

# “Achilles Autonomous”



## Engineering Design Team

*Vincent Arguelles  
Dan Belarmino  
Tom Burdick  
Matthew Chan  
Sam Choi  
Fernando Garcia  
Eric Gesell*

*Leeder Hua  
Christopher Kaczor  
Jeff Kantarek  
Steve Kearney  
Anup Kotadia  
Nick Kutella  
Steven Mockus*

*Nick Novak  
Matthew Podrazka  
Rafal Radulski  
Emil Remolana  
Mihai Rosu  
Garrett Graham  
Sean Ryan*

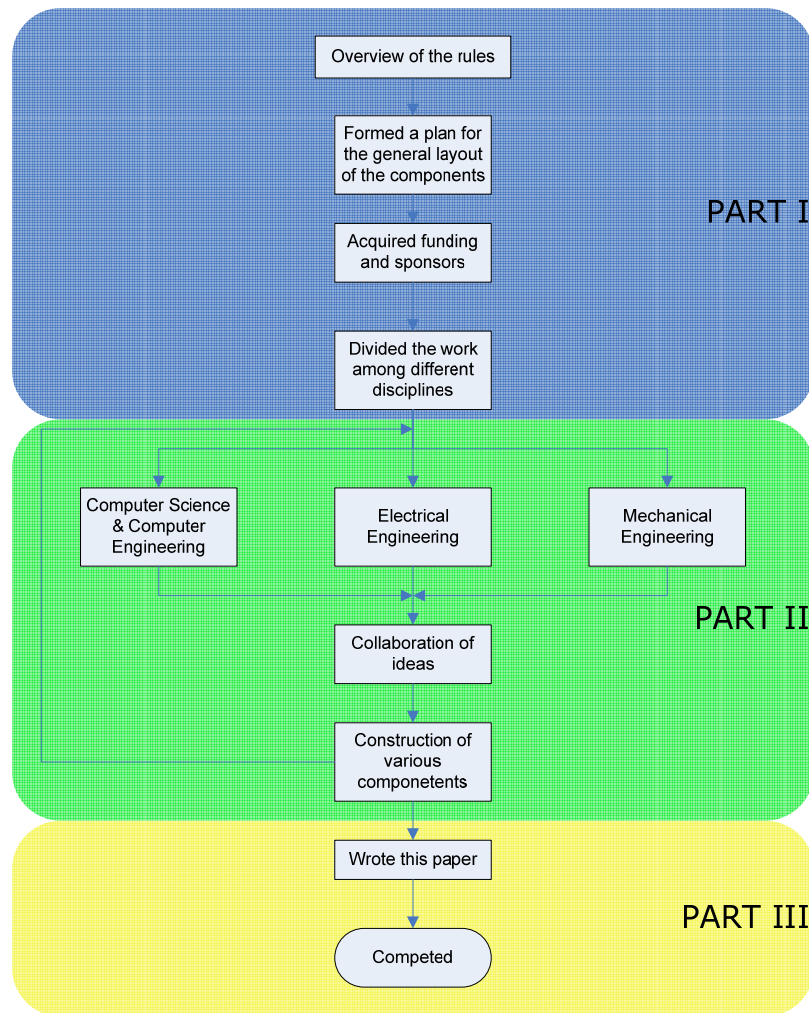
The engineering design in this vehicle by the current student team has been significant and equivalent to what might be awarded in a senior design course.

Faculty Advisor Certification Signature: \_\_\_\_\_

## 1.0 Introduction

Hello and welcome to the Achilles autonomous vehicle. Preparation for this vehicle has been underway for almost a year and incorporates the many disciplines of engineering. This vehicle was constructed solely for this IGVC. We have team members from all levels of engineering working together using their skills, talents, and knowledge.

## 2.0 Design Process

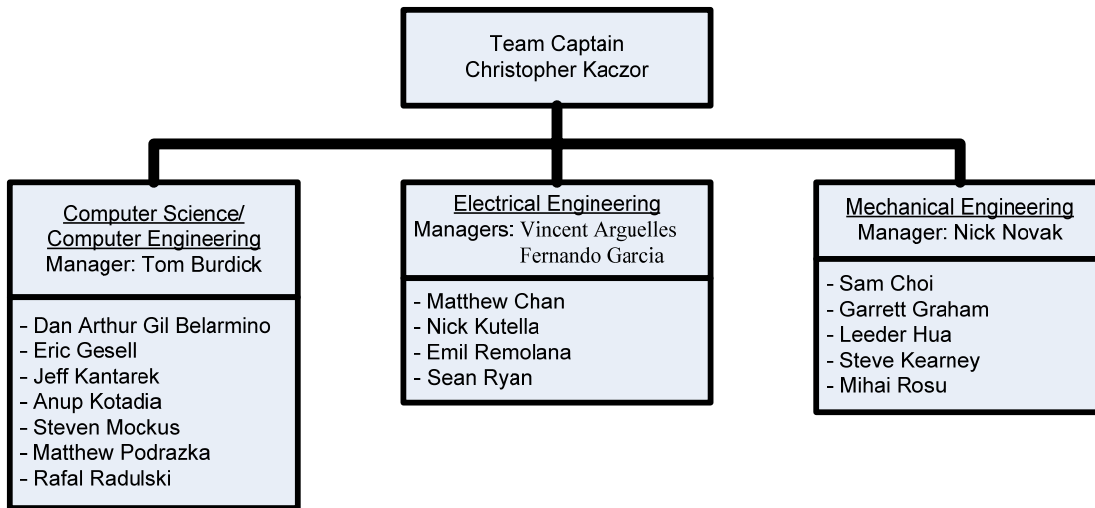


### 2.1 Design Process Part I

As a team we reviewed the rules of the competition and discussed what was needed to compete. We considered various components and construction techniques to achieve an efficient vehicle. A significant factor to determine the equipment needs was to evaluate our budget and acquire funding and sponsors. The work loads were then determined and separated among the engineering disciplines.



## 2.2 Design Process Part II



The different departments of engineering were distributed and the work was dispersed. Although each team member was assigned a particular manager it did not limit or confine them to working in just that area. A free exchange of ideas was always discussed among all members and the final design approaches were then maximized. With all the collaboration among members the best ideas were then finalized for the construction of the vehicle.

## 2.3 Design Process Part III

This paper was written as a collaborative effort among all the members of the team. We are excited and anticipating a successful outcome in our first competition at the IGVC.

## 3.0 Vehicle Design

The main goals for designing the vehicle were simplicity, accessibility to internal components, and durability while being able to achieve its objective of enclosing and protecting all of the systems necessary for autonomous navigation. Simplicity in design allows for undemanding manufacturing processes, reduction in costs, and minimization of potential failures. Accessibility is crucial for trouble free repair and maintenance. Lastly the vehicle must withstand all internal and external forces it will encounter during its operation.





### 3.1 Chassis

The vehicles frame is constructed out of alloy 4130 military grade 1" by 1" steel tubing. Due to our experience with steel tubing, it became the material of choice. The relative ease of creating any length of tubing as well as creating angled cuts makes manufacturing quick and simple. A welded frame can be cut and re-welded, allowing for quick solutions to unanticipated design faults. The vehicles frame is a demonstration of a design that can be very rugged, durable, and strong, while also keeping it lightweight.

### 3.2 Vehicle Interior

With all of the considerations for what was going inside, it was important to make sure that there was adequate space for accessibility. A majority of the components that comprise the vehicle are contained inside the frame and it becomes important to be able to conveniently access them. The frame is composed of two steel tubing boxes that are connected along an edge. A box frame combines good stability and strength with a simple large volume. Since most components themselves are rectangular or approximately rectangular in shape, it becomes natural to create a housing which is of a similar shape. This also permits rearrangements of the internal components if necessary.

### 3.3 Vehicle Exterior

With all of this open space, there comes the problem of exposure to the elements. An important objective is to allow the vehicle to be able to operate while it is raining. With that in mind, the vehicle needs to be water tight, but at same time be easily accessible. The frame is covered in sheets of polycarbonate and aluminum. Sheets of aluminum,  $\frac{1}{8}$ " thick, are used to cover the bottom of the vehicle. It provides enough material to sound protection while allowing the vehicle to maintain its lightweight performance. The top is covered in an opaque polycarbonate,  $\frac{1}{4}$ " thick, to prevent direct exposure to the sun's rays causing unnecessary



heating. Transparent polycarbonate, of the same thickness, is used to cover all of the other sides. Similarly to the aluminum, it serves as protection without adding too much weight. While the transparent look of the vehicle adds to its aesthetics, it also serves the purpose of allowing us to see inside to verify the condition of all parts and wiring. Most of the polycarbonate is attached using latches for quick removal. The other covers are attached directly to frame using weld nuts and machine screws, directly mounting them to frame. Every place there is polycarbonate directly mating with the frame, there are rubber gaskets in between.

### **3.4 Compartment Architecture**

The frame is partitioned into two compartments, the lower front, and the upper rear. The lower front compartment was designed specifically to house the batteries, motors, motor controllers, and the drive train. All of the heaviest components are nearest to the ground to lower the center of gravity and to keep most of the weight on the two front axels, relieving the caster wheel of excess weight. The upper rear compartment houses all of the laptops. In between the two compartments is an intermediate plenum which joins them. It is mainly used for routing cables to reduce clutter and for the placement of the hub that networks the laptops.

### **3.5 Enclosure Ventilation**

One of the major concerns is keeping the upper compartment cool. Heat generated from the laptops must be exhausted, but the vehicle also must be kept water tight. Cool air convection by fans is the method used to cool the laptops. Each laptop is arranged so that the heat outlets all face the same direction. In between each laptop there is empty space through which cool air flows by fans placed at the ends. The fan inlet and outlets are covered by screens, to prevent water from entering.

### **3.6 Aluminum Product**

Selecting the material for the other components within the vehicle such as the E-stop mount, axle mounts, motor mounts, shelf base, was an important decision in the design process for the vehicle. The goal was a material that was light, yet strong, and easy to work with. Aluminum met these requirements and is used for the necessary mounting brackets for various components. The aluminum parts we manufactured were from 1" by 1", 90 degree angle bars, 1/8" thick and from sheets 1/8" and 1/4" thick. This was an efficient way to create precise flanges and mounts that were low profile.



Aluminum was used to make the axle mounts; this is especially important because they need to be machined accurately to hold the axle bearings and the axle shaft. In order to make the axle mounts, we used a vertical milling machine, a machine taper, and a drill press. To cut the x and y dimensions of the mounts the use of a milling machine was employed to make precise and smooth cuts.

### **3.7 Shelving**

The shelves are constructed from sliding rails and aluminum 90 degree angles. The rails are fastened to weld nuts on the frame and the aluminum angles are fastened to rails, creating a draw to seat the laptop. With the added accessibility of the sliding shelves, any laptop can be pulled out for quick access to the vehicles computer system, without having to physically disconnect the laptop itself.

### **3.8 Drive Train**

The vehicle features a sprocket and chain drive chain. The chain driven system allows freedom in placement of the motors and wheels. On each side of the lower compartment is the drive train for that side. Each wheel has its own motor enabling the wheels to turn in opposite directions. This tank drive provides a much needed zero turning radius for ease of obstacle avoidance. Choosing the tank drive therefore not only makes the navigation of the robot less computationally intensive but also reduces the mechanical complexity of the system as opposed to using steering. Since stable and precise wheel rotation is necessary a large gear ratio of 12 to 1 was used which enables the vehicle to climb grades of up to 53 degrees.

### **3.9 Camera Tower**

The cameras are mounted approximately 5 ½ feet off the ground to provide a good overview of the terrain and to be able to see behind taller obstacles to better predict an ideal path. This height is adjustable by 1/32", the distance between the centers of the camera lenses is accurate to 1mm, and pitch of the cameras is accurate to 1 degree. Manual modification of these dimensions is possible through the mounting mechanism.

### **4.0 Electrical Operations**

The electrical operations of the vehicle are positioned in a way that would minimize the amount of wiring and allows the components to be housed safely.

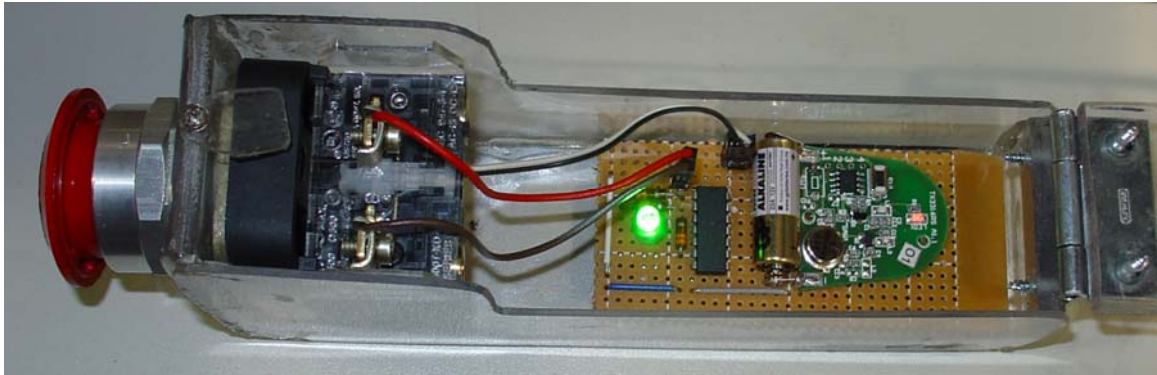


#### 4.1 Buffer boards

The Robostix will output two PWM signals to control the motors. To improve the quality of these signals, we will put them through a buffer.

#### 4.2 Electronic Safeguards

There are two mechanisms that are used to stop the vehicle. The first mechanism is a simple button on top of the robot that cuts power to the motor controllers. The second mechanism is a handheld wireless E-stop button (below).



The button that we used completes the circuit when the button is pulled up and it breaks the circuit when the button is pushed down. We needed to invert the operation of this switch to allow it to turn off when pushed down. We achieved this by using a 12 volt inverter (HCF4069UBEY) IC to invert the signal and reverse the operation of the button. These changes will allow proper operation of the transmitter and its range can exceed 120 feet.

#### 4.3 Power

Type of Battery	Amp Hours	Weight	Device Controlled	Duration (Hours)
PS-1212 12V Lead Acid Battery	1.3	1.32 lbs	Fans/electronics	1.3
PS12120 12V Lead Acid battery	12	8.50 lbs	Hub	4.6
PS-12260 12V Lead Acid Battery	26	18.0 lbs	Motors	.87

#### 4.4 Motion Control

The vehicle uses two 12 volt DC motors which are operated using PWM motor controllers. The motors will be connected to the motor controllers using Amp King 8 gauge wires. The motor controllers that are being used are the Victor 883s which have a rating of 24 volts at 60 amps of continuous current. They are very reliable and are considerably less expensive than most other controllers with the same ratings and interface. The motor controllers



will receive a 5 volt PWM signal from the microcontroller. The speed will be controlled by adjusting the duty cycle on the PWM which will in turn adjust the speed of the motors proportionally.

#### **4.5 Wheel Feedback**

In order to receive velocity feedback from the wheels there will be an encoder installed on the wheels. Robotics Connection Hamamatsu P5587 Wheel Encoder Module was used with the Robotics Connection 2.25 inch diameter 44 spoke encoder disk. This encoder set has a high level of precision that will allow the speed to be monitored. The encoder will send feedback in the form of a PWM signal which will produce a value of 5 volts when it detects the white section of the disk and 0 volts when it detects the black.

#### **4.6 Digital Compass**

In terms of making turns, that will be controlled with the assistance of a digital compass. The compass we will be using is a Devantech CMPS03 with 0.1 degree of resolution and 3 to 4 degrees of accuracy. The direction of the compass will help determine which wheel should spin faster or slower in order to be turned in the desired direction.

#### **4.7 Robostix**

The main objective of the Robostix is to communicate to the sensor devices using Two-wire Serial Interface and analog-to-digital converters. The Robostix gathers the data and sends it to the ARM9 board using UART. In addition, the Robostix is physically running the robot by generating PWM signals to the motor controllers, which directly run the motors. The PWM operation has its own external crystal clock to improve the efficiency of the process by avoiding the interruptions that would occur when doing operations such as I2C.

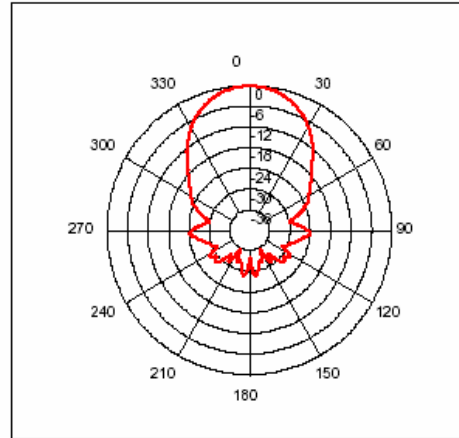
#### **4.8 ARM9**

The Robostix uses C programming. The program starts by initializing the PWM to neutral at 11% duty cycle to avoid any interference to the motor controller. Then the program sends a general call to the I2C devices and verifies that everything is responding and working properly. After making sure that all the sensors are working, the program runs a loop waiting for instructions from the laptop or the ARM9 board. When the robot starts running, the Robostix will gather the data from the tachometer in every loop. In addition, for safety purposes, the ARM9 board and the laptops send a heartbeat signal, which serves to make sure that the laptops and the ARM9 board are working properly and the program is still running.



## 4.9 Sensors

Connected to the Robostix board is an ultrasonic range finder sensor that can identify objects between 3 centimeters and up to 6 meters away. The range that is measured can be in inches, centimeters, or microseconds. The range finder uses about 12 milliamps during a range find. The figure on the right shows the beam pattern of the range finder.



The robot is equipped with an accelerometer to limit excessive vibrations of cameras that could cause blurred image. When the robot moves on an uneven terrain and the ARM9 board detects a motion exceeding the preset value, it will send a signal to the Robostix. Then, the Robostix decreases the motors' speed and informs the main system about the event.

Furthermore, a circuit that senses any loss of power preemptively warns the Robostix of impending failure and the microcontroller then shuts down the motors to prevent the vehicle from running haywire.

## 5.0 Software

Robotic software is often times complicated and computationally intensive. For both of these reasons, a divide and conquer design philosophy in which each individual problem is solved in a separate module was implemented for the robot. To speed development of a rather complicated series of components, existing programs and frameworks were adapted to suit the needs of the robot. The algorithms used to solve each part of the problem were researched thoroughly and planned to ensure seamless integration and accomplishment of all the required tasks.

## 5.1 Equipment

The software equipment consists of an array of sensors, four laptops, one embedded computer based on the ARM9 architecture, and an ATMega128 microcontroller. The four laptops and embedded computer are connected through a five port gigabit switch in a local network. The microcontroller directly connects to the embedded computer.



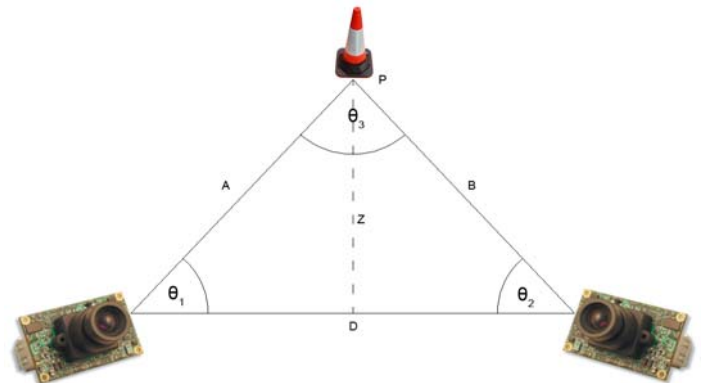
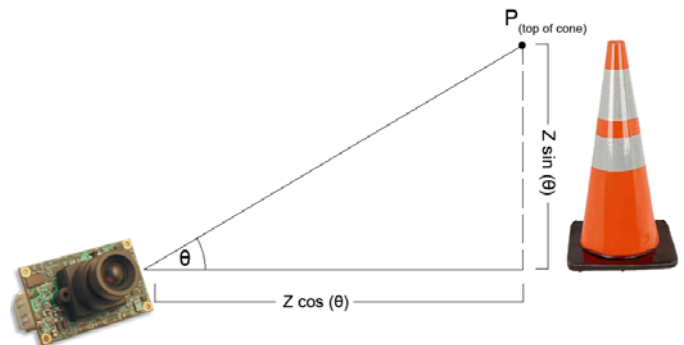
The laptops were used to distribute the computing power required by the components across several machines to take advantage of true parallelism. Some components take the entire processing power of a modern dual core processor in order to accomplish their task in real time and thus multiple computers are necessary to maintain a real time response from the software.

There are numerous sensors on the robot including two IEEE 1394 cameras, four sonar sensors, two wheel encoders, a compass, a gyroscope, a GPS receiver, and an accelerometer. The two cameras are used for both stereo vision and monocular vision to determine where obstacles, lines, and potholes are located. The gyroscope is used for stabilizing the robot while turning and determining the orientation of the robot needed in order to recreate a 3-D interpretation from the stereo vision. The algorithm to manage the vehicles velocity and reduce vibrations and camera blurring makes extensive use of the accelerometer. The compass determines the heading of the vehicle. Finally, the GPS receiver is employed to determine where the goal points are in the open field portion of the competition.

## 5.2 Vision and Object Detection

The robot uses stereo vision to determine the location of obstacles in space. It does so by determining the disparity between stereo images. The software then uses a look up table to determine the Cartesian coordinates of obstacles relative to the vehicle. The stereo vision system allows the vehicle to see past obstacles in a way that a laser range finder system cannot.

The three-dimensional object detection from our vision systems is broken up into five steps: forming a disparity map from stereo input, determining the positions in space of all the pixels in the disparity map, associating which pixels belong to which objects, line and pothole detection, and placing the detected objects onto a map representing the course.



### 5.2.1 Disparity Map

The disparity map is a gray scale image where each value represents that particular pixel's disparity; the distance in the x direction that the matching pixel in the other image is located. This is done by taking the left and right images, which have been rectified for any lens distortion, and sliding them horizontally across each other taking the difference at each step. This produces a three dimensional array known as the cost matrix indexed by  $C(x, y, d)$  where  $d$  is the disparity of some pixel at  $x, y$ . From the cost matrix there are several ways to obtain the matrix. The simplest solution is to find the first minimum cost for a particular pixel and use that as its disparity. While this method is fast it unfortunately produces speckled and noisy results that are difficult to interpret. A more elaborate approach uses an adaptive weight window based on distance and color. This is applied to each pixel in the cost matrix to compute a final cost. This method tends to be more accurate and produce less noisy results. Once this is done, the results can be improved even more by using a method known as dynamic programming.



The vehicle uses an adaptive weight window and “winner takes all” selection since this method is computationally efficient, reduces noise a great deal, and allows the disparity map to be completely run on a graphics card. The use of the graphics card ensures this algorithm runs in real time since a graphics card can perform the needed calculations more efficiently than a multipurpose CPU.

### 5.2.2 Position Interpolation

To determine where each pixel in the disparity map is in three-dimensional space, the technique of parallax is used. The angles at which a point appears in relation to each of the cameras are calculated from the disparity map, as are the angle of elevation of the point. Based on this information and using the distance between the cameras as a baseline, the location of each point the cameras see can be determined in a three-dimensional space relative to the vehicle using the law of sines. Every possible input for each pixel location is calculated ahead of time and stored in a lookup table to ensure speed.

### 5.2.3 Object Association

The next step is to associate each pixel on the disparity map to a particular object. The method used is to divide the disparity map into sections and determine the similarity between



adjacent sections. This method divides the disparity map into small square sections of pixels and calculates the statistical mode of the pixel values in that subsection and then assigns all the pixels in that section the value of the mode and marks those pixels as belonging to a single object. However if a section does not have a certain amount of pixels being equal to the mode, the section will be ignored in order to filter out noise. The algorithm compares the mode of the current section to adjacent section modes and if there are matching adjacent modes, the matching sections are grouped into a single object and this area is further compared to adjacent sections to create bigger objects. The process then continues until either no more adjacent sections can be added to the current object or a certain threshold is reached, which is to limit objects reaching an impractically large size. The object associations are then sent to the object-mapping module of the software.

#### **5.2.4 Line and Pothole Detection**

Line detection is one of the most important functions in the vision system, since leaving the course will disqualify the vehicle. Line detection is done using the open source computer vision library, providing functionality for real time applications. The line detection application receives colored images from the cameras and filters out the red and green components of each pixel. At the same time, the upper  $\frac{1}{4}$  of the image is eliminated to remove the sky. The result is a black and white image with much of the grass suppressed as darker shades while leaving the lines brighter shades. A gray level threshold filter is applied to transform all pixels under a certain gray level as black and the remaining pixels are made white, creating a binary image. At this point, everything that is white in the image represents lines and potholes. The next process is to apply to library Hough Transform functions. The Hough Transform takes the binary image and best fits lines and circles into the image.

This binary image is then entered into another data array similar to the one for object association. The lines surrounding the potholes are also included in the binary image and thus the potholes will appear as ringed obstacles in the subsequent vision and navigation processing. In contrast to object association, no comparisons are made to adjacent sections in the binary image since the binary image with the line data contains no information which can be used to interpolate position in three-dimensional space and so all eligible points must be considered as lines except those filtered out for noise reduction. All sections with modes that are designated as lines are marked as objects and submitted to the object mapping section of the software in a



format identical to that for object association.

### **5.2.5 Object Mapping**

The lines and objects identified in the previous steps are converted to rectangular approximations to be placed on an overhead view map of the course. This is done by taking the four corners of each detected object in the quad tree and using their locations and values to find their respective locations in three dimensional space using the look up table established earlier. These four locations are converted from being relative to the robot to being relative to a global reference frame using a transformation matrix. The four converted locations contain information which is used to determine the smallest rectangle which can surround the specified object and the rectangle is placed in a data structure representing an overhead map of a limited portion of the course. The other lines and objects in the scene are all placed on the map data structure in the same fashion and then the map structure is sent to the localization module.

### **5.3 Simultaneous Localization and Mapping**

One of the more difficult tasks in robotics is determining where the vehicle is and mapping in relation to some fixed coordinate frame. The difficulty stems from accumulating error in the wheel encoder interpretation. SLAM is a set of algorithms that solve this problem by applying probabilistic math. DP-SLAM uses a true grid obstacle map which does not require the interpretation of features or landmarks from sensor readings.

DP-SLAM works much like other popular SLAM algorithms by using a particle filter. A particle filter works in SLAM by distributing particles across an area on the fixed frame map where the vehicle is likely posed. Each particle, representing a particular vehicle pose, is then weighted by how closely it matched what was the likely movement of the vehicle and what the vehicle should see based on what the map looked like previously. The low probability particles are culled and new particles are introduced in a process known as re-sampling.

DP-SLAM is certainly not unique in its use of a particle filter, its unique in the way it stores the particles and their observations of the world. The map stored by DP-SLAM consists of a grid or array in memory where at each location or cell a balanced binary tree of particle id numbers is stored. Each particle itself stores all of the places it has ever made in observation in its lifetime. By doing so the most common pitfall of other SLAM algorithms is avoided, an assumption that only one good map solution exists with a Gaussian probability distribution function.



Once DP-SLAM has produced a map with a number of weighted particles representing possible vehicle poses there are several possible routes to go to determine where the vehicle is and what the current map looks like. The component that does this task uses a method that takes the highest weighted particle, the most likely particle to represent where the vehicle is, and then creates a weighted average based on surrounding high weighted particles.

#### **5.4 Path finding**

Path finding is done by a component written by the authors of Orca. One of the cell decomposition path finders was chosen as it works very well and is already implemented. Cell decomposition works by taking the map, location of the vehicle, location of some goal and then dividing the map up into free geometric areas, also called a cell. Each cell then gets represented as a node in a graph structure where it is connected to adjoining cells with an available path. This is done using a simple graph exploring algorithm. Determining the goal point is done for us by GPS in one part of the competition.

In the track portion of the contest it is more difficult to find a goal and a potential field function is used instead. The potential field works by pushing the vehicle away from obstacles and towards unexplored regions in the direction the lines are traveling.

#### **5.5 Communication**

Communication between the different components is accomplished through a middleware layer called Ice. A set of libraries written with Ice called Orca are used to join the different software modules. Orca is a robotics framework that allows asynchronous communication between components through the Ice middleware. Each component implements an interface defined by a data definition file specific to the Ice framework. By using Ice and Orca, the amount of effort needed to implement networking has been significantly reduced with regard to communication between the various components.

#### **5.6 Benchmark**

	Minimum (m)	Maximum (m)		Software & Controls (ms)
Vision Range	1	10	Reaction Time	150



## 6.0 Vehicle Cost

### 2006-2007 Intelligent Ground Vehicle Competition (IGVC) Budget Report Summary

Qty.	Component	Retail Price(ea.)	Sub Total	Discounted (ea.)	Discounted Total	Supplier
------	-----------	-------------------	-----------	------------------	------------------	----------

#### Power Electronics

2	12V 26AH Batteries	\$ 38.80	\$ 77.60	\$ 10.00	\$ 20.00	Battery.Web.com
2	12V 12AH Batteries	\$ 21.76	\$ 43.52	\$ 5.00	\$ 5.00	Battery.Web.com
1	12V 1.3 AH Batteries	\$ 11.12	\$ 22.24	\$ 1.00	\$ 1.00	Battery.Web.com
2	24V Victor 833 M. Controller	\$ 149.00	\$ 298.00	\$ 149.00	\$ 149.00	IFI Robotics

#### Computer Components and Electronics

2	Cameras	\$ 200.00	\$ 400.00	\$ 200.00	\$ 400.00	Point Grey
2	Lens	\$ 100.00	\$ 200.00	\$ 100.00	\$ 200.00	Jknelectronics
1	Arm9 board	\$ 53.00	\$ 53.00	\$ 53.00	\$ 53.00	Technologic Systems
1	Robostix	\$ 49.00	\$ 49.00	\$ 49.00	\$ 49.00	Gumstix.com
1	Gumstix connex 200xm	\$ 109.00	\$ 109.00	\$ 109.00	\$ 109.00	Gumstix.com
1	USB 12-Channel GPS receiver	\$ 34.99	\$ 34.99	\$ 34.99	\$ 34.99	Amazon.com
2	ASUS Laptop	\$ 1200.00	\$ 2400.00	\$ 1200.00	\$ 2400.00	Newegg.com
1	ASUS Laptop	\$ 1300.00	\$ 1300.00	\$ 0.00	\$ 0.00	Newegg.com
1	HP Laptop	\$ 150.00	\$ 150.00	\$ 150.00	\$ 150.00	CDW
1	6 PORT 1394 HUB	\$ 40.00	\$ 40.00	\$ 40.00	\$ 40.00	Frys.com
1	5 PORT Network Switch	\$ 27.99	\$ 27.99	\$ 27.99	\$ 27.99	Newegg.com
1	2 Channel RF Relay	\$ 39.49	\$ 39.49	\$ 39.49	\$ 39.49	HobbyTron.com
4	SRFO8 Ultrasonics Sensor	\$ 54.11	\$ 216.44	\$ 54.11	\$ 216.44	RobotShop
1	Magnetic Compass	\$ 57.99	\$ 57.99	\$ 57.99	\$ 57.99	RobotShop
1	Wheel Encoder Module (pair)	\$ 36.99	\$ 36.99	\$ 36.99	\$ 36.99	RobotShop
1	Spoke Encoder Disk (pair)	\$ 17.99	\$ 17.99	\$ 17.99	\$ 17.99	RobotShop

#### Chases and Hardware

5	Low-Carbon Square Steel	\$ 34.40	\$ 172.00	\$ 34.40	\$ 172.00	McMaster-Carr
4	Aluminum L-Channel	\$ 19.19	\$ 76.76	\$ 19.19	\$ 76.76	McMaster-Carr
2	Aluminum Stock (1/8 thick)	\$ 74.31	\$ 148.62	\$ 74.31	\$ 148.62	McMaster-Carr
16	Latches	\$ 5.36	\$ 85.76	\$ 5.36	\$ 85.76	McMaster-Carr
3	Polycarbonate Sheets	\$ 61.18	\$ 183.54	\$ 61.18	\$ 183.54	McMaster-Carr
1	Button Screws (100 pack)	\$ 11.06	\$ 11.06	\$ 11.06	\$ 11.06	McMaster-Carr
1	Hardened Precision Shaft	\$ 18.44	\$ 18.44	\$ 18.44	\$ 18.44	McMaster-Carr
1	Sprockets and Chain	\$ 37.87	\$ 37.87	\$ 37.87	\$ 37.87	McMaster-Carr
4	Air Linear Bearings	\$ 7.40	\$ 29.60	\$ 0.00	\$ 0.00	McMaster-Carr
1	8" Pneumatic Caster Wheel	\$ 61.96	\$ 61.96	\$ 0.00	\$ 0.00	Gilmore-Kramer
2	12.5" Wheels	\$ 34.99	\$ 69.98	\$ 0.00	\$ 0.00	Robot Mkt. Pl.
2	12V Motors	\$ 317.00	\$ 317.00	\$ 0.00	\$ 0.00	Palmer Ind.

<b>Total Cost</b>	<b>Retail Cost \$ 6,786.83</b>	<b>Discounted Cost \$ 4,890.93</b>
-------------------	--------------------------------	------------------------------------



## 7.0 Conclusion

The estimated time that our team members contributed to this project is approximately 8000 hours. The talents and skills of each team member were utilized to complete the project with our best efforts. Finding the time between classes and jobs was a challenge but we managed to have a successful collaboration of ideas and knowledge. The entire process of building this vehicle was a very beneficial learning experience for all of the team members. Each of the team members were exposed to the other disciplines of engineering and an appreciation for each others dedication was emphasized. The vehicles stereo vision sets us apart from other teams and our design provides a very rugged and sturdy platform. The use of four laptop computers to communicate to the various sensors provides a very efficient configuration. We are ready for the thrill of the competition at the 15<sup>th</sup> annual IGVC and hope to compete again next year.

Special thanks to UIC faculty, Sponsors, and the UIC College of Engineering for their support in this project.

