16th Annual Intelligent Ground Vehicle Competition
Georgia Tech RoboJackets Design Document
May 30, 2008 – June 2, 2008

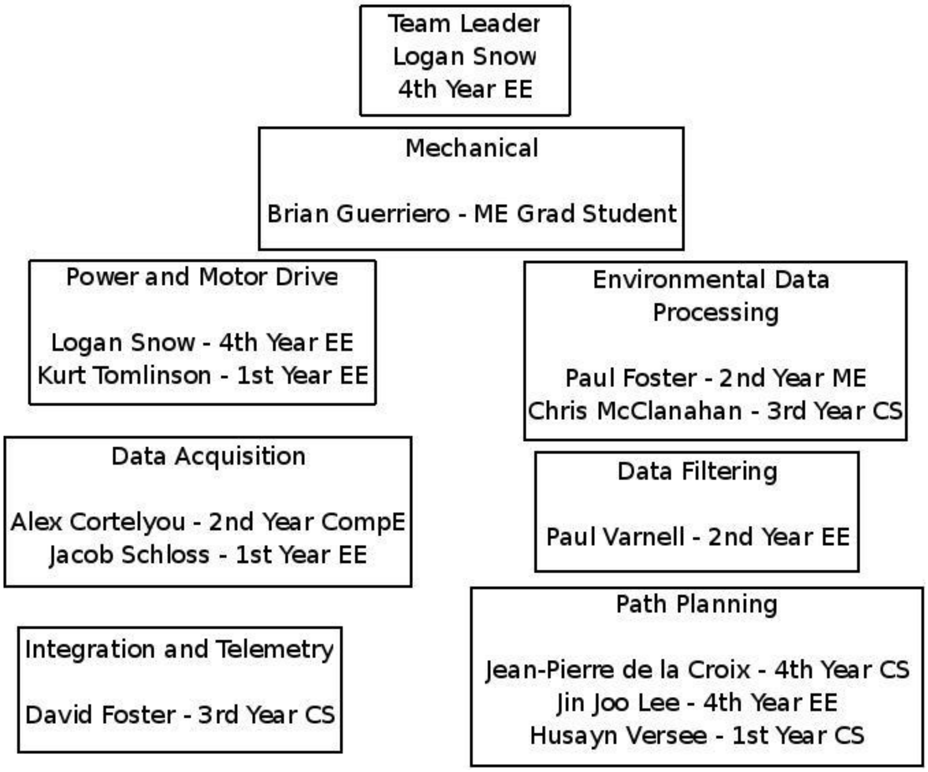
1. Introduction

The 2007 Georgia Tech RoboJackets submission to the Intelligent Ground Vehicle Competition (IGVC) proved to be a welcome success for the team, taking 7th place in the Autonomous Competition. Thus, the objective of the 2007 build season was a success - to create an unmanned vehicle platform that was powerful, maneuverable, and robust enough to handle all challenges of the IGVC. However, the system driving the platform, Candi, was simple, employing a reactive drive algorithm to avoid “dangerous” objects in the path of the vehicle’s camera. The objective of the 2008 Georgia Tech RoboJackets submission was to take Candi from being simply a “ground vehicle” and make her truly intelligent. Therefore, redesign of the electrical and software systems was undertaken to create a machine of far greater adaptability, awareness, controllability than the Candi platform. With great pleasure, the Georgia Tech RoboJackets completed the Candii unmanned vehicle system for the 2008 IGVC.

2. Team Organization

For the 2007 - 2008 design and build season, the RoboJackets IGVC team chose to take a structured approach to the project's management. As such, a pre-season meeting was held for veteran team members to accomplish several goals: exactly define the problem, determine the components required to solve the problem, and understand how best to divide up the ensuring work. As a result, six teams were created within the IGVC team: Power and Motor Drive Systems, Data Acquisition Systems, Integration and Telemetry Systems, Environmental Data Processing Systems, Data Filtering Systems, and Path Planning Systems. Each team was assigned a veteran member as a team sub-manager, reporting to the overall project manager. Thus, as the new season began, new members were provided guidance and divided up into the team best fitting each individual's interests. Figure 1 shows the full team that was assembled.

Figure 1. Layout of team responsibilities and personnel.



As opposed to a more self-determined process, a more deadline-driven design process was approached for the 2007 - 2008 year. From mid August through September, preplanning of how Candii should be, review of last year's weaknesses and strengths, orienting of new members, and new technologies research were pursued vigorously. By the beginning of October, technology solutions were finalized, yielding a clear picture of how Candii would look. Design work thus began concurrently on all systems in terms of: learning more about the selected technologies, determining which hardware or software would be required to implement the system, and step-by-step tackling of basic system needs. The design phase was allowed to continue into March of 2008, with monthly meetings of key officers to ensure that each group was on track with design goals. Implementation and testing were allocated to the remaining period until the IGVC, with no major new systems to be added. Overall, approximately 3000 man-hours went into realizing the upgrades required to create the Candii unmanned vehicle.

3. Vehicle Design - A Team-Structured Approach

3.1. Mechanical

Due to the satisfactory design of the mechanical drive base from the previous competition year, no significant changes were made to the platform. As in previous years, a front-driven differential drive system. This system has been preferred within RoboJackets for its easy implementation and controllability.

The custom frame was fabricated from thin wall one inch square steel tubing, tapering in the rear to allow for easier turning. Each piece was cut and welded from six foot lengths of stock. Aluminum bars were used to mount the drive motors.

The two differential drive motors were donated from the RoboJackets BattleBots team. They are NPC Robotics 1.7 hp 24V DC motors. These motors power the front two drive wheels, moving the entire vehicle. The traditional system of two swivel casters supporting the rear of the vehicle was avoided for several reasons:

1. Swivel casters, from past experience, have introduced significant amounts of wobble into the rear end of the vehicle while traversing through soft grass, making control and vision more difficult.

2. Zero radius turning with traditional casters induces slight forward or backward translation to the entire chassis as the wheel swivels around its caster angle axis, introducing uncompensated motion to the controllers.

3. To maintain a ground clearance of approximately 6-7 inches, casters of only 3-4” diameter would be required, much smaller than those desired for traversing grassy terrain.

The mechanical team decided to abolish the standard practice of utilizing swivel casters and implement one single 12” ball caster, providing one of Candii's largest and most effective innovations. This omni-directional ball caster solves the wobble issues, and can instantaneously compensate for changes in direction due to its zero degree caster angle. The 12” diameter ball will also leave a wider footprint in soft grass. The stainless steel ball is held in place by spring steel “net” of five small ball casters and four small roller bearings around the equator.

Since the main vision system camera was desired to be mounted as high as possible for the optimum field of view, chassis stability was placed as a top priority. To achieve stability, an independent suspension system was implemented on the front drive wheels. The inverted cantilever style suspension keeps the drive wheels behind the main pivot, effectively shortening the wheelbase, allowing for tighter turning. Suspension is provided through the implementation of two single acting pneumatic cylinders with coilover springs. Small valves were installed at each cylinder port and adjusted to introduce damping. Since the 5’ camera post is mounted to the rear of the vehicle, the main rear ball caster base-plate was also suspended by eight small springs to allow for shock absorption.

3.2. Power and Motor Drive

The responsibility of redesigning Candii's power, motor drive, and safety systems was assigned to the Power and Motor Drive Systems team. Innovations in system reliability, power quality, and protection were the key concerns of these upgrades.

The power system was divided into two separate power systems - the motor and logic power systems. The motor power system was composed of: two series connect 12 V lead acid batteries, the emergency stop relaying system, and the Open Source Motor Controller power amplifiers. The logic power system was designed to provide regulated 12 V, -12 V, 5 V, 11 V, and 19 V power to the various computer systems onboard. Care was taken to keep the two systems on separate ground loops, with all connections between the systems made either through magnetic (Hall Effect current sensing) or optical isolation. Since the system required such a custom array of voltages, the power regulation unit was designed completely in-house from National Semiconductor Buck and Sepic DC-DC regulators. All systems were carefully specified for their maximum current ratings and properly fused; additionally, metal oxide varistors were added this year to guard against unexpected switching transients.

The motor controlled system received a significant control upgrade. The original system featured a RoboJackets-designed microcontroller board which took motor speed command, converted them into PWM signals, and sent these to two OSMC's. While this system did operate well enough at the 2007 IGVC, the microcontroller board had unknown stability problems, often causing data to be read incorrectly. In the spirit of supporting open source electronics, the team decided to pursue Arduino Decibella control boards. These boards featured an AVR ATMEGA168 microcontroller, a full library of commands abstracting many microcontroller functions such as PWM outputs and ADC reads, and USB communication all at extremely low cost. With these new boards, several new features were implemented into motor control: a wireless USB joystick was designed to be used by the system with an analog joystick for backup, voltage readings of the motor batteries, current readings of the motors, and monitoring of the state of the emergency stop.

Understanding that even the best systems go awry, the emergency stop system was installed in series with the motor power system. Created from a garage door opener, this system implements a latching relay system that can be triggered by the wireless garage door opener or by pressing of the big red button on the back of the unmanned vehicle. With upgrades made to the motor controller, the state of the e-stop was also able to be monitored. This innovation was especially important, as the new data filtering and path planning systems could severely malfunction if they failed to realize that the motors were not supposed to be responding to their commands. Lastly, a software e-stop system was also added to turn off PWM output from the motor controller board.

3.3. Data Acquisition

The Data Acquisition group's purpose was to handle the implementation of all sensing devices and the hardware/software associated with them. Table 1 shows all of the data acquisition devices implemented by Data Acquisition and what data they provide.

Table 1. List of data acquisition devices and their measurement types.

Device	Measurement
AVT GUPPY F-036 COLOR CCD CAMERA	Raw color image data.
Hokuyo URG-04LX Laser	Position measurements of obstacles relative to the system.
Garmin GPS 18 5Hz	Latitude and longitude.
SR12 Absolute Shaft Encoders	Position relative to the starting position.
Analog Devices ADXRS300 Yaw Rate Gyro	Instantaneous angular velocity on two axes.
Freescale Semiconductor ±1.5g - 6g Three Axis Low-g Micromachined Accelerometer	Instantaneous linear acceleration on three axes.
Honeywell HMC1043 3-Axis Magnetic Sensor 3-Axis Magnetic Sensor	Cardinal directions.

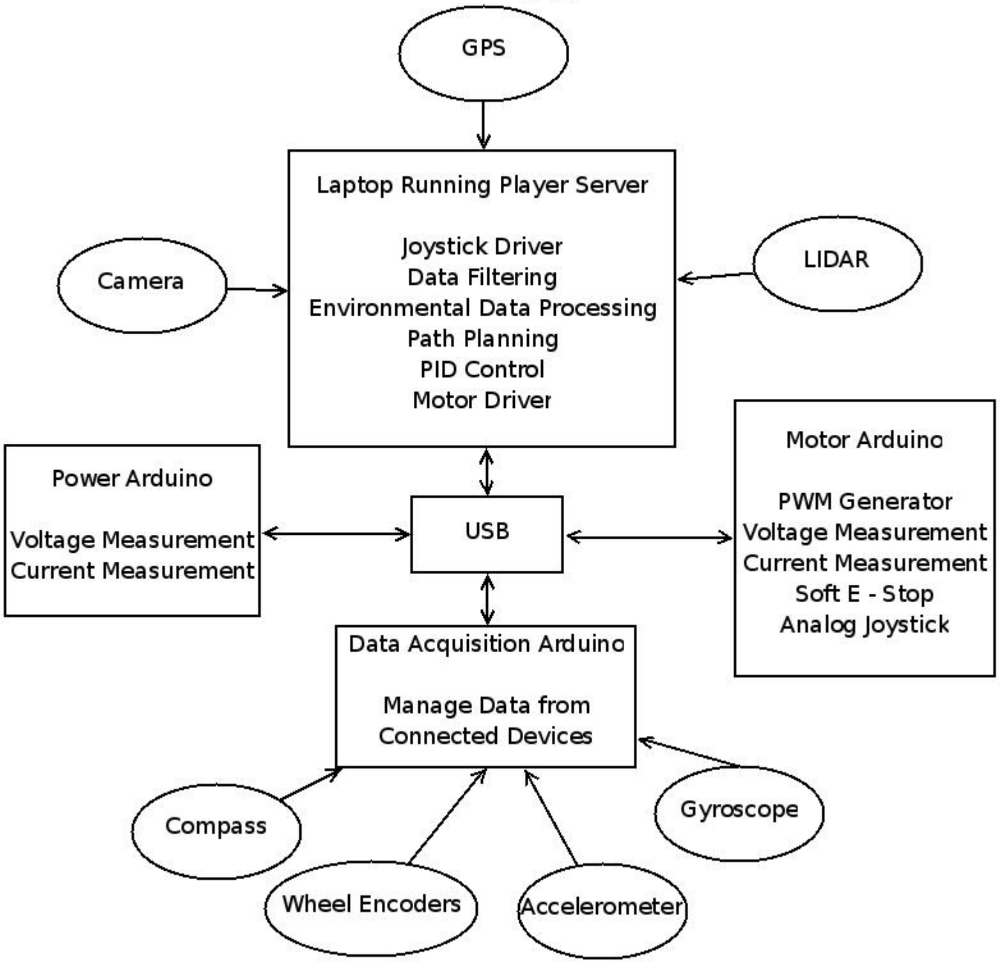
Data Acquisition chose to implement their systems using Arduino Decibella boards and a laptop. Knowing that all primary software systems would require intense computational power, a Dell

Laptop with the following specifications was chosen as the primary controller of the system: an Intel Core 2 Duo CPU clocked at 2 GHz, the operating system Ubuntu Linux, 1 GB of DDR RAM, a 74 GB hard disk, an NVIDIA GPU, IEE 1394 (Firewire), and ample USB ports. Figure 1 shows a map of all logic devices and their interconnections.

3.4. Integration and Telemetry

This group faced the difficult, yet essential task of establishing standards by which the other groups could operate and developing the system. Figure 2 on the following page shows the fully integrated system plan.

Figure 2. Diagram showing integration of system devices and processes.



As with the 2007 implementation of Candi, Candii's central control framework is the open source robot server Player. This server, developed by the open source community and by academia, abstracts multi-threaded processing, exact system signal control, and Ethernet transfer between computers. The Player server offers a powerful, easy, and innovative approach to systems integration. On each computer with an operating system, Player runs as a server providing device interfaces. Device interfaces are implemented as plugins to the Player server. For example, on the EDA Player runs a driver for the Hokuyo URG LIDAR. Processes running on any of Candii's computers may address the LIDAR and draw data from it as if the LIDAR were running on their respective systems. With a built-in queue implemented on each server, Player can handle traffic from a large number of processes trying to communicate with devices to which they have subscribed.

To simplify communication between computers and to allow device drivers for Candii's different hardware components to be developed separately, the electrical/programming team decided to write much of the code as plugins for the Player robot server. In addition, the Player distribution already had a standard driver written for selected LIDAR, so time saved in writing that device driver. However, Player only had a standard camera driver for cameras that are compatible with the raw1394 library. Unfortunately, the camera did not work with the raw1394 library, although it was compatible with the dv1394 library. Thus, a custom dv1394-based camera driver was written in order to talk to the camera using the IEC 61883-2 DV (Digital Video) protocol.

On the embedded board Player server, the ability to communicate with the GPS, compass, accelerometer, and gyro is abstracted as a device driver plugin. Other systems may query the device plugin “IMU” for the following information: global position in latitude and longitude coordinates cardinal direction in degrees, and current position relative to a starting point. By the time of the competition, some of this data will likely have changed due to removed or added features.

3.5. Environmental Data Processing

Data on white lines and orange barrels (for a sanity check with the LIDAR barrel data) is provided by a powerful implementation of the perspective transform followed by white blob and barrel identification and the Hokuyo URG LIDAR by the Environmental Data Processing team. The goal of the perspective transform is to reverse all of the changes that occurred to an image from the time light leaves the ground to when it is detected by the camera. Once the transform has run, all of the points in an image have been correlated to match their real Cartesian coordinates away from the camera. Any points in the transformed image that appear as white lines are added to an environment map; additionally, detected barrels are checked against LIDAR data to ensure that an agreement exists between the two methods. If so, then LIDAR obstacle data is placed into the same map as the camera line data. This method ensures robust, reliable data acquisition from the environment.

In designing the perspective transform, many points had to be considered. Two types of distortion had to be considered – perspective distortion and lens distortion. These distortions had to be inverted in the reserves order of their mentioning. While most camera lenses approximate that of a pinhole camera (a gnomical lens), an equidistant full frame fisheye lens was used on the camera as it has the advantage of wide angle of view without the high monetary cost of a wide angle gnomical lens. This lens caused barrel distortion, where rather than the magnification of an image being constant throughout, it increases with radial distance from the center of the image. By measuring several lens parameters, pincushion distortion was introduced to exactly negate the barrel distortion.

Next, the perspective distortion was negated to “transform” image coordinates to real coordinates. First, the matrix calculations normally used to produce perspective distortion (giving the appearance of perspective) in virtual 3D environments were mathematically inverted. A factorization of the matrix was created by a lack of depth information in the image to those created by a lack of height information from the ground to produce a transformation matrix that correlated to the perspective distortion created by the camera lens.

The perspective transform was designed to take advantage of several features offered by OpenGL, the open source graphics library. Since one of the primary functions of graphics cards is performing perspective transforms, utilizing OpenGL allowed the transform to be processed completely by the GPU of the EDA laptop. Also, data can be dumped directly from the GPU into RAM, leaving the CPU free to process maps and other algorithms. By using the GPU, the lens and perspective corrections were able to process approximately 50 frames -per-second in real time.

Since proven legacy code existed for the detection of barrels and white lines, the software was ported to work with the new system. In order to detect barrels for correlation with LIDAR data, a red green blue (RGB) color space containing data of interest was loaded. Then, the green sample of the color space was subtracted from the red sample. The resulting data represented a measure of the orange intensity of any given subset (pixel) of the color space. Since IGVC barrels were determined to be of a very high orange intensity, pixels were then compared against an experimentally-determined threshold. Pixels that had orange intensity values greater than the designated threshold were then passed to the barrel detection algorithm; pixels failing to meet this criterion were discarded.

White blob detection required more processing to ensure accurate detection. White was determined to be too difficult to process by only thresholding high intensity values in the RGB color space. From experimentation with the camera, many high intensity subsets of the image space were found to appear close to white. In order to solve this problem, conversion to a hue saturation brightness (HSB) color space before processing was proposed. By converting the input image to HSB, intensity values of saturation (purity of the color) and brightness (intensity of the color) could be assessed. The measure of saturation was especially important, as a great deal of the uninteresting white data in images appeared somewhat grayed by mixing with other intense colors. Pixels values with saturation less than a certain threshold and brightness above a certain threshold were then considered line or pothole objects.

The culmination of the Environmental Data Processing group's efforts is shown in Figure 3.

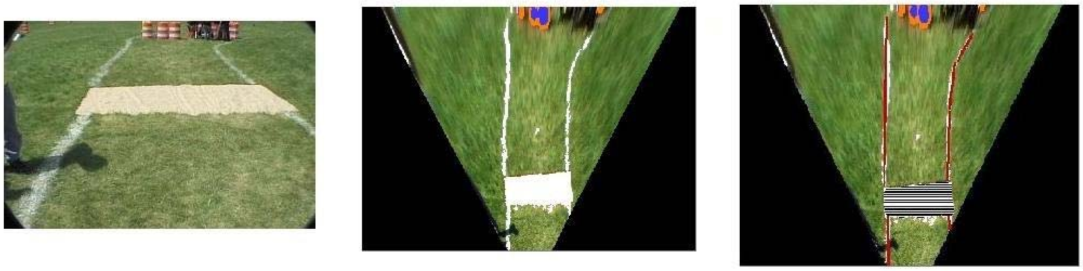


Figure 3. Progression of processed images.

Starting with a standard input image, part one shows the barrel distortion applied to this image in order to remove lens pincushion distortion. Next, the perspective transform is performed, yielding an "overhead" image of the terrain. Lastly, processing algorithms are applied, finding white lines, barrels, and the sand trap in the image. These powerful algorithms extend to finding pot holes and broken lines as well.

3.6. Data Filtering

The Data Filtering team takes the inputs from Data Acquisition, removes intrinsic noise in the data, and creates a good estimate of world position. The unmanned system is designed to determine its global pose using the data from a number of sensors. Triple axis MEMS accelerometers, gyroscopes, and magnometers placed on a custom designed PCB provide some inertial and positional information. This board is connected to an Arduino which reads analog values from the sensors and sends them to the main computer over USB. Two wheel encoders are also connected to the Arduino through an I2C interface. A GPS is connected directly to the main computer via RS232, but could be easily connected to the Arduino if desired. Most of the processing is done on the main computer, but the algorithms used are efficient enough that they could run directly on the Arduino. When designing this system, some uncertainty was present as to the processing power that the filtering would require, so the faster main laptop is used instead of an embedded processor. In future designs, this system will likely have completely independent hardware.

The information from the sensors is combined using an extended Kalman filter, which is a standard algorithm for estimating an unknown state using noisy measurements. The Kalman filter assumes that noise in the sensors is normally distrusted (Gaussian) and is linearly related to the state being determined. The extended Kalman filter does the same except that it linearly approximates nonlinear relationships between the sensors and the state of the system. There are other ways to linearize these relationships such as the unscented Kalman filter, but these take more time to compute and only modestly increase accuracy. There are also algorithms that do not assume normal distributions of noise and can natively use nonlinear relationships, but these use significantly more resources and are not generally useful for this simple of an application. If a sensor that has significantly non-Gaussian noise were used, such as a camera, this type of filter would be required. Before coming to the Kalman filter, each sensor is filtered independently to remove outliers, remove noise specific to that type of sensor, and equalize the sample rate of all of the sensors. This eliminates the sudden jerks caused by large jumps in GPS position and similar glitches in other sensors.

There are many commercial products that do exactly the same thing as this system, but most of them cost several thousand dollars and use closed source software. In parts, our system cost approximately 250 dollars and is nearly identical. The RoboJackets are considering releasing our work on this system, both hardware and software, under an open source license. This will give other robotic hobbyists like the RoboJackets the opportunity to use a very accurate positioning system at a low cost.

Using more sophisticated positioning methods that involve mapping the environment with camera and LIDAR has been considered, but this simpler solution was pursued for several reasons. Time and resources are fairly limited. Also, due to the current nature of the competition, path planning systems do not require long term memory of the environment, since there are no large or obscured loops or dead ends. This greatly reduces the advantage a more complicated mapping solution would have over a simple positioning system. In the future, the Data Filtering team may experiment with these methods, but it will be mostly for academic reasons.

3.7. Path Planning

The Path Planning team created a system that would takes in a map of the environment and return a path for the vehicle to traverse. Such a world map is represented as an occupancy grid map; this map is a matrix where each grid point is the centroid of a unit area with a 0 representing an unoccupied space and a 1 representing an occupied space (obstacle). An algorithm takes in the world map and produces a cost map, which represents the cost from the current location to the goal. This cost is the metric by which the path planner determines the optimal path by finding the “cheapest” route. Another algorithm then adds a bias around the obstacles to allow some tolerance for the vehicle's size needs. The route creation is accomplished by a third algorithm that takes in the cost map and produces a vectorized line representing the path from the starting location to the goal location.

This cost map algorithm is a "sweeper", starting at the source location, the Manhattan distance to each point on the grid is calculated – the distance to that point is not calculated if an obstacle lies at that point. The Manhattan distance, which represents the absolute sums of the horizontal and vertical distances on the areas of the grid, was chosen for the first test due to their relative simplicity to Euclidean distances. However, since the Euclidean distance (which represents diagonal traversals) was also needed, a geometric approximation using a quadratic formula was created to allow diagonal as well as horizontal and vertical paths to be considered.

Once a proper cost map was generated, finding the optimal trajectory was a relatively trivial task. The path algorithm simply calculates the gradient at the point of its location and follows the downward-most slope. As a final result, Figure 4 shows the same cost map with several obstacles placed randomly. The vehicle, its geometry represented as a rectangle or a triangle, maneuvers around the course.

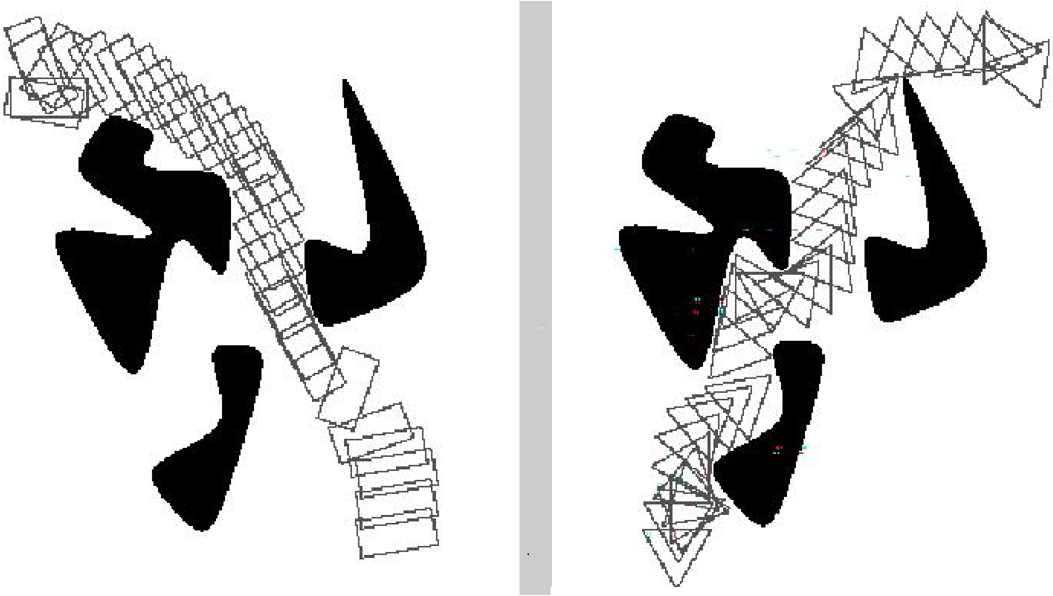


Figure 4. Path traversal for some sample geometries.

4. Predicted Performance

Although the system is not entirely functional at the time of this report's generation, a good prediction can be made based on operation at the previous IGVC and simulation of the control algorithms. The powerful Candii platform has the ability to move as fast as seven mph, but in practice is limited to four mph. Her speed varies based on the situation of obstacles, however. Candii is also highly capable of climbing ramps and with a properly planned path can scale over them without losing much speed. Since the final path is constantly being calculated, Candii can react to new situations with tenth of a second speed. Rapid changes in camera image, however, can be problematic, so turning speed is typically limited. In terms of battery life, she can run for a day of heavy testing (four to five runs an hour) for about three hours before batteries must be recharged. Based on camera and LIDAR tests, she should be able to see all orange and white obstacles up to about four meters away; other obstacles are limited to one meter in bright sunlight due to the LIDAR's nature. As shown at the previous IGVC, the vehicle is more than maneuverable enough to fit into tight spaces and handle traps. She is also programmed to turn and search for new paths should she reach a corner trap. Navigation waypoint accuracy is limited by the GPS, and tends to find waypoints of 2 meter radius without a problem. Smaller waypoints are not consistent, however.

5. Estimated Cost

Table 2. Estimate of system cost.

Item	Estimate of Cost
Mechanical Accessories	$300
Steel	$200
Aluminum	$200
Custom Electronics	$400
Motor Drivers	Donated
Laptop	Donated
LIDAR	Donated
Camera	$580
Camera Lens	$133
GPS	$200
Other Sensors	$400
Garage Door Opener	$80
Arduino Boards	$105
Total	$2,598