Proudly Presents:

Sparta

Intelligent Ground Vehicle Competition 2011

Team Members

Phil Barnett, Dan Bosse, Nick Cappello,

Andrew Donihe, Ben Edwards, Takeshi Ei, David Griffin,

Steve Hinderlider, Ed Miller, Zach Zeiders

Introduction

The senior engineering class of York College is proud to present Sparta for the 2011 Intelligent Ground Vehicle Competition. Sparta is a rugged vehicle intended to navigate toward a goal without external input. The vehicle was developed as a capstone design project by electrical, mechanical, and computer engineering students beginning in May of 2010. These students began their work before York College's debut IGVC entry "Green Lightning" had participated in the IGVC, so in many ways, this year's entry did not have the benefit of experience. While the "Green Lightning" vehicle did not qualify in the 2010 IGVC event, the team received many compliments on the custom-made frame and tracked suspension system. Due to this positive feedback, Sparta employs several of the concepts from the previous design, while also implementing several fundamental and necessary changes.

Team Organization

Sparta was designed from the ground up by a team of undergraduate students beginning their 6th semester of coursework. This team of ten students was divided into three sub-teams: sensors, decision making and communication, and structure and motion. The sensors team was responsible for selecting sensors to collect information from the robot's surroundings, testing the effectiveness of the selected sensors, and writing the code necessary for their successful implementation into the overall programming. The decision making and communication team was responsible for mapping the sensor data and planning a path around the obstacles. The structure and motion team was responsible for selecting, designing, and fabricating the body, frame, and ground contact of the robot as well as the drive and power systems. Figure 1 below shows the engineering disciplines involved with each sub-team.

	Project Team Leader	
	Dan Bosse (ME)	
Sensors Team	Decision Making and Communication Team	Structure and Motion Team
Phil Barnett (EE) - Sub-team leader	David Griffin (CE) - Sub-team leader	Nick Cappello (ME) - Sub-team leader
Steve Hinderlider (CE)	Andrew Donihe (EE)	Takeshi Ei (ME)
Zach Zeiders (EE)		Ben Edwards (ME)
		Ed Miller (EE)
		Dan Bosse (ME)

Figure 1: Team Organization

Design Process

The initial step of the design process was to establish a set of detailed specifications for the project. Some of these specifications were based directly off of competition specifications, but most were created by the team to detail how the vehicle should perform. The team then searched through a wide set of options that might satisfy these constraints, trying to find solutions that were both cost effective and able to be implemented within the time allotted. Once a solution was found that the team was confident in, simulations and tests were performed to prove the concept. After basic subsystem testing the team ventured further into the design process by targeting a functional integrated electro-mechanical system. This prototype system became the test platform for further improvement, testing, and revision of the higher level system functionality needed in the competition.

Design Innovations

As the design process for Sparta was begun before York's first vehicle had competed, much of this year's efforts were spent on basic functionality and rising to the level of other entrants. As such, much of Sparta's design might not be considered innovative by seasoned judges. However, our vehicle still incorporates many features that we believe to be distinctive among IGVC entrants which warrant mention.

1. Custom passively damped tracked ground contact system
2. Multi-rate sensing and control
3. GPU Acceleration

The passively damped tracked ground contact system uses a series of plastic track sections that can be easily disassembled for modification or maintenance. Sparta also has a working suspension system. The suspension design utilizes four sets of road wheels on each side of the vehicle. The sets of wheels are paired up and connected to a rocker assembly that is connected to the frame by a long arm. By having only one arm for every two sets of road wheels, rather than for every set of road wheels, the suspension system was able to be constructed using fewer parts. This results in a lower total weight and increased suspension movement. The entire suspension system has also been designed with adaptation in mind. Most of the suspension components are held together using pins, allowing parts to be replaced quickly. Other, more permanent, designs would have restricted the team to continue using a part that had been deemed unsuccessful or required a large amount of deconstruction and assembly time to fix any problems encountered.

A functional suspension system benefits all of the sensors of the vehicle as well because it dampens the effects experienced from uneven driving surfaces, resulting in more reliable sensor data, especially from the vision system. Without a suspension system, the images captured by the vehicle camera would have a higher probability of being blurred. Blurred images would be especially problematic when trying to determine the location of white boundary lines. The suspension system also results in less movement in the body of the vehicle. Since the LIDAR mounted directly to the body of the vehicle, the data recorded by the LIDAR is far more reliable than if it were mounted on a vehicle where the wheels were attached directly to the frame.

In order to achieve fine grained movement control, it was necessary to implement multi-rate sensor sampling and control loops. Essentially, if the robot were to be controlled from the same loop that vision processing and path finding are, the reaction speed would only be as quick as those algorithms could update. Therefore, the robot implements two control loops; a primary loop and a motor controller loop. The primary loop collects data from the slowest components (the LIDAR, GPS, and Camera), formats that data into a map, and plots a path through it. The movement control loop uses that path to estimate where the robot should be when the primary loop next finishes (based on its current speed, surroundings, and loop run time). It then begins a cycle of repeatedly polling the IMU and Encoders for position information, recalculating the distance to its local goal, and actuating the motors appropriately to reach that goal. Doing all of this allows the motor controller to act independently of the primary loop's run time and allows for a much higher degree of control over the robot's movement.

In order to lower the total system response time, the navigation and mapping algorithm offloads some of the more computation heavy tasks, such as the vehicle width mapping algorithm, to the laptop's GPU. The vehicle width mapping algorithm is responsible for making the Map account for the width of the robot and it works by having each node on the map inherit the highest neighboring value within a certain radius. However, in order to make the algorithm easily parallelizable over a GPU, it needed to avoid data dependencies and resource protection. We were able to come up with a thread safe, data independent system by having every single node search the area around them for higher values rather than taking nodes with higher values and propagating the value outward to other nodes (causing a multiple writer problem). By parallelizing width mapping over the GPU, we were able to achieve a speedup of about 15.

Design Specifications and Vehicle Performance

Sparta was designed to be just larger than the smallest physical dimensions allowed by the competition. While this increased the difficulty of packaging all of the components, it led to a reduced footprint for the robot, making it easier to navigate near close obstacles, and reduced robot weight, allowing for a quicker vehicle response with lower power consumption.

Table 1: Physical Characteristics of Sparta

Weight	175 lbs (without 20 lb. payload)
Width	26 inches
Length	39.5 inches
Height	47 inches (including camera tower)
	17 inches (without camera tower)

Using many of the team-developed specifications, a vehicle was designed that could last long enough for the competition, travel at a desirable speed, climb ramps and other sloped surfaces, and detect and respond to obstacles and goals in a timely manner.

Table 2: Performance Statistics of Sparta

Maximum Speed	8 mph
Ramp Climbing Ability	Traction limited to 32 degrees
Response Time	< 250 ms
Battery Life	Minimum of 30 minutes
Obstacle Detection Range	> 33 feet (60 feet maximum)
Accuracy of Arrival at GPS Waypoints	Within 0.6 meters
Vehicle Power	900W

These figures were reached through a combination of component specifications, simulation testing and physical testing.

Lane Following

The system uses a Creative Live! Webcam to detect the white lines on the grass that determine the edge of the course for the Autonomous Challenge. The webcam is located at the top of the camera tower and is pointed down toward the ground at an angle such that the camera's field of view is adjacent to, but does not include, the edge of the payload. The camera has a wide angle lens that enables it to sense a larger amount of space. In order to detect the lines, the image is processed through a series of image processing steps shown in Figures 2 and 3 below.

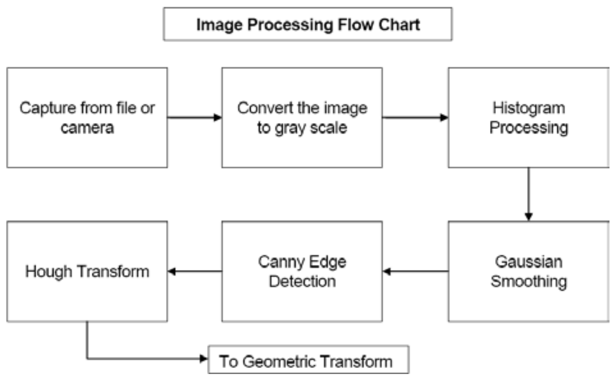

Figure 2: Image Processing Flow Chart

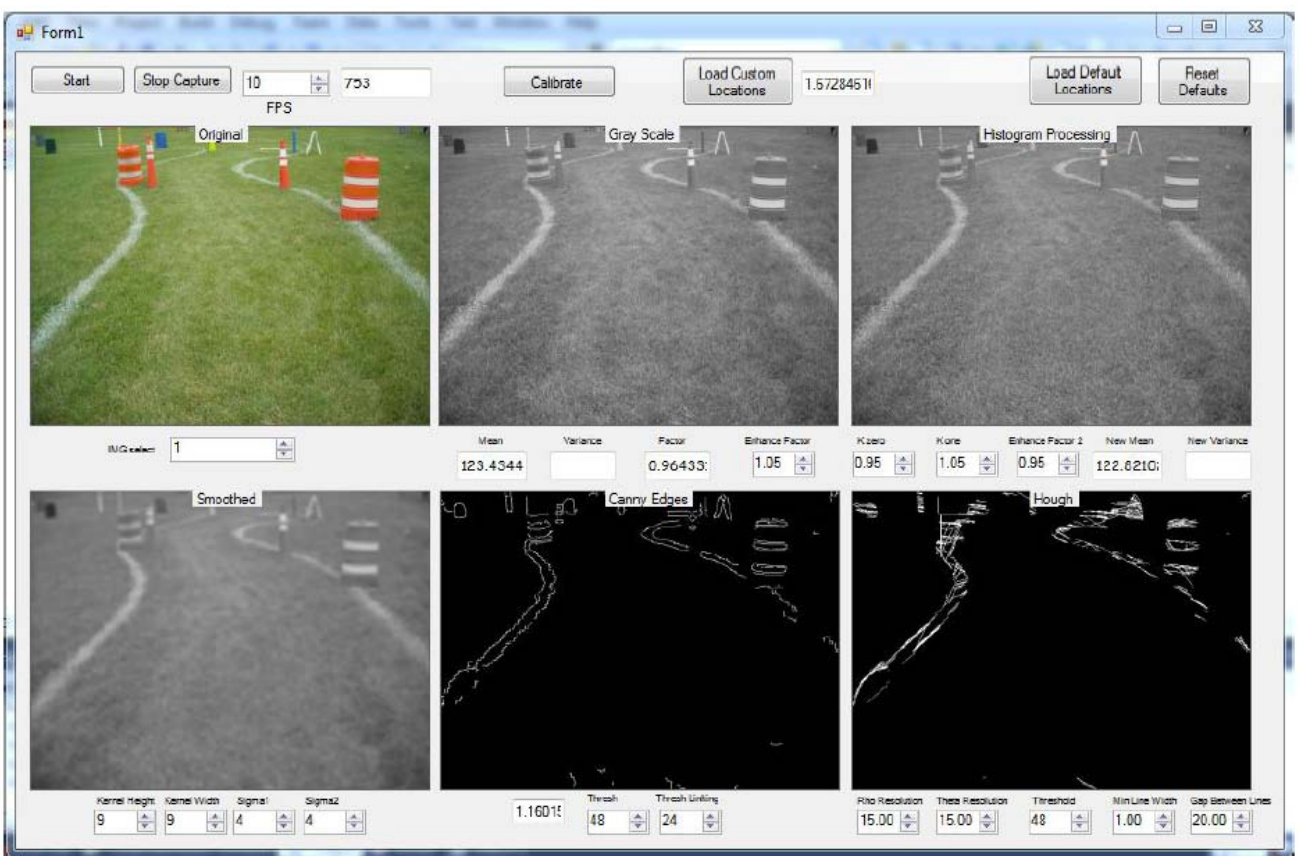

Figure 3: Visual Depiction of Image Processing Flowchart (Top to bottom, left to right)

Once the image has been processed, and the lines have been identified, the image enters a geometric transform operation to determine where the lines are located with respect to the vehicle. The geometric transform takes the location of each pixel on the camera's 2D representation of the robot's 3D surroundings and maps it to a location on a 2D Cartesian map as viewed from above the robot facing down. The locations of the lines are then represented as probable obstacles on the probabilistic map from which Sparta plans its path.

Obstacle Avoidance

Avoiding obstacles within the white boundary lines that govern the edge of the course is also of high importance since distance penalties and threats of run termination are consequences of striking an obstacle. To ensure obstacle detection, the vehicle uses a SICK LMS111-10100 LIDAR sensor. By centrally mounting the sensor, and ensuring there were no vision obstructions, the full 270 degree viewing angle of the LIDAR is capable of detecting any obstacles to the front and sides of the vehicle. By using only one sensor as the main means of obstacle detection, the amount of data that needs to be interpreted before an obstacle can be plotted on the map has been reduced. Testing has shown that the LIDAR capability to take samples every 0.25 degrees rather than every 0.5 degrees increases the reliability of the sensor while incurring little risk. Taking twice as many samples with the LIDAR was potentially costly, because it could increase the processing time and therefore system control period. During testing, it was determined that recording twice as many data points only increased the runtime by a maximum of only 1.7 milliseconds.

Table 3: LIDAR Runtime

		Runtime (ms)		
		0.25°	0.50°	Slowdown
# of Attempts	100	4.850	3.760	28.99%
	1000	5.550	3.915	41.76%
	2000	5.671	3.929	44.34%
	10000	5.082	3.940	28.98%

Probabilistic Mapping

In order for our path finding algorithm to correctly determine where the vehicle needs to go, the program first needs to determine where all of the obstacles are using probabilistic mapping. In order to create a probabilistic map, the robot takes all of the sensory data from the encoders, LIDAR, IMU, and webcam and through a series of filters and calculations determines where the obstacles are in relation to the vehicle. Figure 4 shows the process of taking sensory data, passing it through filters, and making a probabilistic map before sending using the path finding algorithm to determine a direction of travel.

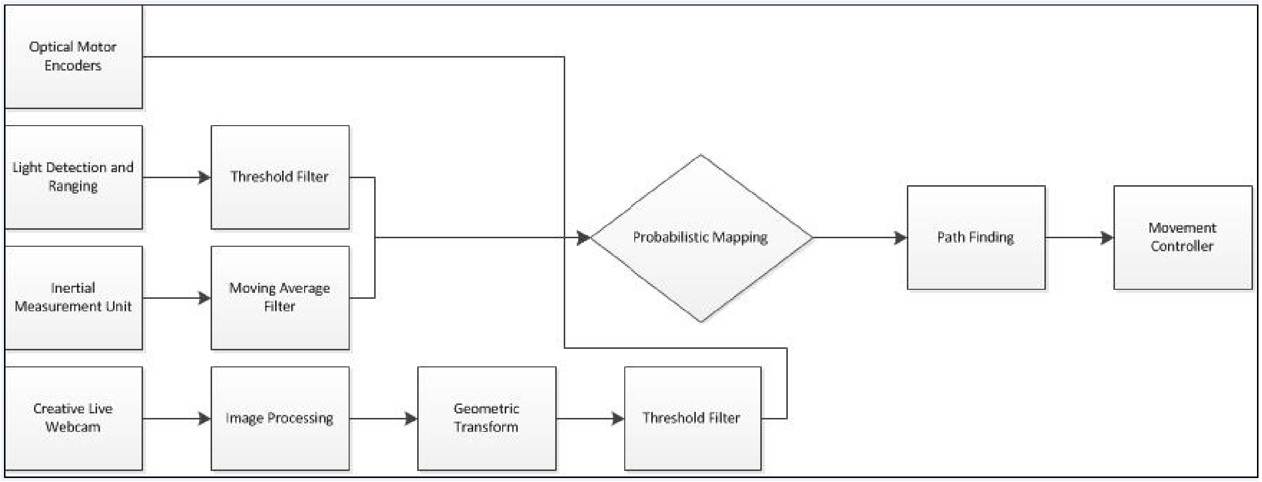

Figure 4: System Algorithm Flow Diagram

A Cartesian square map is formed using this sensory data and probabilistic fading to allow previously viewed objects to remain on the map for a set amount of time. The mapping program that has been implemented places the robot at the center of the map to allow the robot to have a memory of some of the objects it has already passed. In Figure 5, the map is being filled mainly using data points taken by the LIDAR to find the obstacles around the robot. Since the LIDAR has a 270 degree viewing angle, the 90 degrees of the map from the center to the bottom two corners include obstacles with lower probabilities. These are obstacle positions that the robot has previously detected, but currently has no data points for. The map utilizes an initial probability based on the accuracy of the sensor for obstacles which gradually decreases until the data point fades completely off of the map. This memory will help the robot when dealing with dead ends and complicated obstacles because it will allow the robot to remember, with slight uncertainty, where obstacles are located behind it before it tries to back up and correct its path.

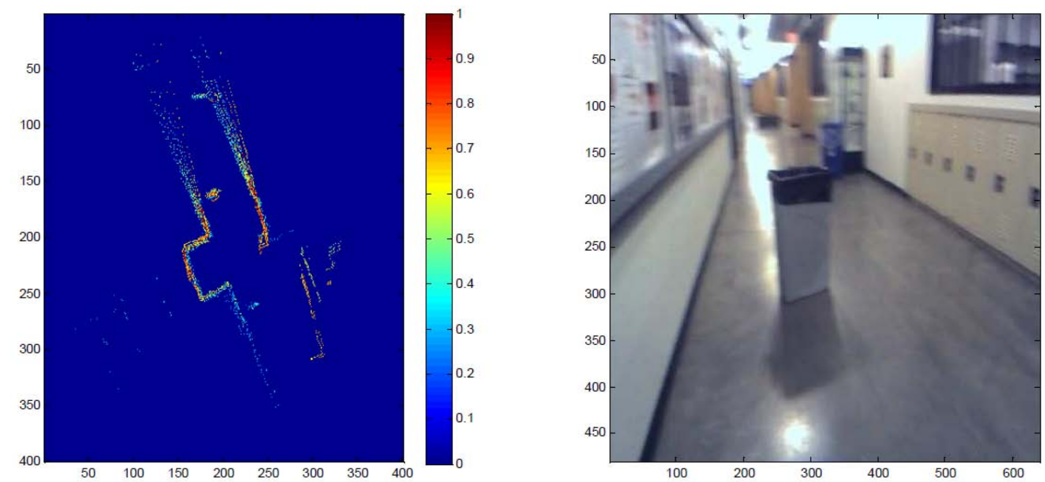

Figure 5: Probabilistic Map vs. Visualization of Surroundings

The map also takes in data points from the vision system of the robot to help plot the white lines and other 3D obstacles onto the map for avoidance. In Figure 6, the robot is able to see the white squares running down the hallway and adds them to the probabilistic map. The seemingly curve distortion observed in the map is due to the Geometric transform procedure that is performed on the vision data to determine the distance to objects such as white lines, which will be corrected before the competition.

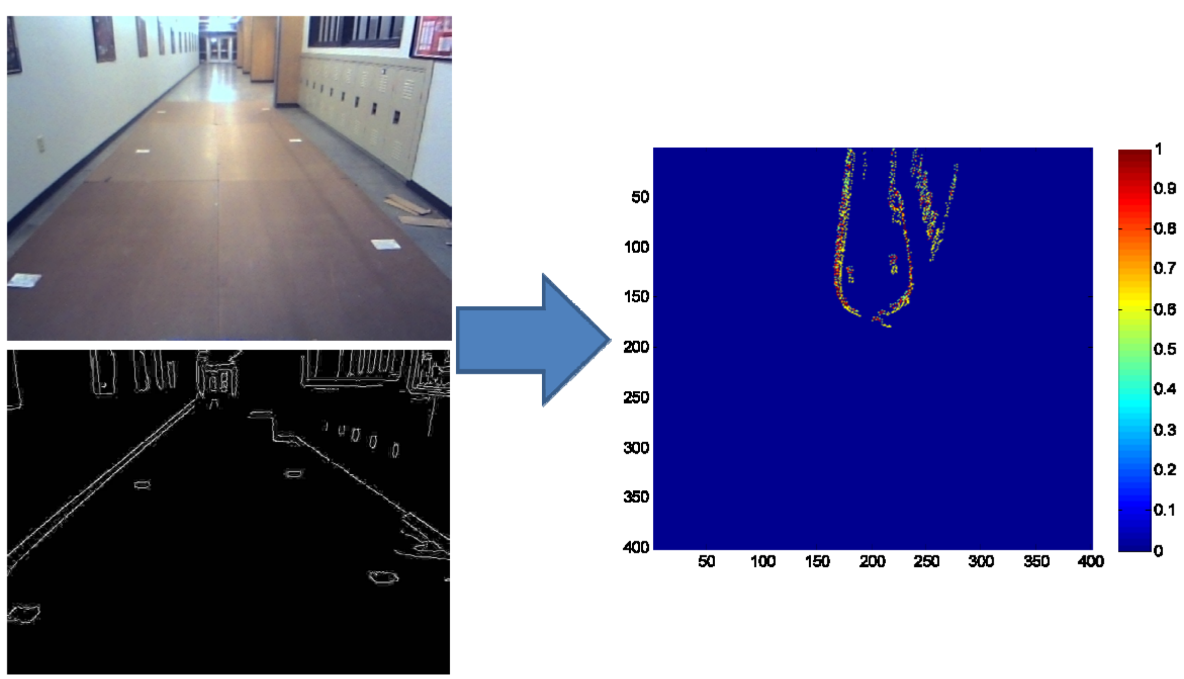

Figure 6: Camera View and Edge Detection to Probabilistic Map

Navigation

Once a probabilistic map is created, the robot needs to be able to determine how it will move through the set of obstacles. Several mapping algorithms were explored to determine which one would best suit our needs. Dijkstra's algorithm was researched as a possible solution but the aspect that it explores all possible paths available on a map before making a decision proved to be too time consuming and inefficient. A* was then considered due to its ability to make intelligent exploration decisions because of estimates and costs based on partially explored paths and heuristics. Since it requires less time and explores less possible paths, A* was selected as the path finding algorithm for our project.

One of the problems presented by path planning algorithms is that they essentially plan a path that is one node wide and, in an attempt to find the shortest path possible, have a tendency of planning the paths using the nodes directly next to detected obstacles. The planned path illustrates where the center of the vehicle should be travelling. Since the vehicle we have designed has a width that is greater than the 2" node width that we have decided to use, using the algorithm directly would cause our vehicle to collide with obstacles. In order to eliminate collisions between the vehicle and the obstacles along the path, we have to coax the algorithm into believing that the obstacles are wider than they actually are in a process that the Sparta team members refer to as vehicle width padding.

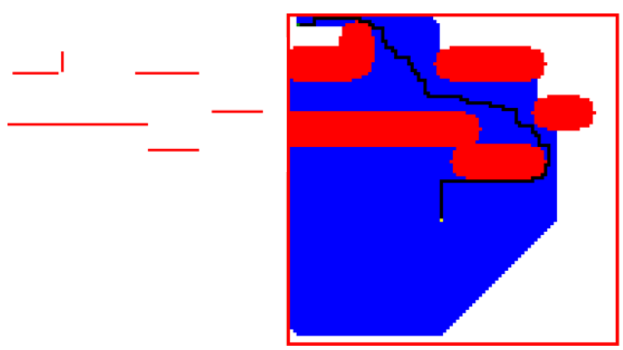

Figure 7: Detected Obstacles and Path Planned after Vehicle Width Padding

Left image: thin wall obstacles, right image: padded obstacles (red), explored paths (blue) and decided path (black)

Vehicle width padding takes the high probability locations of obstacles and expands them by a predetermined radius with respect to the map. The radius is found by computing the vehicle's width in relation to the number of cells in the map and the area that they represent. Since our vehicle is capable of zero-turns, the widest that our vehicle could possibly be is the diagonal length of our robot. Since the path is planned from the center of the robot, the padding radius, R, of the obstacles only needs to be the diagonal length divided by two. So, the way we accounted for the vehicle width was by having each node in the map inherit the highest neighboring node's value within a certain radius. By having each node on the map inherit the highest neighboring value we eliminated the multiple writer problem associated with having each node propagate its value to its neighbors, making the algorithm trivially parallelizable. Furthermore, we determined that by running the algorithm twice using a circle of radius of R/2 that we could complete the padding on the probabilistic map in a shorter amount of time than by doing one pass with a radius of R. An example of this can be seen in Figure 8 and Table 4.

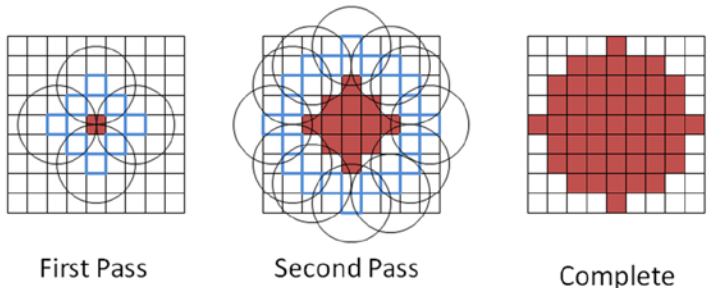

Figure 8: Two Passes of Width Padding with Half Diameter Circles

Table 4: Averaged Run Time Over 50 Iterations Using Different Sized Circles on a 400x400 map

	15" diameter		30" diameter	
	Parallel	Serial	Parallel	Serial
2 circles	134ms	296ms	459ms	974ms
1 circle	228ms	486ms	835ms	1723ms

Essentially, by convolving two circles of radius R/2 together the resultant shape on the map becomes a circle of radius R. However, the area of two circles of radius R/2 is less than the area of a single circle of radius R, thus the strategy uses less computation power, as shown in Table 4 above. Additionally, this algorithm is also currently being adapted to an Nvidia GPU using CUDA. Initial tests show that the algorithm will complete on the GPU in fewer than 10 milliseconds, a speedup of over 10. This improvement allows Sparta to plan much more detailed maps in a much shorter period of time, giving us much finer grained and more responsive control than would be possible otherwise.

Software Strategy

Our robot uses map which consists of a square grid of nodes that each represent a location relative to the robot. The corresponding area that each node covers is determined by the amount of nodes that make up the map and the total area the map represents. So, a 400x400 node grid representing a 66.6'x66.6' area equates to approximately 2x2 inches per node. This map is then populated in the Primary Control Loop using a probabilistic mapping algorithm being fed data from the sensors and then solved using the A* pathfinding algorithm. The results of the path finder are then sent to the Movement Control Loop, which continually makes decisions based on the current path information. A more detailed explanation can be found following Figure 9 below.

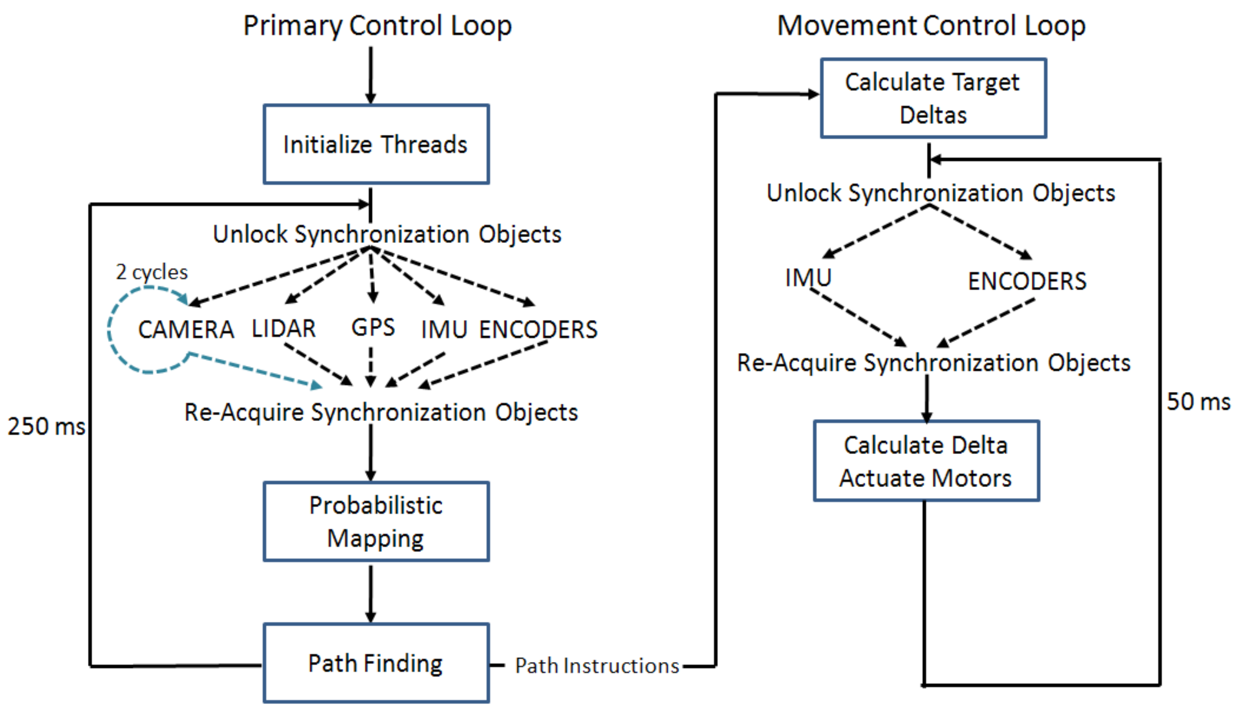

Figure 9: System overview and component integration

The basic idea of the system is to separate the robot into two loops; a primary control loop and a motor control loop. The reason this needs to be done is because certain tasks, like image processing and path finding, take large amounts of time to compute and bottleneck the system. By spinning the motor controller off into its own control loop, we can employ advanced movement tactics that are only possible when using a second loop.

For every iteration, the primary control loop passes the movement control loop the path instructions, primary loop runtime, and relative risk assessment of nearby objects. The movement controller uses the runtime and risk to calculate where the robot should be by the time the primary loop finishes again. It then enters a cycle of polling the IMU/Encoders and using that data to recalculate the delta between the current location and target location. It then proceeds to actuate the motors appropriately depending upon how large the delta and risk of nearby objects are. So, the overall effect of doing this is that the robot picks a local goal on the current path and makes a sweeping turn to it by quickly reading and reacting to its position and motion sensors. Once the primary loop refreshes, a new local goal is chosen.

Waypoint Navigation

In many ways the waypoint navigation challenge is approached as an extension of the course navigation challenge. Instead of trying to reach a moving goal line, as it does in the navigation challenge, Sparta places a moving goal line in the general direction of the closest unaccomplished GPS waypoint. As the GPS waypoint becomes close enough to the vehicle to be plotted on the probabilistic map, at a distance of roughly 33 feet, the moving goal line is replaced by a set goal circle. The A* algorithm then plans a path to that point. The path is dynamically updated as the vehicle records more detailed information with regards to the positioning of obstacles, allowing for alterations to the original path to circumnavigate vehicle obstructions. Once the vehicle reaches the goal circle at the GPS waypoint, the system removes that location from the list and repeats the process for the next closest GPS waypoint. The process repeats until all of the waypoints have been reached. To locate the opening in the Mesa screen the vehicle utilizes its LIDAR sensor to scan for a gap in the fencing where it can pass through. A future plan for Sparta is to implement the capability for the vehicle to be able to record its position as it is passing through the Mesa fencing to enable it to easily return back through the fencing without having to search for the gate again.

Electrical Design

The energy system of the vehicle is broken down into two separate subsystems, the motor drive system and the auxiliary or logic energy system. The motor drive energy system consists of two 12 V Absorbent Glass Mat (AGM) sealed lead acid batteries connected in series to supply two 450 W Brushed DC drive motors with 24 VDC. The motor drive batteries have a capacity of 28 Amp hours to allow for a minimum 30 minute runtime. Due to the high capacity required of the motor drive batteries, lead acid battery technology was the only type that fit within our budget. AGM lead acid batteries were selected for their ability to be placed in any position without leaking. The motor drive batteries are connected to a duel channel Roboteq AX3500 motor controller. The controller supplies the appropriate amount of power to the drive motors depending on the decision making teams' desired speed, and direction. US Digital 2500 CPR optical encoders are used to supply feedback to allow the motor controller to operate in closed loop.

The auxiliary energy system supplies energy to every component on the vehicle other than the drive motors. This includes the GPS, LIDAR, wireless E-Stop receiver, safety light, and the logic power to the motor controller. All of these devices can be run off of a 12 VDC supply. The auxiliary power source consists of two 6 VDC, 5000 mAH Nickel Metal Hydride (NiMH) battery packs connected in series to create 12 VDC. This battery technology was selected for the auxiliary power source because the capacity requirements for the auxiliary system is small compared to that of the 24 VDC system. For the same capacity, NiMH batteries have a more compact design and more desirable discharge curves that keep the terminal voltage higher throughout its life cycle. In transient conditions, the DC drive motors can draw significant surge current that lower the battery terminal voltage. The spikes of current could cause the terminal voltage of the motor drive batteries to drop below the vehicle's components required input voltage, resulting in shutdown or restart of auxiliary system components. It is imperative that components such as the LIDAR, GPS or the motor controller do not shutoff or restart to ensure functionality of the vehicle.

As a consequence of the LIDAR being mounted 14 inches off the ground and in the center of the vehicle, an auxiliary power box was mounted on the camera tower for extra space. The auxiliary power box on the camera tower houses switches and fuses for each component hooked up to the auxiliary power supply as well as the wireless E-stop receiver and push button E-stop. The fuses used for the components powered by the auxiliary power supply are glass AGC fast blow fuses.

The NiMH auxiliary battery packs were not placed in the auxiliary power box on the camera tower in order to keep the weight suspended to a minimum to prevent vibration of the camera tower. In order for the box to receive power, conductors were run from the battery packs at the base of the vehicle up to the box. In addition, three conductors were run for serial communication to the GPS, two conductors for the E-stop control circuit, and conductors for the safety light control, motor controller and LIDAR. For this purpose a single cable consisting of 12, size # 20 AWG twisted shielded conductors are run from the base of the vehicle to the auxiliary power box. This single 12 conductor cable cannot run directly to the power box because the camera tower has to be removable for transportation purposes. A Deutsch HD30 series receptacle and plug was used to allow for a simple, single weatherproof connection to allow the camera tower to be removed.

Safety

Safety has been an important consideration for the team this year. Several emergency stop mechanisms have been implemented that will bring the vehicle to a complete and sudden stop should the robot perform in a manner that is undesirable. There is a large, red button on the back of the camera tower that causes the motor controller to fault. Additionally, a wireless receiver was mounted on the vehicle that will also cause the motors to stop. The remote transmitter has a range of over 100 feet that has been verified through testing.

Factor of Safety was also a huge consideration in the design of the mechanical members of the vehicle. Finite Element Analysis was used to determine the stresses and deflections experienced by the members. In an attempt to ensure that a mechanical failure would not occur, a Factor of Safety of at least 5 was designed for all structural members. Figure 9 below illustrates a Factor of Safety of 9 in this torsional rigidity test where a 100 lb force was applied to the front corner of the frame while the other three corners were fixed.

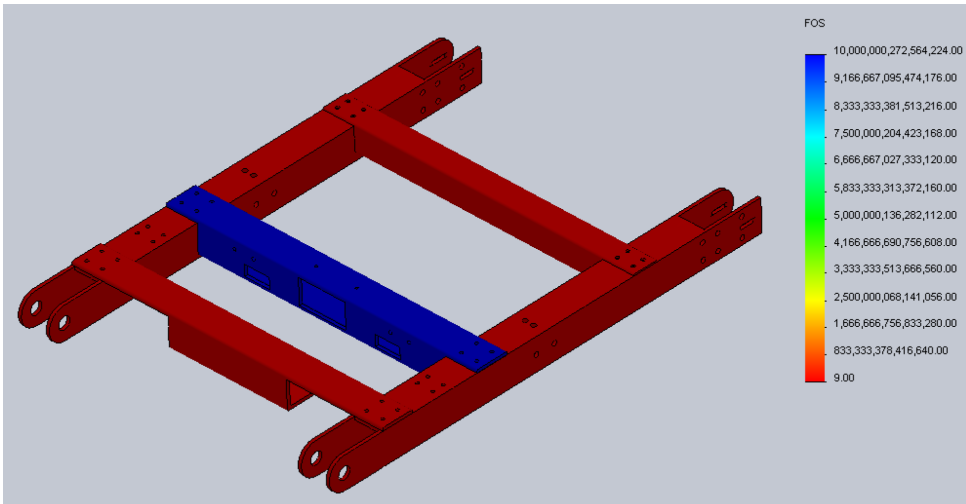

Figure 10: Torsional Rigidity Factor of Safety FEA

Durability and Reliability

Sparta is a very durable and reliable machine. The frame of the vehicle, especially when supported by the skid plate and under-side support beams, allows virtually no relative motion between members. Such rigidity allows the suspension system to react precisely as designed. The data from the LIDAR is highly reliable both in detecting the distance to an object and the angle at which the object is located relative to the vehicle. The LIDAR is capable at scanning every quarter of a degree and providing distance values accurate to within 30mm or just over an inch. Sparta is also reliable on all of the possible competition terrains. The larger area of ground contact gives Sparta greater traction than other wheel driven counterparts especially in scenarios involving wet grass or loose dirt.

Problems and Solutions

During the individual testing stage of our project, several problems were encountered that resulted in the robot functioning unpredictably, or not at all. First we encountered encoder problems, where the robot was not relaying accurate encoder data to the laptop. After some investigation, the team discovered that the encoder wires were positioned parallel to the 40 amp motor cables, which at the time were unshielded. This problem was resolved by shielding the motor cables and repositioning the encoder wires to allow them a path that was subjected to less interference. There were also problems with the tracks of the vehicle. The original track system allowed too much lateral movement. When the robot was required to do a high-speed zero-turn, the tracks would come loose from the road wheels rendering the vehicle immobile. The manner in which the tracks were connected was improved to decrease the amount of lateral movement and minimize the chance of the tracks coming loose from the rest of the suspension system.

Cost and Time Spent

To keep the cost of our project down, several components used on this year's vehicle have been recycled from the vehicle entered last year. Additionally, some of the more expensive components were donated to the team or borrowed from the school. The LIDAR and Laptop were donated from the school for use in the project, the GPS was recycled from the 2010 IGVC entry, the motors were spares left over from the 2010 team, and the camera tower was donated from one of Sparta's sponsors. Table 5 details an estimated cost for someone to reconstruct our project from scratch.

Table 5: Bill of Materials, with Quantities and Prices

Item	Price	Quantity	Total
Frame Materials	$200.00	1	$200.00
Treads	$50.00	2	$100.00
Motors	$85.00	2	$170.00
Encoders	$110.00	2	$220.00
LIDAR	$5,000.00	1	$5,000.00
Laptop	$2,000.00	1	$2,000.00
Motor Controller	$450.00	1	$450.00
Batteries	$90.00	2	$180.00
Suspension Materials	$500.00	1	$500.00
GPS	$2,000.00	1	$2,000.00
Webcam	$50.00	1	$50.00
Camera Tower	$500.00	1	$500.00
Electrical Components	$200.00	1	$200.00
Total			$11,570.00

The ten students on the design team for Sparta have spent two semesters working on designing, manufacturing, assembling, and testing the vehicle. Each student has spent more than 500 hours on this project. These 500 hours include everything from researching part availabilities and prices, spending time in team meetings, and constructing the robot. In total, the project has been worked on for over 5000 hours.