The College of New Jersey

International Ground Vehicle Competition 2007

TCNJ's Autonomous Vehicle Team Presents:

A.M.B.E.R.

Adaptive Mobile Basic Environment Robot

New Jersey Autonomous Vehicle

Team Members:
Joseph Custodio, Michael Esler, Nathan Kupp, Matthew Pettineo, Kevin Wolfe

http://www.tcnj.edu/~njav

1. Introduction

The 2007 New Jersey Autonomous Vehicle (NJAV) Team is proud to present their entry in the 2007 IGVC, the Adaptive Mobile Basic Environment Robot (AMBER). AMBER is the product of two semesters of development by five seniors at The College of New Jersey under the guidance of two faculty members. While the mechanical frame was inherited from the previous year's team, substantial revision of the mechanical platform was undertaken by this year's NJAV team. In addition, nearly all of the electronic systems were reconfigured, the software was completely rewritten, and a new laser rangefinder unit was implemented, collectively representing considerable progress for the NJAV project. This report documents key points of improvement in mechanical, electrical, and software components of vehicle design during the 2006-2007 academic year.

2. Design Planning Process

Due to the limitations on time and team members, an iterative approach to design was selected. At the beginning of the project it was recognized that to have ample time to complete the project with only five team members, the design and production phases of the project would need to be undertaken concurrently. Therefore, team members were assigned areas of focus as shown in Figure 2.1, and work began immediately at the start of the academic year.

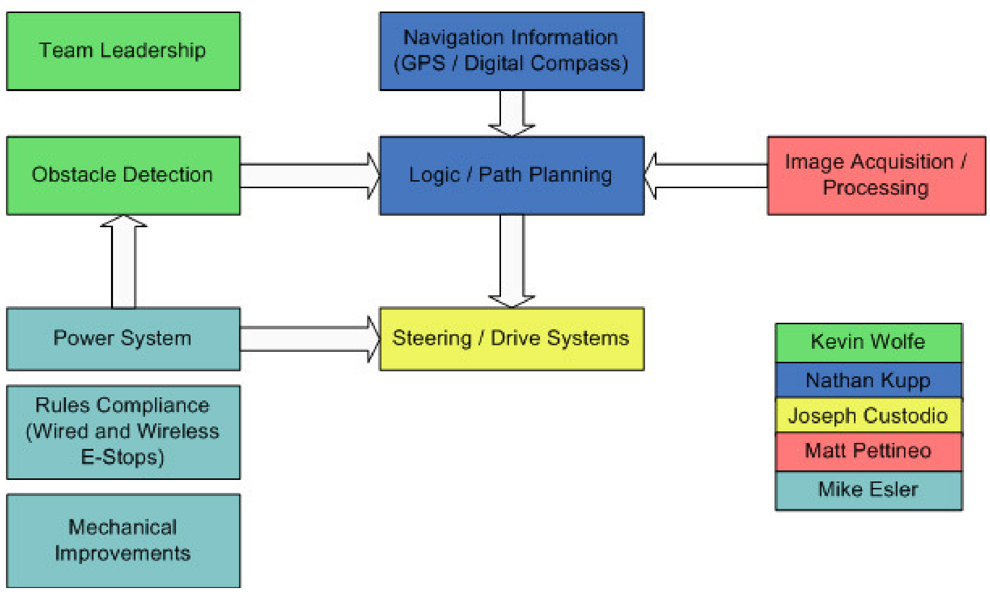

Figure 2.1 – Team Division of Responsibility

This process of continuous improvement proved extremely effective in developing reliable systems. At the conclusion of the design and production phase conducted during the fall semester, the testing phase of the project was begun with a similar iterative approach. This ultimately proved very effective in enabling construction of a reliable, well-tested vehicle.

3. Mechanical/Systems Integration

The mechanical design for AMBER had several primary objectives. The previous year's team left an existing mechanical platform, so one of the first objectives for this project was to analyze and adapt the existing NJAV mechanical design. Also additional components were required, which needed to be integrated with the vehicle. Finally, as described in the Design Planning Process section above, the vehicle was tested and adjustments were made as necessary.

Chassis

The only modification to the chassis that was necessary comes as a result of the 2007 team's acquisition of a laser range finder. To mount the laser rangefinder, the front beam that traverses the width of the vehicle was moved back and two supports were welded into place. A 1" diameter pipe was also welded across the front of the chassis for structural and convenience purposes.. Screen shots of the old and new design can be seen in Figure 3.1.

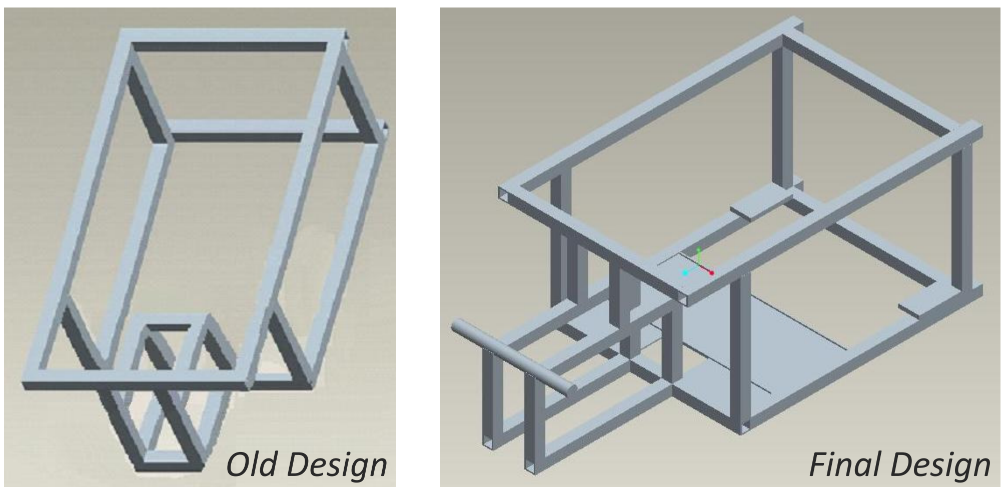

Figure 3.1 – Chassis Modeled in Pro/Engineer

Suspension System

Another well-designed component of the inherited frame is the suspension system. This system effectively reduces vibrations throughout the vehicle, thereby protecting and promoting the integrity of sensitive onboard electrical components. The 2006 NJAV team spent a good deal of time designing this suspension system, and therefore this system was kept without significant changes. Figure 3.2 shows two pictures of the fully assembled suspension system.

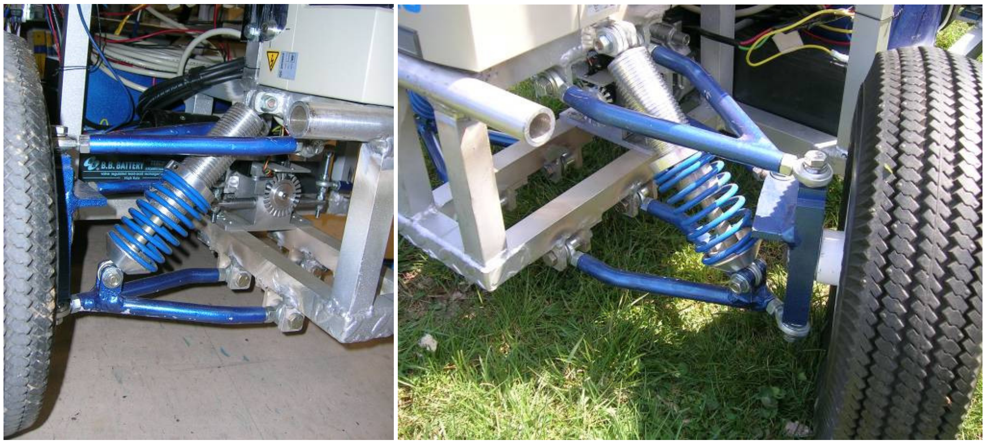

Figure 3.2 – Fully Assembled Suspension System

Drive Motor

The motor used to drive the vehicle is a UNIPAC Drive System Motor, model MK35HE. This motor is an entirely sealed system, with lifetime lubrication. It uses no belts or chains and is capable of a 175 RPM output. The motor has been used in previous NJAV designs and works very well for the needs of the project (See Figure 3.3).

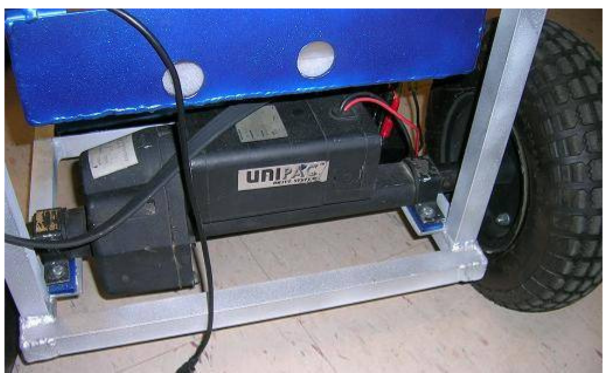

Figure 3.3 – UNIPAC Drive Motor

Steering Motor with Rack and Pinion and Centering System/Optical Encoder

The steering motor used in the NJAV was donated to the team several years ago. It is believed to be a windshield washer motor from a Volkswagen car. The motor is bolted to an aluminum housing setup that allows its axle to protrude outward from the center of the housing, which is connected to a rack and pinion system to control the front wheels of the vehicle.

Two optical encoders were used in conjunction with this steering mechanism. The first was configured to provide the software with rotation information with an encoder wheel affixed to the axle of the steering motor. The second was attached to provide centering information by connecting a centering L-bracket to the rack of the steering system. Half the L-bracket was removed, enabling the computer to determine whether the car was turned left or right and turn the steering accordingly to center the vehicle. Figure 3.4 below displays the completed steering mechanism with the optical encoders.

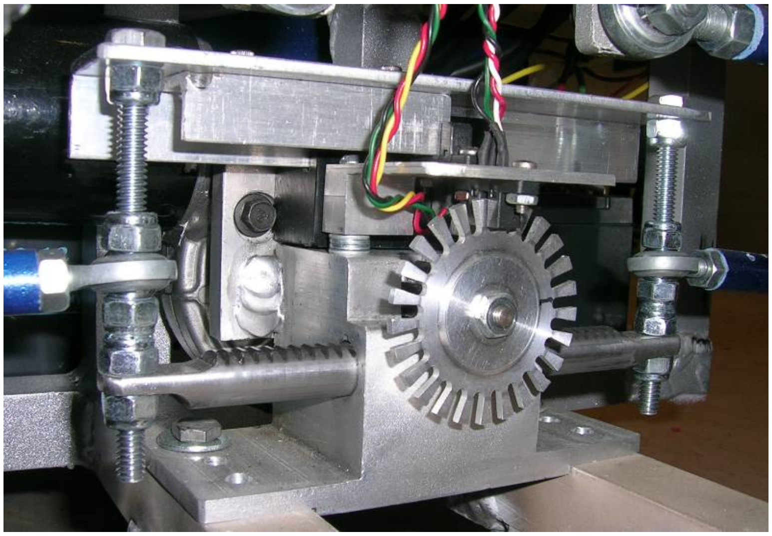

Figure 3.4 – Fully Assembled Steering Motor with the New Centering System Attached

Feedback System

In an effort to attain accurate data corresponding to vehicle speed, a feedback system was implemented by the 2007 NJAV team. Figure 3.5 below displays the feedback system implemented.

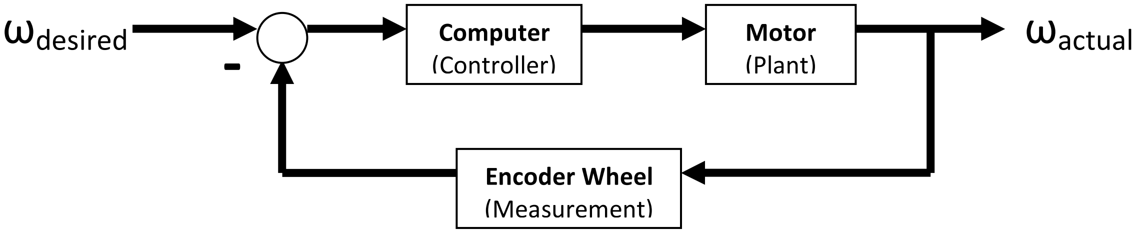

Figure 3.5 – Negative Feedback Loop Logic Diagram

The feedback system tells the computer to increase or decrease the motor speed as certain situations dictate. In the presence of hills or declines, the feedback system serves to maintain a constant speed of the vehicle in an effort to ensure optimal performance. Such a feedback system is implemented within the vehicle via an encoder wheel, collar, opto-switch, and computer. The encoder wheel was manufactured by the team and bolted to a collar that was subsequently placed directly onto the rear drive axle, rotating in unison with the axle. An optical encoder was attached over the teeth of the encoder wheel to permit measurement of the rotational velocity of the rear axle. The assembled encoder wheel system is shown in Figure 3.6.

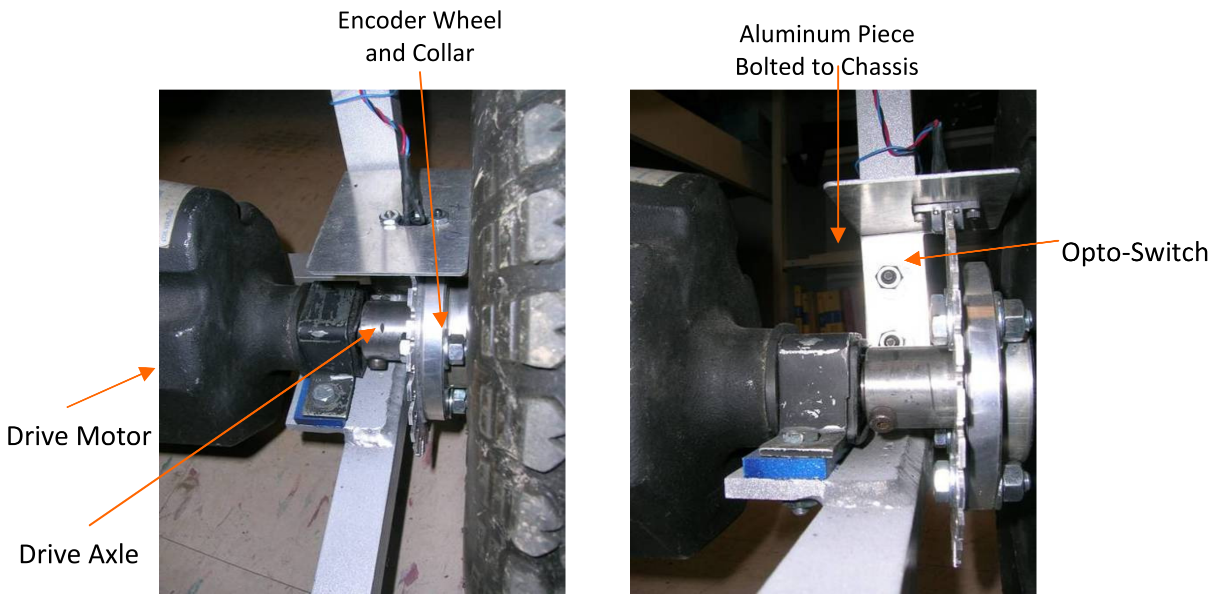

Figure 3.6 – Assembled Encoder Wheel/Feedback System

By attaching the optical encoder leads to the computer a vehicle feedback loop was created. By measuring the rate of rising and falling edges produced by the opto-switch and encoder wheel, the computer can accurately determine the speed of the vehicle. With a desired output velocity established, the feedback loop uses the difference in desired output and actual output to control the motor. This feedback system was tested and proven to work on short grass, long grass, inclines, and declines.

Power Systems

Onboard power systems for the 2007 vehicle were redesigned in anticipation of supporting a new set of electrical components. By estimating the power consumption of the electronic devices located within the vehicle a power system was designed to provide adequate electricity to all components. Of first importance was the improvement of the existing power system used to provide power to the steering and drive motors. A triple-pole, double-throw switch was implemented to enable the batteries to either be connected to 12-volt charging terminals in parallel or the motors in series (24-volts).

The newly developed and installed setup allowed for charging of all the batteries while they remained onboard and untouched. By setting the switch to the "charge" position the circuit placed the four batteries in parallel, which permitted all four to be charged simultaneously, and the switch also isolated the motors from the charge current. When the switch was set to the "on" position, the charger leads were disconnected and power was routed directly to the motors, with a 10 amp fast blow fuse for the steering motor and a 15 amp fast blow fuse for the drive motor. In this configuration the motors both received 24 volts from the battery configuration of two pairs of 12 volt batteries in series. With the sealed lead acid battery individual characteristics of 12 volts and 17 Amp hrs., the "on" configuration created a total of 34 Amp hrs. through the systems leads. An identical, discrete system was employed to power the electronics systems – two power systems were employed to provide isolation of electronics from motor transients.

To ensure sufficient battery life, the battery life of the newly acquired batteries was analyzed. The assumed current draw for all onboard electronics was computed to be 1.51 amps (note that this does not include peripherals directly connected to the laptop or the laptop itself, since this operated on the laptop battery). With this current draw value and a minimum voltage output of 12 volts, the battery discharge time was estimated to be just over two hours, based on manufacturer discharge specifications. Testing of the vehicle confirmed this discharge time at approximately two hours.

Water Proofing System

Another important mechanical modification to the vehicle was the installation of a weatherproofing system for the vehicle. The design chosen was a marine-grade vinyl covering secured to the frame via bolts and hook and latch strips. The video camera unit was protected individually using a combination of aluminum shielding and vinyl. The finished design successfully shields the camera from light precipitation as can be seen in Figure 3.7.

Figure 3.7 – Vehicle and Camera waterproofing

Miscellaneous Mechanical Improvements

Three additional mechanical modifications were introduced to the vehicle. The first component created was a pushbutton tower for the local emergency stop. A weatherproof structure was created to both protect the switch as well as meet the minimum button height requirements set by the IGVC. The second improvement to the vehicle was the design and installation of a payload tray. IGVC rules state that an 18” x 8” x 8” box, weighing 20 pounds, is required to be carried by vehicles during competition, necessitating a payload tray. The third mechanical design added to the vehicle was completed in an effort to increase user convenience. With a sliding tray, the user can access the laptop easily without needing to remove the weatherproofing system. Figure 3.8 displays these mechanical improvements.

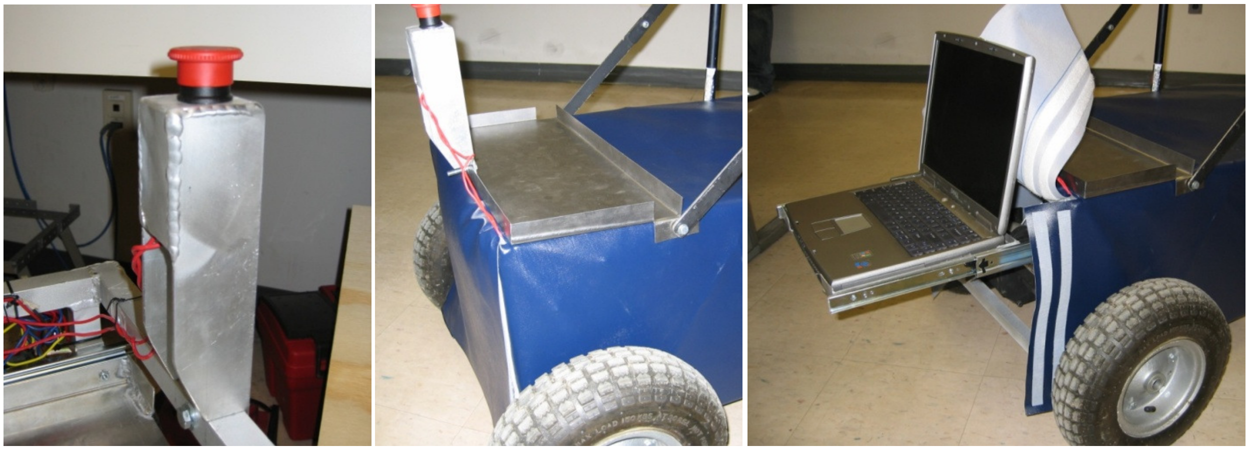

Figure 3.8 – Miscellaneous Mechanical Improvements

4. Electrical Systems

The electrical systems of AMBER are all connected to a central Dell laptop computer. A National Instruments DAQmx device connects to this laptop and provides a USB interface to the optical encoders required for the steering and drive feedback systems. It also provides interfacing to both the drive and steering motor controllers.

Drive Motor Controller

A Curtis PMC model 1208C motor controller is used to control the drive motor. This motor controller is rated to 70 Amps for a nominal input voltage of 24 Volts. Via computer input, this controller functions to start and stop the motor, control the motor speed, select forward or reverse rotation, regulate the torque, and protect against overloads and faults. The motor controller is designed to be controlled by an external potentiometer connected to three terminals on the board, along with an enable line. The enable line was connected directly to the National Instruments DAQmx for direct control in software. Since it was desirable to control the vehicle speed in software for the feedback system described earlier, the potentiometer used was a digital up/down potentiometer with two control lines which could be used to set the value in software through the DAQmx. An interface board was created with an opto-isolator along with the digital potentiometer unit to interface the motor controller to the DAQmx.

Steering Motor Controller

A new motor controller was constructed for this project. This controller utilizes two inputs from the DAQmx, along with BJT transistors, resistors, comparators, and an opto-isolator. The opto-isolator implements one-way movement of the voltage and eliminates potential spikes and transients returned from the motor that could damage the computer-hardware interface. This team-designed motor controller features two relays. The first implements a simulated H-bridge to control the direction of the motor rotation. Switches inside the relay are set to one of two positions that correspond to either forward or reverse motor rotation. The second relay enables regenerative braking, which aids in the accurate centering of the vehicle's wheels by limiting overshoot. A schematic of the assembled controller is displayed in Figure 4.1.

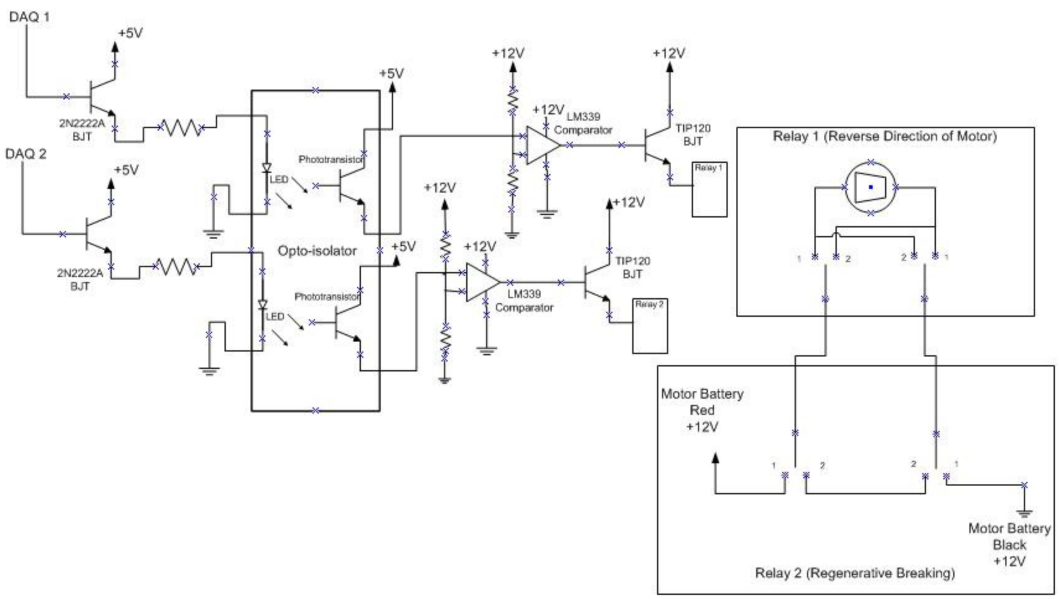

Figure 4.1 – Steering Motor Controller Circuit Diagram

Sensor Systems

A Garmin Geko 301 GPS unit was used to provide position data, and a Silicon Laboratories C8051F350 unit was used to provide heading data. Also implemented was a SICK LMS 291 laser rangefinder unit to provide obstacle data to the laptop. All three of these sensors interfaced directly with the laptop computer by simulating RS-232 connections over USB ports. Finally, a Basler 302fc video camera was implemented to provide color image data to the laptop to enable line and pothole detection and avoidance. This sensor interfaced directly with the laptop as well, utilizing an IEEE 1394a connection.

5. Software Systems

The software architecture of AMBER for this year's entry had several key areas of focus. The first of these areas was the path planning system. Also, the success of the path planning system was facilitated by a supporting image processing system, a second area of software development focus. To implement these systems, the software systems of the previous year's team were completely replaced. The first of the changes made was changing from MATLAB to C++, as the speed of C++ makes it well suited for time-critical applications. The autonomous vehicle will be moving at a moderately fast pace through the competition course. Because of this, quick processing speed was deemed critical to ensuring that the software's knowledge of the environment and processing of sensor data to select a best-path forward occurred in real-time. The previous software implementation in MATLAB required the vehicle to stop and process every few feet, expending considerable time unnecessarily. Speed improvements aside, the switch to C++ was also done out of necessity, as the interfacing libraries for the new camera were available for C++ only. Finally, it was apparent that the car needed a more effective navigation algorithm to compete successfully in the IGVC. The algorithm implemented by the previous year's team was a simple avoidance algorithm, selecting from three steering angles: left, forward, and right, based on obstacle detection data. This was deemed overly simplistic for a competitive entry in the IGVC.

After replacing the code base with C++, a completely new navigation system was devised. This system was designed with three discrete subsystems, which could be activated or deactivated as needed, as shown below in Figure 5.1. For the navigation challenge component of the competition, the GPS and Path Planning systems are activated, enabling the car to navigate to GPS waypoints while avoiding obstacles. Note that boundary lines are not present for the navigation challenge, eliminating the need for the Image Processing subsystem. For the obstacle avoidance challenge, the GPS subsystem is disabled and the other two subsystems enabled. The obstacle avoidance challenge contains no GPS waypoints, and therefore only the Path Planning and Image Processing subsystems are required.

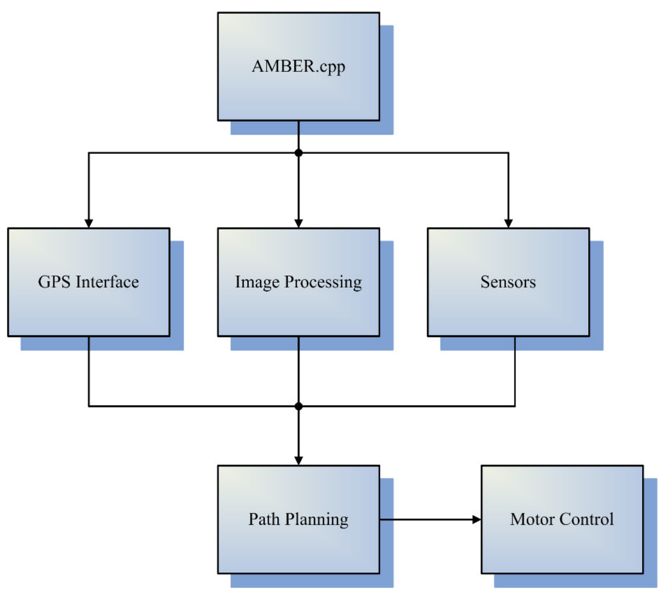

Figure 5.1 – Overall Software System Architecture

Path Planning System

The fundamental structure of the path-planning algorithm created for this project is based on an arc generation and selection process. To choose a best path, a series of fourteen arcs representing possible paths for the car to take are generated. These arcs are based upon the Ackermann steering characteristics of the mechanical vehicle, as well as the minimum turning radius the steering will allow. Within these limitations, nine arcs were chosen as a default set, with a six-foot arc length, and four arcs were chosen for close range maneuvering at three feet in the event that the long-range arcs are blocked. The high-resolution laser rangefinder implemented on AMBER, along with the image processing software system described in the following section, provided sufficient data about the surroundings to the path planning system to enable the car to select between these fourteen arcs effectively. Figure 5.2 below shows a diagram of this arc generation process.

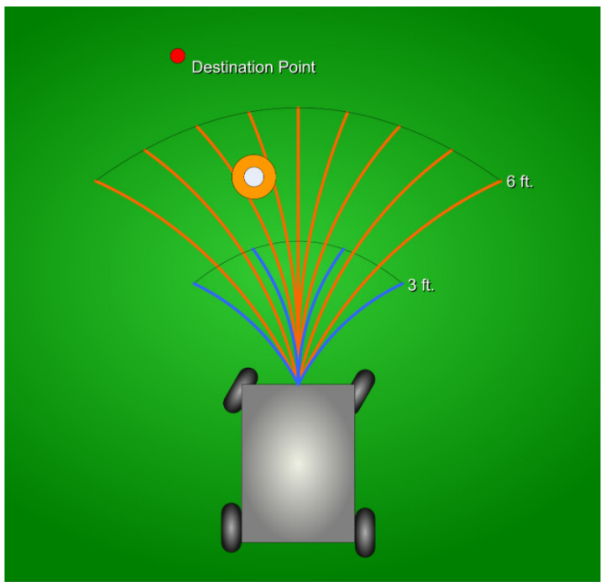

Figure 5.2 – Arc Generation Process

After generating the arcs in front of the car, the sensors are polled to determine relevant obstacle data. If the navigation course is being traversed, the GPS/Digital Compass interface class is also polled at this point to determine the car destination point based on GPS and digital compass sensor data, as represented by the small dot at the upper left-hand side of Figure 4.2. Note that if the course being traversed is the obstacle avoidance challenge, the destination point is always directly forward of the car at the top center of the map of Figure 5.2. Also, if the vehicle is in obstacle avoidance mode, the Image Processing subsystem will be checked to determine the location of any white boundary lines or potholes. Finally, based on the destination point, obstacle/boundary/pothole information, and the current angle of the car wheels, a weighted score is generated for each of the arcs. The equation for this score is shown below in Equation 5.1, where A, B, and C are the experimentally determined weighting coefficients for the destination, wheel angle, and obstacle sub-scores, respectively. To determine the best values for the coefficients A, B, and C, the vehicle was run through a short test course. C was chosen to be a value much greater than A or B to ensure that if an obstacle, boundary, or pothole was detected intersecting an arc, that arc would not be chosen. A and B were then chosen to be 150 and 10, respectively, through iterative testing of the vehicle through the short test course. Also, if none of the nine six-foot arcs are clear, the car will switch arc-sets to the five three-foot arcs and re-attempt path planning using the same process. This process is then repeated several times a second. Experimental testing indicated that the path planner executed approximately 10 times per second. Because of this rapid pace of path selection, the movement of the car remains fluid despite the finite and discrete set of arcs being selected from, and this fluidity is maintained both in simulation and with experimental testing.

$$\text{Score}_i = \sum A \cdot \left(Dest_{x,y} - Current_{x,y}\right)_i + B \cdot \left(Arc\theta - Current\theta\right)_i + C \cdot Obstacle_i$$

Equation 5.1 – Arc Score Generation Equation

Image Processing System

A computer vision software architecture was necessary to take advantage of the video camera. This system is primarily responsible for acquiring image data, as well as providing the path-planning logic with the location of the boundary lines that designate the edges of the competition obstacle course. To improve upon the vision system designed by the previous team, the decision was also made to select a new camera with properties better suited the specific needs of the vision system. A Basler 302fc video camera was donated by Basler Vision Technologies for use with the vehicle. The camera produces RGB images at a standard resolution of 640 x 480 pixels. After consideration, it was determined that the Basler camera would provide a significant improvement over the general-purpose still-image camera utilized by the previous year's design and was therefore a suitable choice for the vision system.

Using C++, software algorithms were developed for detecting lines and potholes in the images acquired from the camera. These algorithms utilize several image processing functions to specifically chosen to separate white and yellow portions from the expected green background provided by the grass. The first step of this process is to downscale the 640 x 480 image from the camera to a size of 160 x 120. This greatly reduces the amount of time required for image processing, thus allowing for a level of performance adequate for the software's overall refresh rate. The second step is to apply a Gaussian blur to the image. The step removes sudden discontinuities in the derivative of the image due to noise, which can adversely affect the edge-finding process.

The third step of the algorithm begins image segmentation by subtracting a weighted green component from a weighted blue component of each pixel. Manipulation of the green and blue channels in this fashion enhances white portions while diminishing green background portions.

Under typical conditions, manipulation of color channels results in levels of noise unsuitable for direct integration into the path-planning system. In order to sufficiently reduce this noise, a threshold is applied to the image to ensure only pixels with high values remain in the result. Noise is further reduced with a median filter, which replaces each pixel with the median of the set of pixels composed of the pixel in question as well as the surrounding pixels within a specified radius.

The final step of the image processing chain is to detect line edges by measuring the derivates in the X and Y directions. By weighting the X direction more heavily than the Y direction, weak horizontal edges are filtered out. This is appropriate for the vehicle's line detection, since most lines are expected to be close to parallel with the vehicle's direction of movement. To obtain the gradients, a Sobel Edge Filter is applied to the image. The result of the edge-detection process is demonstrated in Figure 5.3.

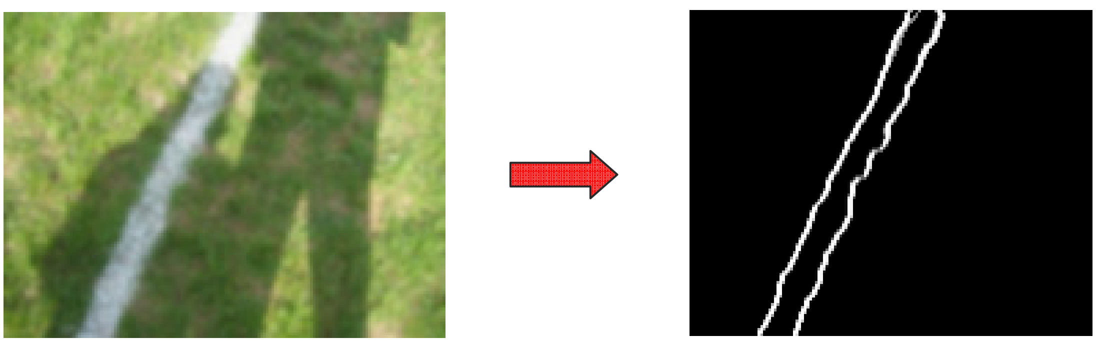

Figure 5.3 – Application of Sobel Edge Filter

The Sobel Edge Filter is applied by convolving the Sobel operator with the source image. This operator, seen in Equation 5.2, results in the image's derivative in the positive-X direction when convolved with the source image. By rotating the matrix, derivatives in the negative-X, positive-Y, and negative-Y directions can be determined. In the equation, *A(x,y)* is defined as the image intensity at position (x,y), while G_x is defined as the X-component of the gradient.

$$G_x = \begin{pmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{pmatrix} * A(x,y)$$

Equation 5.2 – Sobel Operator

In order to integrate the resultant data from image processing with the path-planning system, inverse perspective transformation must be applied to the pixels of the result image. This transformation results in the actual world coordinates of the pixel relative to the car, based on the known height and orientation of the camera. To be able to perform this transformation accurately, it was necessary to calibrate the camera for the specific orientation and lens being used. This calibration was accomplished by securing the camera into its mount on the vehicle, and then analyzing a single image taken of the grid of 1 foot x 1 foot tiles. This image can be seen below in Figure 5.4. From this image, a multi-variable linear equation was devised for calculating the real-world horizontal displacement from the pixel's X and Y coordinates.

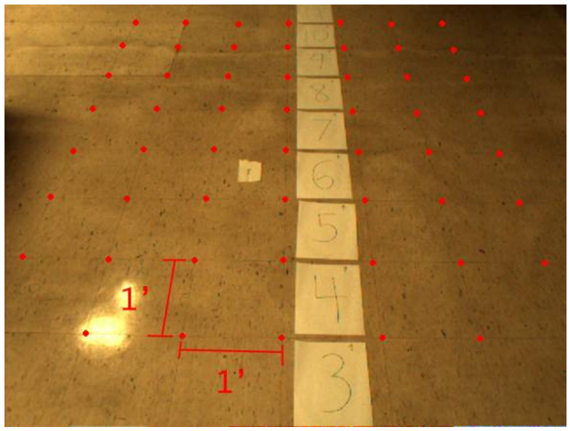

Figure 5.4 – Camera Calibration Image

Figure 5.5 demonstrates the results of the mapping system when used to detect a line, with the rightmost image representing a 2D overhead map of the area directly in front of the vehicle. The green area represents the trapezoidal field of view of the camera, while the white pixels represent the results of the image processing.

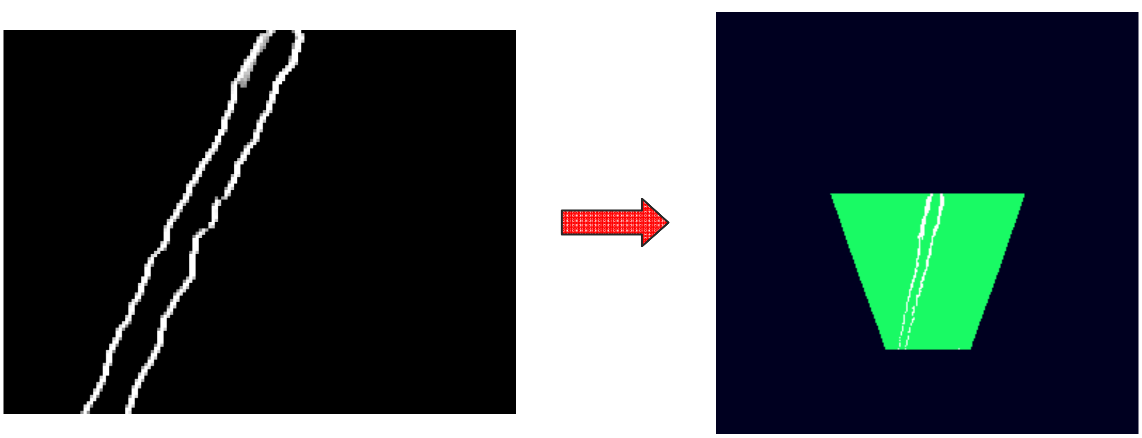

Figure 5.5 – Application of Inverse Perspective Transformation

The image-processing algorithms and software interface for the vision system was implemented entirely in C++, as opposed to the MATLAB-based implementation used by the previous team. MATLAB was considered as a possibility due to its built-in image-processing functions, however C++ was ultimately chosen for efficiency reasons. The C++ implementation of the vision system was timed for multiple successive runs, and the average processing time was determined to be 35 milliseconds. According to this average time the algorithm is 31.42 times faster than the previous algorithm, which required 1.1 seconds of processing time.

Path Planning Simulator

In order to verify the path-planning algorithm before a working hardware and software platform was completed, a software simulator was developed. The simulator, which was coded entirely in C++, utilized the path planner class to make path-planning decisions based on vision and sensor data provided by the simulator. Output is provided by a graphical rendering system, which utilizes the Microsoft Direct3D API for generating polygonal models of objects in the simulated environment. An image captured from the graphical output can be seen below in Figure 5.6. This simulator enabled the team to fine-tune the path planning system long before the mechanical platform was complete, which facilitated rapid implementation of the software when the mechanical platform was ready.

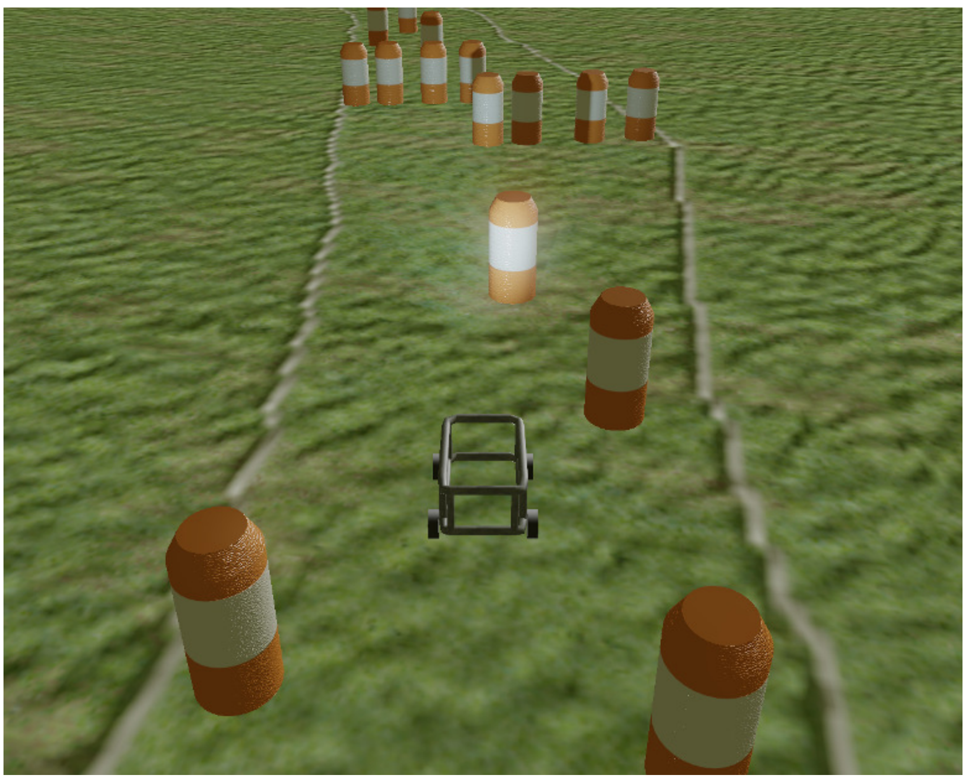

Figure 5.6 – Logic Simulation Program

6. Projected Cost

Due to the donation of the Basler camera, the cost to the team for the components used to construct the vehicle was significantly less than the expected retail cost. A tabulation of the components and costs is shown below in Table 6.1.

Part	Estimated Retail Cost	Real Cost
Digital Compass	$ 75.00	$ 75.00
NI DAQmx Digital I/O Board	$ 100.00	$ 100.00
Wireless E-stop	$ 21.00	$ 21.00
Miscellaneous Power System Supplies	$ 16.00	$ 16.00
Electronic Systems Battery	$ 20.00	$ 20.00
Scanning Laser Range Finder	$ 3,411.00	$ 3,411.00
Digital Camera	$ 1,800.00	$ -
Camera Lens	$ 152.00	$ 152.00
Firewire Hub	$ 28.00	$ 28.00
Weatherproofing Vinyl	$ 39.00	$ 39.00
Miscellaneous Mechanical Parts	$ 75.00	$ 75.00
Miscellaneous Electrical Parts	$ 220.00	$ 220.00
Total	$ 5,957.00	$ 4,157.00

Table 6.1 – Cost Breakdown

7. Conclusion

AMBER is an autonomous vehicle constructed by the NJAV team at TCNJ. The fusion of mechanical, electrical, and computer systems by the team has enabled the development of a vehicle that is ready to compete in the 2007 IGVC. The vehicle is anticipated to perform very well in obstacle avoidance, as testing has demonstrated excellent navigation of various obstacle configurations, and navigates even difficult situations such as switchbacks. The navigation challenge will likely prove more difficult for AMBER, as the mid-range GPS unit employed has limited resolution, with a best-case of 3 meters. A more expensive GPS unit would likely eliminate this problem, but the acquisition of the laser rangefinder unit this year limited further expenditures on expensive items. However, due to the effectiveness of the path planning algorithm, the vehicle is expected to perform moderately well on the navigation challenge.