UNIVERSITY OF MINNESOTA

13th Annual Intelligent Ground Vehicle Competition
2005 Design Report

Design Report prepared by:
Shawn Brovold Gregory Rupp Eric Li Nathan Carlson

Faculty Advisor:
Max Donath

Faculty Certification

I, Max Donath, Faculty advisor of the 2005 University of Minnesota IGVC team, certify "the engineering design in the vehicle (original or changes) by the current student team has been significant and equivalent to what might be awarded credit in a senior design course."

X___________________________________ Date: __________________

1. Introduction

This year marks the 4th time that the University of Minnesota has entered the IGVC competition. The last University of Minnesota robot entered in the competition was the Autonomous Above Ground Gopher (AAGG) in 2001. This year's vehicle is based on the AAGG vehicle chassis, but it has undergone a ground-up rebuild. Major changes to this year's vehicle are structural, electrical, and software-based. Since the robot has kept same basic layout of the previous vehicle, this year's vehicle has been named G2 (Gopher 2).

2. The Team

The University of Minnesota has assembled a team of students from the Department of Mechanical Engineering. Four students have worked on the 2005 IGVC vehicle throughout the course of the school year as part of an extracurricular project. See Table 1 for a description of each student's role on the project and an estimate of the number of respective hours spent.

Table 1: 2005 University of Minnesota IGVC Team

Name – Degree Program	**Primary Role**	**Hours**
Gregory Rupp – Undergrad (Sr.)	Image processing & Vision strategy	500
Nathan Carlson – Undergrad (Sr.)	Vehicle design & Motion Control	500
Shawn Brovold – Graduate	Navigation strategy & Vision strategy	500
Eric Li – Undergrad (Sr.)	Motion Control & Obstacle Avoidance strategy	500
Max Donath – Professor	Faculty Advisor – Mechanical Engineering	N/A

3. Design Process

The design process used for the robot construction and development followed the pattern expressed in Figure 1. The process began by defining the project constraints and tasks at hand, such as size requirements, necessary safety features, and computer processing objectives. With this in mind, the tasks were divided among team members, so each member had equivalent share in the overall project. The results of brainstorming ideas were documented and assembled into the initial robot concept. Based on the initial concept, ideas were implemented in both software and hardware, and trial runs were used to test if concept objectives were met. Those objectives left unsolved were corrected or added in the reiteration phase, and returned the team to the implementation phase. Upon completion of a solid frame and control strategy, the team moved beyond the testing phase. The only thing left at this point was to streamline the processes and repair any bugs that lingered throughout the other design phases.

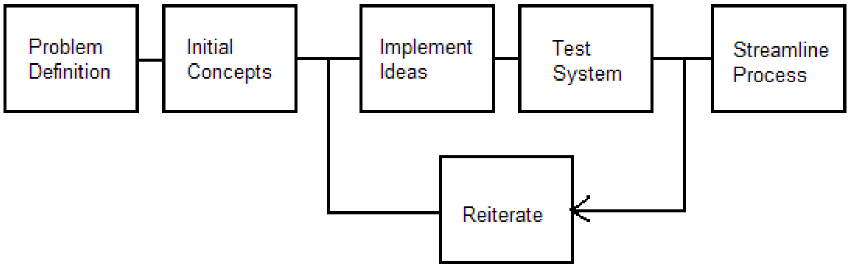

Figure 1: Design process diagram

4. Vehicle Description

The 2005 vehicle borrows many of the chassis components from the 2001 vehicle, but several key changes were made. The chassis is based on a one-inch steel tube structure from the previous robot, but modifications were made to shorten the wheelbase. By bringing in the tires, the overall width has been reduced by more than a foot, and the robot is now capable of navigating narrower passages with improved clearance. The addition of larger diameter snowblower tires improved traction and raised the ground clearance of G2. To add additional rigidity, 1/8” aluminum side panels were bolted on to the frame. The resulting chassis is shown in Figures 2 and 3.

4.1 *Vehicle Design*

Figure 2 below shows the vehicle layout. Two independently controlled servomotors are used to drive G2. Due to space considerations, the motors are mounted 90° (perpendicular) to the front wheels. The configuration was originally chosen due to its simple design, yet decent maneuverability. Since each wheel can be controlled independently, our robot is capable of stationary turns by rotating the wheels in opposite directions. There is a third wheel mounted centrally at the rear of vehicle. The third wheel rotates freely, and only provides stability.

Figure 2: Top view of G2 layout

Figure 3: Perspective view of G2 chassis

4.2 *Hardware*

The power propulsion system for the robot is based on two Kollmorgan servomotors with a HP rating of 1.1 each. To reach adequate torque ranges, the servomotors are linked to a 20:1 worm gear reduction via rubberized joints to remove any eccentricity in the shafts. Beyond the output shafts of the speed reducers, snowblower tires are mounted with ball bearings to remove any cantilever beam loading. The tailing wheel is a standard caster wheel designed for loads in excess of 800 lbs.

4.3 *Electronics*

Due to the large power draw of the PC and electronics, G2 uses a hybrid power supply of DC batteries and an AC gas powered generator. Four 12-volt automotive batteries are wired in series to produce a 48-volt power supply for the servomotors. A set of DC brushless servo amplifiers is used to control power to these servomotors, and motor feedback is provided by a set of optical encoders. Since the power draw from the computer and AC powered electronic sensors was relatively stable at around 650 Watts, a generator turned out to be practical means of providing power. The selected model was the inverter series available from Yamaha, which produced 1000 Watts and a clean 120 Volt sine wave signal.

In an attempt to organize wiring, a Plexiglas electrical box serves as the main hub for all internal electrical components. The box features quick connections for the equipment, allowing the electrical box to be removed with ease and rewired as necessary. Additional wiring was also added to ease the installation and removal of the four 12 volt batteries through the use of in-place battery clamps.

Results from field-testing indicate a battery life of 5 hours for the DC motors and a generator life of 8 hours running off a ½ gallon of unleaded fuel. Motor torque and battery power proved to adequate during test conditions, as the robot was capable of climbing a 20-degree incline and return down the slope without losing control. Maximum speed of G2 is approximately 4.5mph.

5. Software

LabView by National Instruments was used as the software package for programming of the robot. The previous University of Minnesota team had programmed in C and C++ using Microsoft's Visual Studio. An attempt to revive some of the code was made at the beginning of the project. However, after dealing with poor documentation and numerous bugs, the team decided to consider a new software package. The shift to LabView has been one of the best choices made by the team. The intuitive nature of LabView's graphical environment allowed the team to learn programming quickly, and the team was able to take advantage of LabView's motion control and vision packages.

6. Sensors and Computer

This year's IGVC team has kept the same sensors as the 2001 team, but added a magnetic digital compass to the array of sensors on G2. The overall array of sensors now includes a laser range finder, GPS unit, magnetic digital compass, and a vision system. Figure 4 shows a flow diagram of how all of G2's sensors and electrical components are linked. A cost breakdown of these sensors and other components is located in Appendix A.

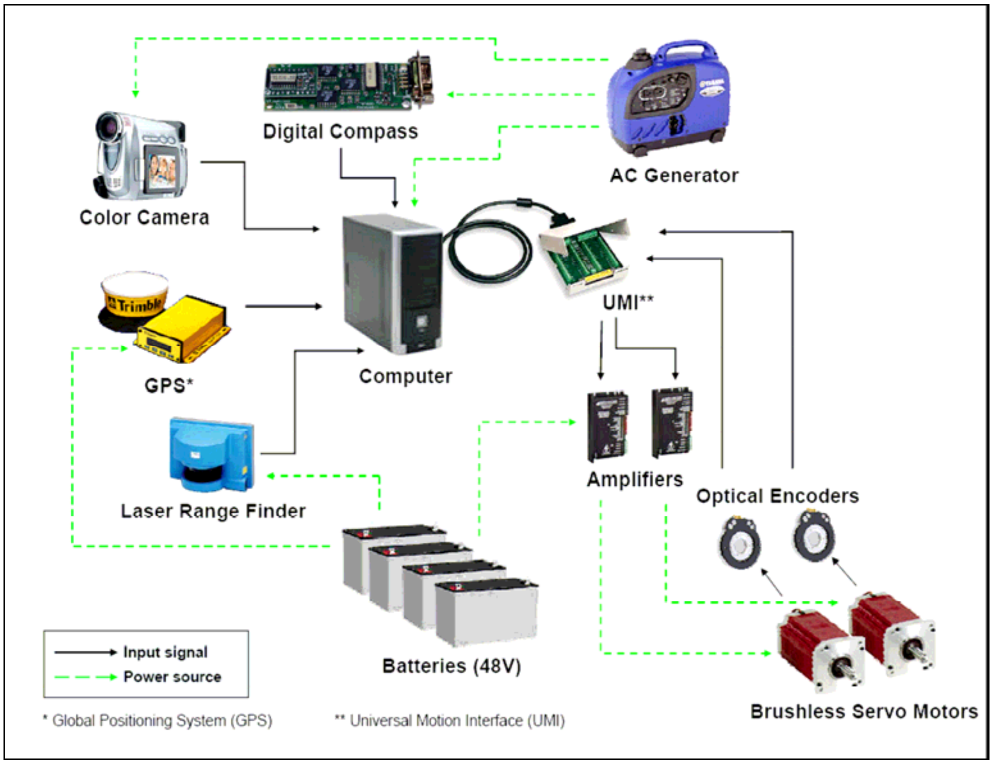

Figure 4: G2 component flow diagram

6.1 *Laser Range Finder*

The laser range finder used on G2 is the SICK Laser Measurement System (LMS), which is used for obstacles avoidance. The SICK LMS has 180 degrees scanning range, and it can detect obstacles within 8 meters. The sensor has 10 mm measurement resolution, and it has the capability to scan every half-degree with a 0.25 degrees angular resolution. For fast processing speed of this sensor, G2 is set to scan in one-degree increments.

Figure 5: SICK LMS

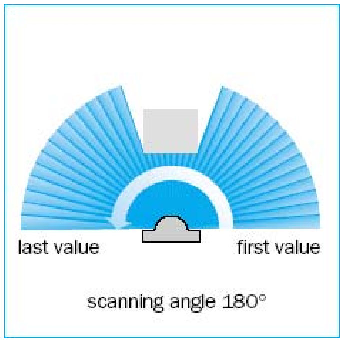

Figure 6: Measuring Range

6.2 *Differential GPS Unit*

The robot uses an AG-124 Differential GPS unit manufactured by Trimble®. The AG-124 uses RTCM beacon correction signals provided by the US Coast Guard. The $GPGGA and $GPRMC NMEA sentences provide position, heading, velocity, and GPS quality information to G2. Updates are provided at 1Hz, and specified horizontal positional accuracy is less than one meter. Although, observed accuracy during trial tests has been within a foot.

6.3 *Magnetic Compass*

For heading information, the robot uses an HMR3300 magnetic digital compass manufactured by Honeywell. The compass is a three-axis magnetometer with tilt compensation provided by two accelerometers. Observed heading accuracy is about 5 degrees due to noise in the accelerometers caused by vibrations of the robot. Calibration of the compass is used to compensate for variation in heading caused by ferrous objects on the robot. Heading updates are provided at 5Hz.

6.4 *Vision System*

G2’s vision system uses a Canon ZR200 digital camcorder equipped with a polarized lens to reduce glare. This particular camera was chosen because of its image stabilization, manual override features, video output format, and for its price. Wide-angle (720 x 480) images are captured using a National Instruments frame grabber about every 225 ms.

6.5 *Computer*

All software and computer hardware runs on a desktop PC. The computer system uses a 3.2 GHz Intel processor with 1 Gigabyte of memory. A desktop PC was chosen over a laptop computer because of the number of peripheral cards used. Mounted within the computer is a National Instruments analog image capture board for interfacing with the camcorder. A National Instruments 4-axis motion control board is used to interface with the servomotors. A mutliport serial card provides the interface for the laser sensor, GPS unit, and compass. The PC was also cost effective and allowed the greatest flexibility in terms of upgrading or replacement of parts in the event of a component failure. However, a laptop hard-drive is used for data storage because of the increased shock resistance over a standard PC hard-drive.

7. Autonomous Challenge

For the Autonomous Challenge, G2 uses information gathered using the vision system, the laser sensor, and feedback from the motors. The vision system is primarily used to find the location of the lane lines. The laser sensor is then used to locate the position of the obstacles. Then, using the information from these sensors, a path is determined to avoid the obstacles and remain within the lanes. The path is passed to the motion control strategy, and the path is updated during each computation cycle.

7.1 *Image Processing*

Once the image has been captured by the vision system, it is resized to 320 x 240 pixels. Resizing the image reduces the amount of noise and the processing time. G2's software will then extract all obstacles; including but not limited to, white five-gallon pails, orange cones, and orange barrels. Obstacles are found by applying an "intelligent" threshold to the individual color channels of the image. This process is applied to the original image twice to capture all types of obstacles. First using the red channel to extract bright orange and white objects and then using the green channel to extract dull orange objects. A unique threshold is selected based on the histogram slope and the brightness of each image. As a result of the threshold, the image is converted to binary.

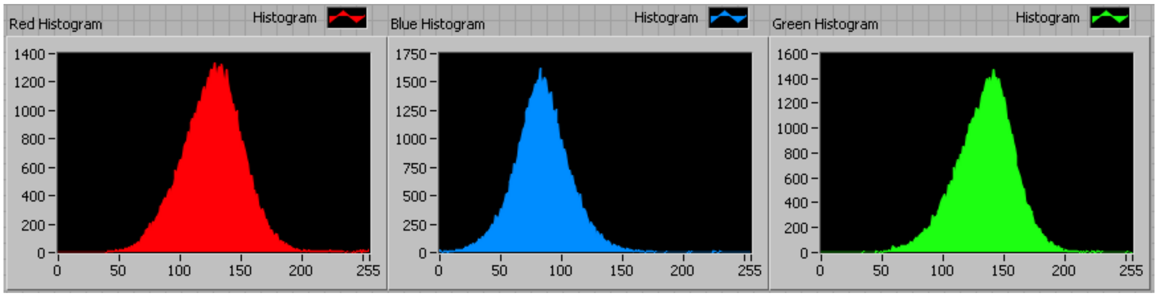

Figure 7: Histograms of all three channels (RBG) used for thresholds

From the resulting binary image, noise is then subtracted out based on its area and perimeter; leaving just the obstacles in the image as shown in Figure 9. This image can be used to verify obstacle location and to detect potholes.

Figure 8: Captured Track Image

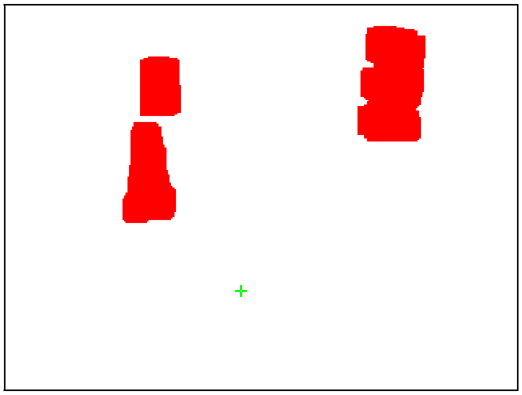

Figure 9: Obstacles found in Image

Using the same intelligent threshold process described above, the lane lines are then isolated in a new image. To deal with portions of the obstacles remaining as noise, the obstacles are subtracted out of the image so only the lane boundaries remain. Figures 10 and 11 below show the image before and after the obstacles are subtracted out.

Figure 10: White objects found in Image

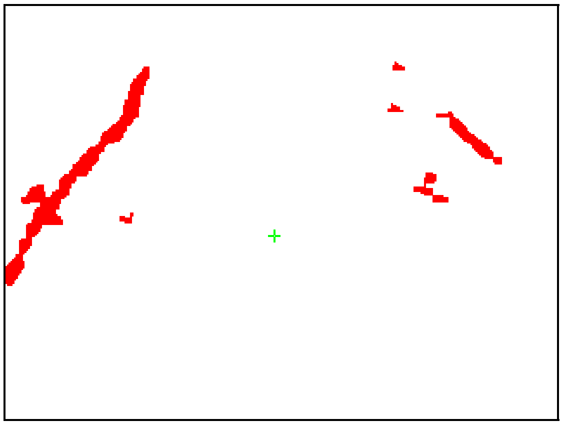

Figure 11: Obstacles removed from Image

Once the lane boundaries are isolated a Hough transform is applied to the image. This process will find straight lines within the image based on the number of pixels that are collinear, while ignoring noise. Figure 12 below shows the final image that defines the lines.

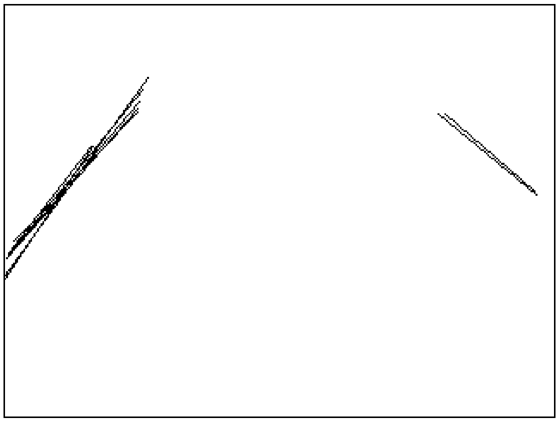

Figure 12: Hough lines drawn over track lines

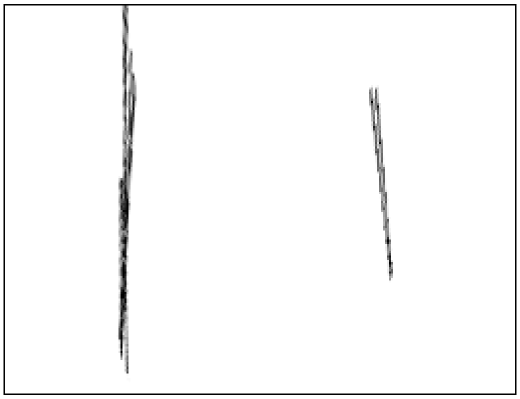

Figure 13: Calibrated Image

7.2 *Path Following Strategy*

The path following strategy is based on the output of the Hough image. With the objects removed, the remaining image only contains the Hough lines to represent the location of the actual path lines. The Hough image is first calibrated from a perspective view, to a two-dimensional, top-down view as shown above in Figure 13. From this image, we are able to determine the real-world location of the track lines and determine the optimal drive path. The resulting image has a linear relationship between pixel locations and their real-world location, making it straightforward to interpret the location of the track lines based on their pixel locations. The track line coordinates are then measured with respect to an origin centered at the robot's laser sensor. This also simplifies integration of the track line locations with obstacle locations gathered from the SICK laser sensor.

After the image is calibrated, the optimal drive path is found. The image is scanned using several horizontal search lines. See Figures 14.1 and 14.2. The search lines look for a jump in the pixel intensity. Since the image is binary, a value of zero represents no line, and a value of 1 represents a possible line. Using several search lines, multiple coordinates along the Hough lines are found, and resulting lines are fitted to represent the lanes. The optimal drive path is then calculated as series of coordinates down the center of track and assuming there are no obstacles. This path is then passed to the software performing the obstacle avoidance strategy where it is modified once the locations of the obstacles are determined.

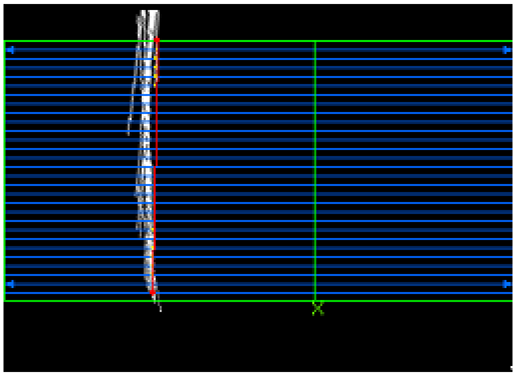

Figure 14.1: Path offset from single line

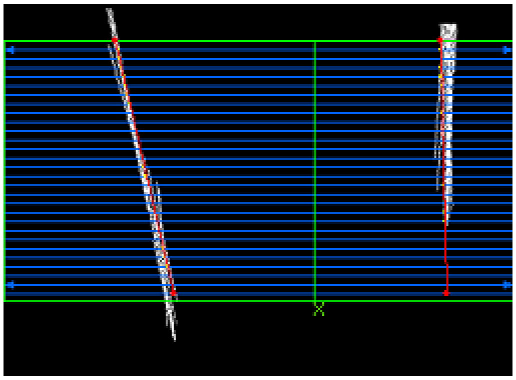

Figure 14.2: Path found from both lines

This strategy also keeps track of which lines are left or right of the robot. Each line is assigned it’s own search region, and the search region moves relative to the position of the line in the image so that left and right lines can be tracked. If two lines show up in the image, then the optimal drive path is calculated as centered between the two lines. If only one line is found, the drive path is offset to the correct side of the line by five feet. Five feet represents half of the assumed track width. If no lines are present, the robot uses the location of the lines gathered previously to continue on a path. When the lines are present, the robot also verifies the location of the lines in the current image with the previous images. This reduces the chance of a bad image adversely affecting the path of the robot, and also allows G2 to deal with dashed lines.

The scan line method was chosen so that stray lines could be dealt with in the lane finding strategy. Stray lines may be caused by noise in the image, so it is important for the path following strategy to handle them. By averaging several pixels in the search line, single lines caused by noise can be ignored by selecting a threshold on the resulting average pixel value. Also, by using multiple search lines and determining a best fit of the calibrated Hough image, bad coordinates returned by stray lines can be averaged out.

8. Navigation Challenge

For the Navigation Challenge, the robot uses information gathered from the robot’s GPS sensor, a magnetic digital compass, the laser sensor, and feedback from the motors. The GPS system is used to determine the distance and location of the waypoints in a Cartesian coordinate system relative to the robot. The compass is used to determine the current heading of the robot. The laser sensor is then used to locate the position of the obstacles. Using the information from these sensors, a path is determined to avoid the obstacles while driving toward the waypoints. The path is then passed to the motion control strategy, where the path updated during each computation cycle.

8.1 *Path Planning*

The first step in the path planning strategy determines the drive order of the waypoints. Since only the coordinates of the waypoints are given, and not the obstacle coordinates, the order is determined assuming no obstacles are present. A program was written to calculate the shortest path between all waypoints from a starting position. The program returns the distance of the shortest path, along with the corresponding order of successive waypoints. If a better path is determined based on a visual inspection of the field, the order can be modified prior to successive runs.

The GPS sensor and compass are used to calculate the optimal drive path from the robot's current position to the next waypoint. The waypoints and the current GPS position are both converted from the Earth centered Earth fixed (ECEF) latitude/longitude position to a locally projected Cartesian coordinate system (Michigan State Central at this year's competition). This simplifies the calculations used to determine the location of the robot relative to the next waypoint. The compass is used to determine the current heading of the robot. This heading value is also verified using GPS. If the heading value from the magnetic compass seems erroneous, perhaps due to interference from nearby ferrous objects, the heading from GPS is used. From the heading and position, the location of the waypoints relative to the body axes of the robot is calculated. See Figure 15 below. Therefore, the robot's current position is always (0,0), and the relative position of the next waypoint is re-calculated during each position update from the GPS unit.

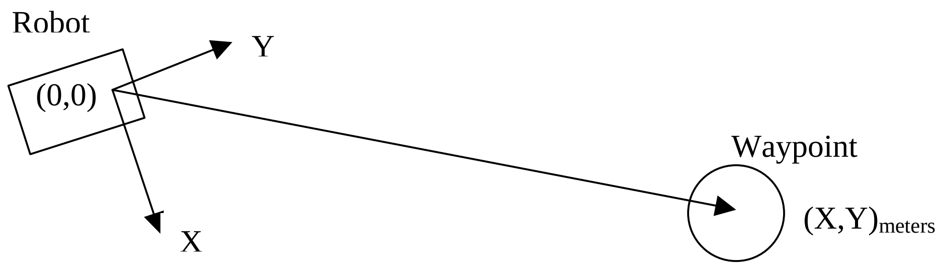

Figure 15: Waypoint relative to robot axes

At this point, the path planning strategy is similar to the strategy used in the Autonomous Challenge. However, for the Navigation Challenge, an optimal path is found using the GPS and compass. The optimal path is a straight line from the robot's origin to the waypoints. This path is then passed to the obstacle avoidance strategy, where the path is modified to avoid obstacles.

Since the horizontal (positional) accuracy of the GPS unit is less than one meter, it is assumed that the GPS position is also the absolute position of the robot. Therefore, the GPS position is sufficient for the robot to cross the diameter of the waypoint circle. Once the robot reaches a position within one meter of the waypoint's position, the optimal drive path is directed toward the next waypoint in the order. Once the robot reaches the final waypoint, the software commands the robot to stop.

9. Obstacle Avoidance

The strategies for obstacle avoidance in the Autonomous and Navigation Challenges are similar. G2 first obtains the updated desired path. Then, based on the data from the SICK LMS, the software calculates the available openings ahead. An optimal opening is selected by G2, and the desired path is then modified to pass through the opening. This modified path is sent to motion control to perform path following.

9.1 *Defining Openings*

Data from the LMS is used to locate potential openings that are clear of all obstacles. In Figure 16, it appears there are obstacles located in the entire search area in front of the LMS. However, the distances between the obstacles and the LMS are not the same. Therefore, openings are assumed if there is a large change in distance in between angle measurements, see red arrow in Figure 16. This method has also been found to eliminate traps as potential openings. Once openings are found, it is necessary to check for adequate clearance. This is done by calculating the distance between obstacles in the assigned opening (blue area) to be sure that there is enough clearance for the robot. Openings passing this test are candidates for G2 to drive through.

9.2 *Path Modification*

Since there is often more than one candidate opening, G2 picks the best opening based on the desired path. If the desired path lies in an opening with substantial clearance, the path does not need to be modified. Otherwise G2 adds one more destination point to the path so it will pass through the opening, as shown in Figure 17. The modified path will then be sent to motion control for path following.

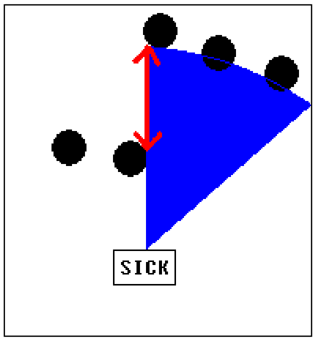

Figure 16: Opening example

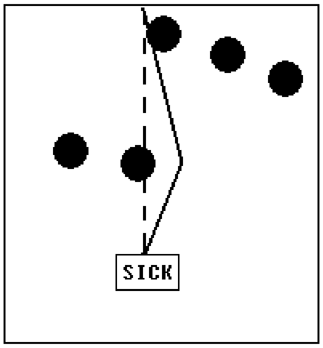

Figure 17: Modify path

Since the processing time to read data from SICK LMS and to finish modifying the path is about 0.3 seconds, it is only necessary to pick one point each cycle. With the fast processing time, and a measuring distance of the SICK LMS of about 8 meters, G2 is able to modify its path and avoid obstacles even at high speed. Observed reaction time is one computation cycle (0.3 seconds).

10. Motion Control

Once a path is retuned from the obstacle avoidance strategy, G2 uses dead reckoning to follow the reference path. Dead reckoning is based on the servomotor encoder feedback, and the robot's path is updated during every calculation cycle.

10.1 *Reference Path*

The reference path for motion control of G2 is the modified path calculated by the obstacle avoidance strategy. Since the path is formed by an array of points, G2 lines up the previous point and the next point, and follows the path by minimizing its perpendicular distance from that line segment. See Figure 18. The heading of the reference path is also determined using the coordinates of the line points. Therefore, G2 also attempts to match it's heading with the line's heading.

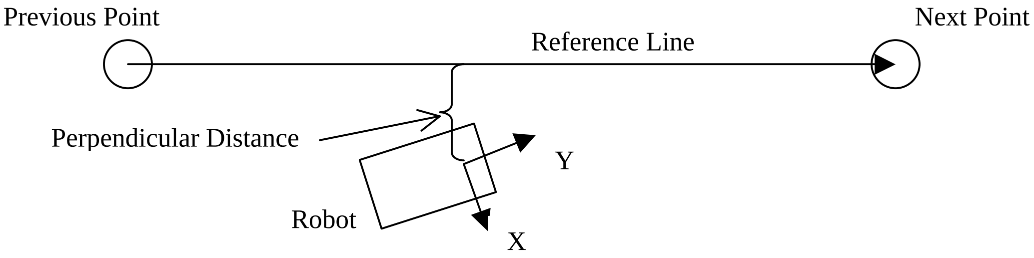

Figure 18: G2 relative to reference line

10.2 *Turning*

When a new reference path is sent to G2, or if the robot strays from the current path, it must execute a turn. The turning radius and length of the turn arc is based on the distance between G2 and the reference line. When G2 is very close to the reference line, it maintains a heading that is parallel to the line. When G2 is far from the line, the turn radius and arc length are calculated so the robot can return to the line.

11. Conclusion

Since the last University of Minnesota team to enter the IGVC competition was in 2001, there were no returning members for this year's team. Therefore, this year's team started with little prior knowledge of the vehicle and previous strategies. Since the 2005 team was able to use several of the AAGG vehicle's components, the team concentrated on developing new software from scratch and correcting the issues that afflicted the previous team. As a result, the 2005 vehicle, G2, is more compact, has more reliable software, and will hopefully have improved success in this year's competition.

12. Acknowledgments

Special thanks needs to be given to the Intelligent Transportation Systems (ITS) Institute at the University of Minnesota for financially supporting this year's team. In addition to our faculty advisor, we would also like to thank our other advisor, Dr. Don Krantz from MTS Systems Corporation, for his support on this project.

Appendix A: Spec Sheet and Cost Breakdown

Description	Qty.	MSRP	Team Cost
Sensors			
SICK Laser Range Finder	1	$5,700.00	$0.00
Canon ZR200 Camera with polarized filter	1	$450.00	$450.00
Trimble AG-124 DGPS receiver	1	$2,995.00	$0.00
Honeywell HMR3300 Digital Compass	1	$495.00	$495.00
Computer			
3.2GHz Desktop PC	1	$800.00	$800.00
15" Flat screen LCD Monitor	1	$500.00	$0.00
NI 4-axis Motion Control Board	1	$1,650.00	$1,650.00
NI Image Capture Board	1	$865.00	$865.00
Comtrol Multiport Serial Card	1	$599.00	$0.00
Hardware			
Kollmorgen DC Brushless Servo Motors	2	$1,200.00	$0.00
Boston Gears 20:1 worm gear speed reducer	2	$800.00	$0.00
Frame Expenses	1	$700.00	$700.00
Power			
Yamaha 1000 watt inverter generator	1	$669.00	$669.00
12 volt 40 A-H TOYO batteries	12	$650.00	$650.00
Brushless DC Amplifiers	2	$950.00	$0.00
Software			
LabView 7.1 - Full Development Version	1	$2,395.00	$0.00
LabView Vision Development Module	1	$2,595.00	$0.00
LabView Motion Control Extension	1	$0.00*	$0.00
*included with purchase of control board			
	Total	$24,013.00	$6,279.00