then go in reverse. This algorithm is the basis for checking the image to see if the path is clear.

For the Autonomous Challenge portion of the competition, the image processing is the main part of the algorithm. The first thing the algorithm does is 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. The process keeps repeating until the image processing portion does not give an angle. If this happens, the vehicle needs to back up and find a new angle to head in.

5.6 GPS Navigation

For the Navigation Challenge portion of the competition, the GPS is an instrumental part of the navigation. 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 give 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.

5.7 Vehicle Technical Characteristics

- Speed = 5.0 mph forward
- Maximum weight: 130 lbs.
- Ramp climbing ability
- Reaction times
- Battery: Uses two 12 Volt batteries and & two 9Volt Batteries with a 1hr life
- Distance at which obstacles are detected= minimum 30 cm maximum 3m
- How the vehicle deals with dead ends, traps, and potholes
- Accuracy of arrival at navigation waypoints, approximately 5m

6. COSTS FOR THE ROBOT

Qty	Item	Description	Price	Price to Team
1	Vehicle Base	Power Wheels Chevrolet Silverado Pickup Truck	\$300	\$200
4	Wheel Bands	Rubber bands for wheel traction	\$15	\$15
1 box	Golf Spikes	Rear wheel traction	\$6	\$6
A/R	Wood	1x1s and particle board for base support	\$15	\$15
2	Tie Rods	Teflon fabricated tie rods	\$5	\$0
5	Steering Eye Joints	Bearing style eye joints	\$2	\$0
1	Pittman DC Motor	Steering motor	\$25	\$0
2	Cogs	From servo to steering joints, with chain	\$2	\$0
1	Relay Circuit	Power systems relay circuit	\$3	\$0
1	RC Remote	RC remote control and receiving circuitry	\$14	\$14
2	Inverting Amp	Inverting amplifier circuits, per direction	\$1	\$0
2	Full-Wave Rectifiers	Full-wave rectifier circuits, per direction	\$1	\$0
	H-Bridges	Power systems H-bridges	\$5	\$5
1	Antenna	Antenna for long-distance wireless control	\$1	\$0
1	Stop Button	Industrial emergency stop button (red)	\$10	\$0
A/R	Plastic hoods	Weatherproofing plastic hoods	\$10	\$10
1	GPS	Garmin GPS eTrex unit with serial cable connection	\$90	\$90
1	Power System	Power systems + voltage regulators	\$20	\$20
4	Sensors	Polaroid 600 Sonar Ranging Sensor	\$80	\$80
1	Camera	350 Pixels Mini USB Web-Cam	\$30	\$30
1	Microcontroller	MiniDragon HC12 Development Board	\$158	\$0
1	Computer	Laptop with min: 2 USB, 1 Serial Port, 1 Parallel Port	\$1500	\$0
1	Matlab Program	Matlab 7.0.1 Program with Image and Data Acquisition Toolboxes	\$500	\$0
1	Dead Reckoning	Encoder with Hewlett Packard HEDS 5500	\$20	\$20
1	Digital Compass	CMPS03 Digital Compass	\$42	\$42
Total Cost			\$2855	
Total Team Cost			\$547	

7. CONCLUSION

The components of this vehicle have been designed for the specific task of participating in the IGVC. This project gave real life experience in many areas where similar designs and techniques could be used in numerous other applications. The potential of many real-world mechanical devices being able to control themselves is growing, and in some cases is very desirable. Each possible obstacle which the vehicle could encounter had to be realized and then acted on accordingly. Combining the various detection components into this navigation system provides MARVIN with the information needed to properly navigate through the unknown environment. The system will examine the data and come to an intelligent decision on how to navigate through a given course.

Faculty Advisor Statement

We, Dr. Rocio Alba-Flores, Dr. Fernando Rios Gutierrez, and Mr. Scott Norr of the Department of Electrical and Computer Engineering at University Of Minnesota, Duluth, together do hereby certify that the engineering design of the vehicle, MARVIN, has been significant and each team member has earned four semester hour credits for their work on this project.

Signed

Dated

Signed

Date

Signed

Date

