Wunderbot
An undergraduate autonmous robot project
PHŒNIX CONTACT
SICK
Developing new solutions...
and educating the public

**Faculty Statement**
I certify the "Wunderbot 4" has made substantial improvements in the areas of theory, hardware acquisitions, and software control from the previous "Wunderbot" entries into the IGVC competition.

Joseph T. Wunderlich, Ph.D
Project Advisor
Physics & Engineering Department
Elizabethtown College

Wunderbot Project
One Alpha Drive
**Elizabethtown College**
Elizabethtown, Pennsylvania
17022
ph: 717-361-1000
fx: 717-361-1400
www.etown.edu

**Table of Contents**

**Team Members:**
James Painter, CE&CS
David Coleman, CE
Jeremy Crouse, CE
Chris Yorgey, EE
Mike Patrick, CE
Dan Fenton, CE

**Project Advisors**
Joseph Wunderlich, Ph.D.
Thomas Leap, Ph.D.
Troy McBride, Ph.D.

## 1. Introduction

The Robotic and Machine Intelligence (RMI) Laboratory at Elizabethtown College is proud to announce its third entry – the Wunderbot 4 - into the 2008 Intelligent Ground Vehicle Competition. Coming off a mid-field showing in the 2006 competition, the Wunderbot 4 team has made significant improvements in the area of obstacle avoidance, GPS navigation, and white line detections. While the main chassis may bring back memories of the Wunderbot 3 (2006 competitor), the new acquisitions of vision, LIDAR, and digital compass – along with complete redesign of internal software including implementation of JAUS – are steps in the right direction for the versatile platform that will serve the Elizabethtown, PA community in web-based autonomous tours in the near future.

A listing of major developments since the 2006 IGVC onboard the Wunderbot platform is listed in Table 1. A thorough documentation of the overall design process, hardware implementation, software development, as well as IGVC challenge solutions will be discussed within this paper.

| New for 2008 | |
|---|---|
| Chassis | 4’ vertical tower for camera<br>GPS/Compass integrate mount on tower<br>Waterproof attempt |
| Hardware | Front-mount LIDAR<br>DVT camera<br>JAUS access point<br>Remote control (for manual drive) |
| Software | PID control<br>Sub-system partitioning<br>JAUS implementation<br>White-line detection<br>Obstacle avoidance<br>$O^3$ GPS navigation scheme |

**Table 1 – Wunderbot platform developments since 2006 IGVC.**

## 2. Innovations

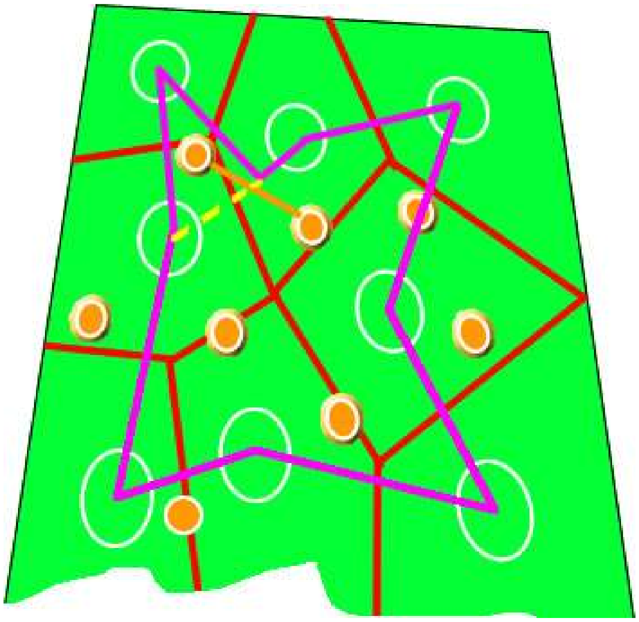


**Figure 1 – Sample GPS waypoint challenge course with coordinates outlined in white. The chosen path is shown in pink with the $O^3$ ability of "path deviation" shown in a yellow dotted line.**

The Wunderbot 4 features a new navigation scheme known as $O^3$ for GPS navigation and obstacle avoidance within GPS navigation which alters the optimal sort explicit definition of GPS coordinates as provided by the IGVC judges. The $O^3$ method was published in March in the IEEE proceedings of the Advanced Motion Control Workshop (available in Appendix B). By using the three stages of $O^3$ (**o**ptimal explicit path planning, local points of **o**pportunity, and **o**bstacle avoidance) a path between two GPS coordinates is not chosen based of local availability, but rather in a global context and the relationship between the GPS coordinate and obstacle density in the vicinity. A sample course is shown in Figure 1.

## 2. Design Concepts

The primary focus of the Wunderbot 4 team was to improve on the systems currently onboard the Wunderbot 3 platform while developing new algorithms and techniques to solve the challenges faced at the IGVC 2008. A constant process of evaluating the current system, proposing new solutions, developing these solutions (combining theory and simulation results), implementing the solution with technology donated by corporate sponsors, and testing the solution to provide reassurance of quality, controllability, and proof of concept is shown in Figure 2.

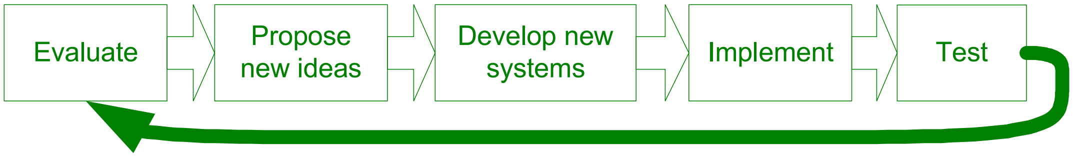
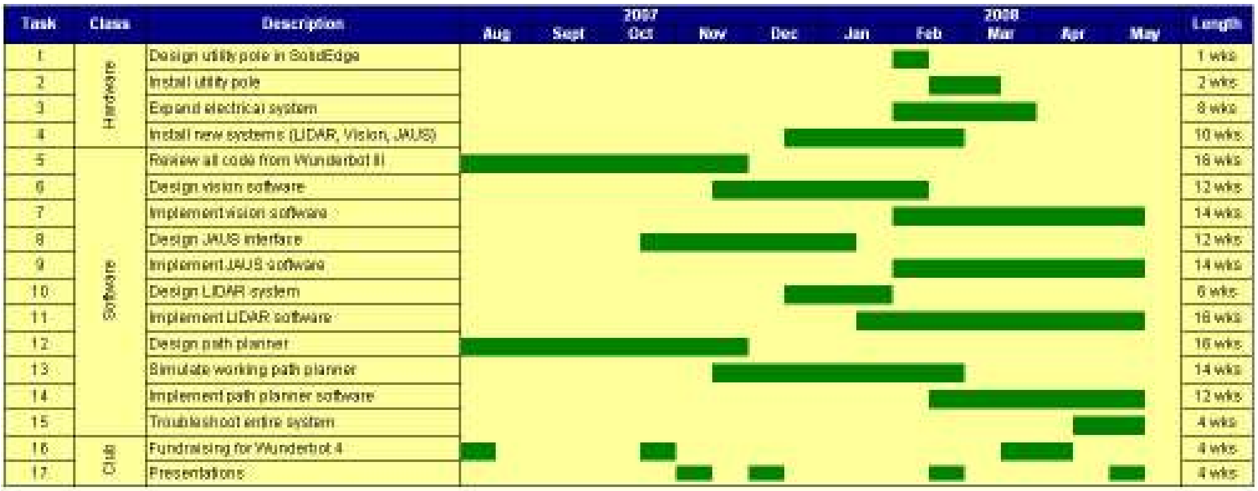


**Figure 2 – Design process governing the transition from Wunderbot 3 to Wunderbot 4.**

## 2.1. Evaluation – The IGVC 2006

The success of the Wunderbot 3 team in 2006 proved the systems developed to-date were at or above par to the rest of the competitors. However, there was significant room for improvement specifically in three areas: 1) obstacle detection - which was being handled by minimal processing on-board the camera – 2) GPS navigation – no optimal path between multiple nodes was present and 3) JAUS communication – no attempt was made in 2006. The complete evaluation period lasted about six months and the design process was about one year to develop and write the necessary items. The team's Gantt chart along with estimated 2000 man-hours is shown in Table 2.



| Task | Class | Description | | | 2007 | | | | | 2008 | | | Length |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Aug | Sept | Oct | Nov | Dec | Jan | Feb | Mar | Apr | May | |
| 1 | Hardware | Design utility pole in SolidEdge | | | | | | | 🟩 | | | | 1 wks |
| 2 | | Install utility pole | | | | | | | 🟩 | 🟩 | | | 2 wks |
| 3 | | Expand electrical system | | | | | | | 🟩 | 🟩 | | | 8 wks |
| 4 | | Install new systems (LIDAR, Vision, JAUS) | | | | | 🟩 | 🟩 | 🟩 | | | | 10 wks |
| 5 | Software | Review all code from Wunderbot III | 🟩 | 🟩 | 🟩 | 🟩 | | | | | | | 16 wks |
| 6 | | Design vision software | | | | 🟩 | 🟩 | 🟩 | 🟩 | | | | 12 wks |
| 7 | | Implement vision software | | | | | | | 🟩 | 🟩 | 🟩 | 🟩 | 14 wks |
| 8 | | Design JAUS interface | | | 🟩 | 🟩 | 🟩 | 🟩 | | | | | 12 wks |
| 9 | | Implement JAUS software | | | | | | | 🟩 | 🟩 | 🟩 | 🟩 | 14 wks |
| 10 | | Design LIDAR system | | | | | 🟩 | 🟩 | | | | | 6 wks |
| 11 | | Implement LIDAR software | | | | | | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 16 wks |
| 12 | | Design path planner | 🟩 | 🟩 | 🟩 | 🟩 | | | | | | | 16 wks |
| 13 | | Simulate working path planner | | | | 🟩 | 🟩 | 🟩 | 🟩 | | | | 14 wks |
| 14 | | Implement path planner software | | | | | | | 🟩 | 🟩 | 🟩 | 🟩 | 12 wks |
| 15 | | Troubleshoot entire system | | | | | | | | | 🟩 | 🟩 | 4 wks |
| 16 | Club | Fundraising for Wunderbot 4 | 🟩 | | 🟩 | | | | | 🟩 | 🟩 | | 4 wks |
| 17 | | Presentations | | | | 🟩 | 🟩 | | 🟩 | | | 🟩 | 4 wks |

**Table 2 – Wunderbot 4 team Gantt chart showing concurrent works throughout the 30 weeks over two semesters of 2007-2008 academic periods.**

## 2.2. Constraints on Wunderbot platform

By working with the Wunderbot 3 platform a few variables in robotic design were already guaranteed from previous successes including maneuverability, mobility, versatility, and safety as outlined in [1]. Additionally, the team balanced resources, aesthetics, and functionality to develop the new systems as outlined in Table 3. Finally, the team reviewed the constraints of the IGVC as well meeting additional requirements of Elizabethtown College and travel abilities.

| Description | Actual | Constraint | Placed by |
|---|---|---|---|
| Vertical Tower for Vision | 5’ | Under 6’ | IGVC |
| Horizontal Rear Length for Vision | 5’ | 3’-9’ | IGVC |
| Front bumper | 28” | 36” | Door width in RMI Lab |
| Wireless E-stop | Yes | Yes | IGVC/RMI Lab |
| Waterproof | 90% | None | Need |

**Table 3 – Constraints placed upon Wunderbot 4 design**

## 2.3. Chassis Improvement

Two needs were addressed with the addition of a 4’ tall tower on the rear of the robot: 1) more stable GPS receiver mount and 2) more focused viewing region for the camera. The tower was designed in SolidEdge with the specifications shown in Figure 3. The tower is constructed from one-inch square galvanized steel piping at a height of 4’ (122 cm). It was spray painted black to match the look of the robot and to prevent rust. The tower has one shelving unit approximately halfway up the tower, which is protected by plexiglass and houses the GPS receiver and digital compass. At the top of the tower at 40.5cm behind the rear bumper the camera is mounted at an adjustable angle. The angle of camera is adjusted in a coarse manner by its angle mount and finely adjusted by its setscrews in the housing unit. More information on the camera mount can be seen in section 4 and Appendix C.

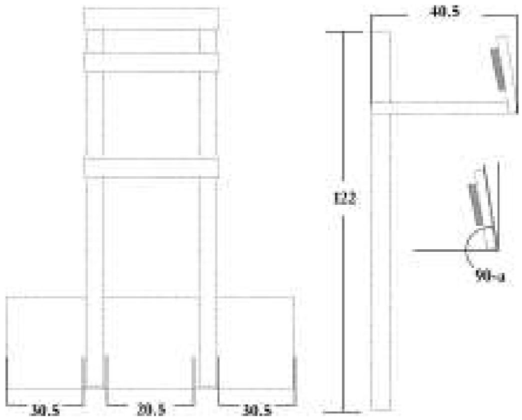


**Figure 3 – Tower design measurements. The camera can be adjusted coarsely by its mounting unit and finely by its setscrews. This angle is marked by ‘a’.**

## 3. Electrical System

The majority of the electrical system was carried over from the previous Wunderbot 3 platform with minor changes including 1) wider-conduit to hide additional communication cables, 2) increased distribution power blocks, and 3) increased number of fuses. A complete breakdown of the electrical system can be seen in Figure 4.

### 3.1. Power

Two 12V 60-amp hour batteries connected in series provides approximately two hours of operating time. A 480W 24V DC-DC ATX power supply provides voltage regulation for the onboard PC and all system components.

### 3.2. Control

Due to the castor resistance affecting the turning radius of the Wunderbot, this year's team made the decision to reverse the orientation of the drive system. It now features a rear-wheel drive controlled by two independently controlled RobotEQ AX2550 motor controllers. These controllers communicate via RS-232 and parameters are passed via LabVIEW. The RobotEQ motor controllers also feature safety parameters such as

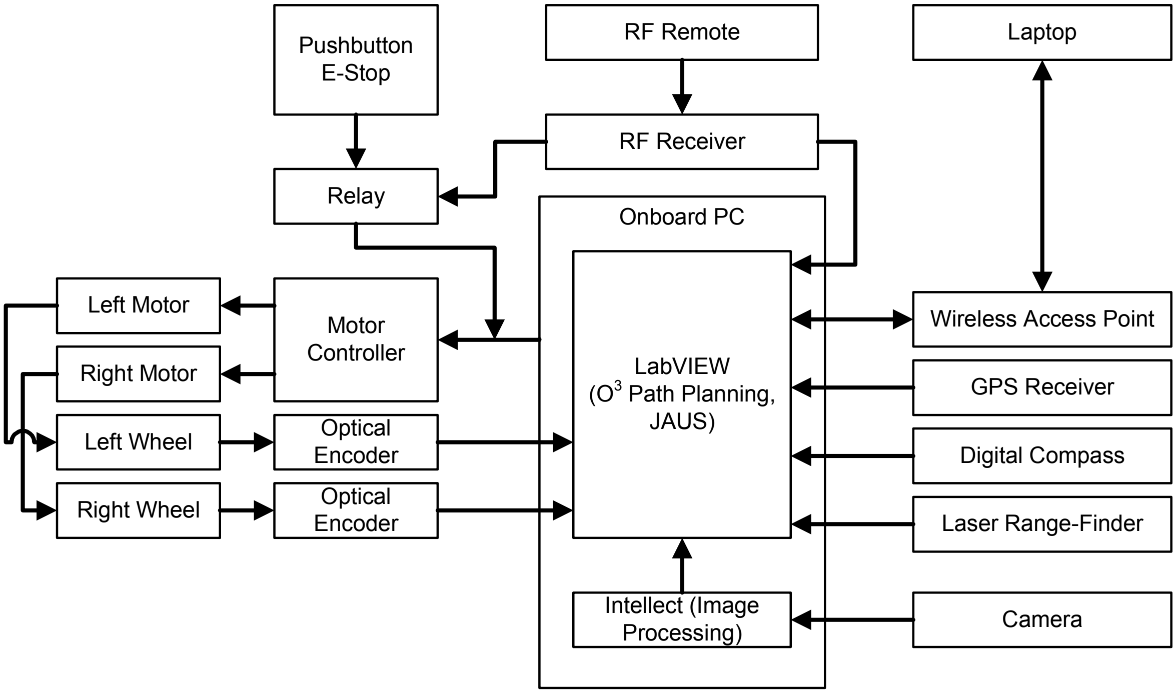


**Figure 4 – Block diagram of Wunderbot 4.**

temperature, battery voltage, and current draw to ensure proper performance and damage to equipment is prevented.

One of the most costly problems encountered during early testing was inaccurate vehicle motion. When operated on smooth indoor surfaces, Wunderbot was able to move roughly in the intended direction, but once the vehicle was tested outdoors on grass, motion response had a large degree of error. The largest cause for error is the front casters, which require a disproportional amount of force in order to change direction. This problem is a typical case for a PID controller to amend.

The PID closed-loop control was developed in LabVIEW and is very straightforward. The P, I, and D are all user-adjustable via the front panel, and feedback comes from the U.S. Digital optical encoders. Unfortunately, the robot itself is extremely difficult, if not impossible, to model via differential equations, hence classic methods of control theory could not be instituted to determine the value of the PID's constants. Instead, it was a trial-and-error procedure, which led to P=0.500, I=20.00, and D=0.001. Very subtle variations in the derivative constant led the robot to accelerate out of control. A PID controller's derivative constant in general is highly susceptible to noise, and therefore an adjustable low-pass filter was designed for the D. This kept the D from fluctuating too rapidly, while still allowing it to quicken the output's rise time.

### 3.3. Emergency Stop

The Wunderbot 4 now features four ways of stopping the robot: two hardware and two software. The onboard hardwired e-stop normally open button will instantly ground the motor controllers and motion will only be activated with a program reset. This same relay is available wirelessly through a remote switch.

With the addition of a remote control for manual drive an additional emergency stop was introduced and is controlled initiated through software. By activating this e-stop button the control program (LabVIEW) will immediately abort the program and the communication between the PC and motor controllers will be eliminated; thus stopping the robot. Finally, anytime the remote LabVIEW program is stopped the robot will halt its motion and therefore, would be considered another e-stop method.

## 4. Software Strategy

The largest change made from competing in 2006 was the software strategy onboard the Wunderbot 4. The team used the previous code as a guide and first developed a base flow diagram of how the decision should be made within LabVIEW. This flow diagram is shown in (Figure 1 of) Appendix A and is elaborated upon within this section.

## 4.1. Vision System

The location at which the camera is mounted has enormous impact on the image-processing step of the vision system. Various configurations were tested, comparing the field of view and corresponding image processing times. With the camera 1.2m directly above the rear bumper and 40.5cm back, the viewing distance extends to roughly 2.25m, missing some data from directly in font of the front bumper, as seen in Figure 5.

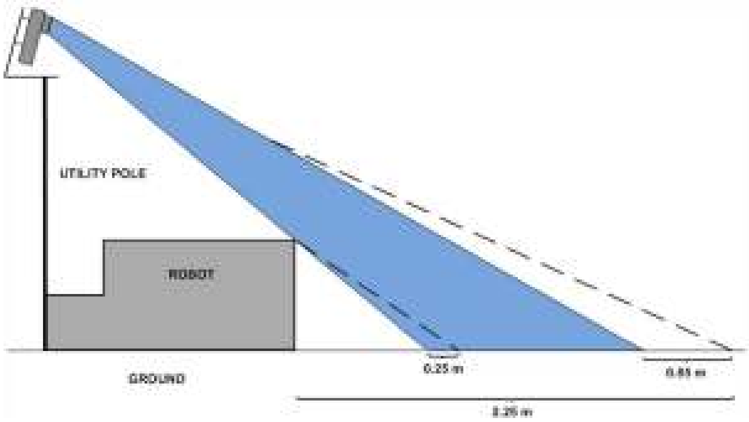


**Figure 5 – Camera angle versus distance of sight. The blue region represents that of a fixed angle at a position equal to the rear bumper. The dotted lines represent the distance gained by moving the camera back 40.5 cm.**

### 4.1.1. Signal Processing

The sacrifice of image data from in front of the bumper is acceptable, due to the trade off between seeing farther ahead and trimming the top edge of the image to reduce processing time. A feasibility study on the processing time reduction resulting from cropping the top edge of the image showed that the decrease in processing time was significant enough to allow for image cropping. The percentage speedup was a nearly linear relationship to the percentage of the image that was trimmed, and the final implementation incorporated the cropping of the top 200 lines, for a reduction of about 110ms in processing time, to about 90ms.

#### 4.1.2. Line Detection

The line detection is performed from within the camera's proprietary software, DVT Intellect v2.2. First, an erosion filter is applied to the image, using a 3x3 kernel. This closes many holes of noise, such as small dirt patches that appear through the grass, while still maintaining the shape of the desired white lines. Larger kernels could produce an even more accurate image, but processing times increase sharply as the kernel grows larger. Once noise has been filtered, an Intellect ``line thickness'' sensor is applied. This measurement sensor first uses a 60% intensity threshold to deduce a binary image. The sensor then scans every row in the image to find the two edges closest either side. Final line pass/fail conditions are used to filter shadows and other undesirable objects in the field of view. A maximum width condition of 300 pixels is combined with a ``straightness'' condition.

#### 4.1.3. Path Planning

The data from the camera software is then sent to LabVIEW to be used for path planning. In general, when two lines are found, the following equation is used:

$$\left( \frac{x_{right} + x_{left}}{2}, y \right) \tag{1}$$

When only one line is found, the target becomes the point directly centered between that line and either the left or right edge of the viewable region. If the line is on the left, the target is placed on the right, and vice versa. If no lines are found, the target is placed in the center on the horizon, such that the robot will move directly forward at full-speed.

## 4.2 LIDAR

The LIDAR is a SICK LMS200 laser range finder mounted approximately six to eight inches from the ground on the front bumper of the vehicle. This equipment is used for obstacle detection in a planar view and can deliver 180-degree resolution up to 80 meters away. For the Wunderbot 4 the LIDAR is configured to deliver 360 data points (½ degree resolution) at approximately ten meters.

### 4.2.1. Signal Processing

The current obstacle avoidance on-board the Wunderbot 4 has two parts. The first is a simple local obstacle detection scheme with a dynamic viewable window. This viewable window goes through a two-fold level of calculations to determine the best path for the robot to follow. A radial filter is first placed on the incoming data of the LIDAR. The data transmission of the LIDAR is in polar coordinates so a radial filter is best. The radial distance is determined by the following equation:

$$\text{Filter}_{\text{radius}} = (\text{Window}_{\text{height}}{}^{2} \text{ x } \text{Window}_{\text{depth}}{}^{2})^{\frac{1}{2}} \tag{2}$$

After the filter is applied the obstacles found less than *r* are converted to (x,y) using the following:

$$x = r * \cos(\Theta) \tag{3}$$

$$y = r * \sin(\Theta) \tag{4}$$

### 4.2.2. Obstacle Avoidance

Finally, an obstacle is "within the window" iff:

$$x < \tfrac{1}{2} * (\text{Window}_{\text{depth}}) \tag{5}$$

$$\text{and} \quad y < \tfrac{1}{2} * (\text{Window}_{\text{height}}) \tag{6}$$

The result of the first part is if an obstacle is found within the window a decision is necessary. A polar histogram is developed from the "window" obstacles and is shown in Figure 6. This histogram [2] is useful in determined which direction (left, center, right) has the highest obstacle density and should provide the highest cost function, locally.

### 4.2.3. Path Planning

Until now the methods discussed are local methods for obstacle avoidance. Each part does not contain starting and goal positions relative to the obstacle being detected. A* provides the ability to incorporate the obstacle detected with the end goal to develop a heuristic approach to obstacle avoidance and overall guarantee an optimal prune of the search arena.

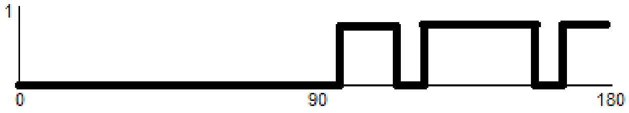


**Figure 6 – A binary polar histogram is useful locally in determining which direction to turn based on the obstacle density within a region: left (0-45 degrees), center (45-135) and right (135-180).**

The A* method was used in simulations and is implemented on-board the Wunderbot 4. The simulation versus implementation, however, is different as the simulation provided a few assumptions, which cannot be assumed on the real application. These assumptions are listed in Table 4.

As stated the benefit of using A* is the knowledge of the starting and destination coordinates in the cost function. The cost function of A* is three parts:

| Simulation assumptions | Actual constraints |
|---|---|
| Uniform motion in any direction | Closed loop control necessary to maintain speed/direction after command issued. |
| Zero-degree turning radius | Turning radius approximately < ½ meter. |
| 360-degree sensing | 180-degree line of sight |
| No time delay in data acquisition | Equipment transmission delays |
| Coordinates of absolute location | |

**Table 4 – Assumptions made during simulations and actual constraints on project.**

g = distance from start node

h = distance to goal node

f = g + h (7)

The distance calculations are done using the Manhattan method, which states that only square paths are to be taken in the X and Y direction independently. Diagonal paths are acceptable at higher costs.

The simulation results can be found in Appendix F with the actual implementation windows shown in Figure 7. Notice the only available windows for the cost function in the implementation are the three windows in front of the autonomous vehicle and that the global solution is maintained using a mix of local detection and global heuristics.

### 4.3 GPS/Digital Compass

The orientation in space is obtained through two sensors: 1) GPS receiver and 2) digital compass. The combination of these two sensors allows for specific path planning and destination within one meter. The GPS unit is a Trimble AgGPS 114 receiver with DGPS service provided by OmniStar. The digital compass provided by PNI features 3-axis roll, pitch, and yaw measurements.

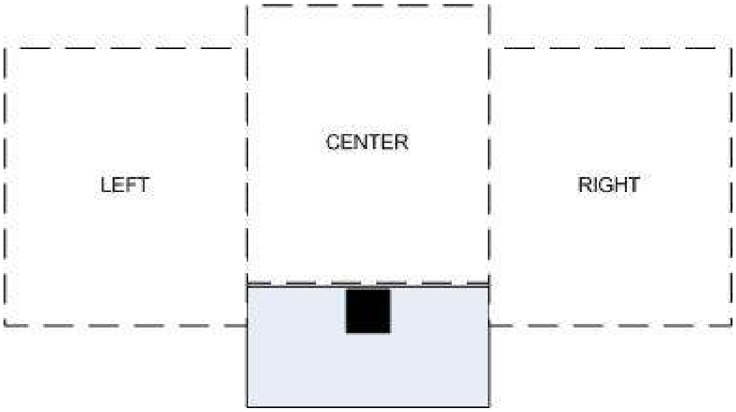


**Figure 7 – The actual window and line-of-sight for the Wunderbot is only 3 of the 9 squares available compared to the simulation. This is due to the limited (180 degrees) vision of the robot.**

#### 4.3.1. Signal Processing

The GPS information is transmitted via RS-232 at a sample rate of one hertz from the GPS receiver to the PC in NMEA sentence of the following sample format:

```
$PTNL,GGK,172814.00,071296,32723.46587704,N,12202.26957864,W,3,06,1.7,EHT-6.777,M*48
```

The first nine of eleven fields are used by the Wunderbot 4, which include UTC time, UTC date, longitude, N/S orientation, latitude, E/.W orientation, GPS quality, number of satellites, DOP of fix.

The digital compass is also transmitted via RS-232 and is read in at a variable rate. In the control software the port is read when 60 bytes of information are available from the device. The transmit time therefore ranges from 10-20ms.

### 4.3.2. Waypoint Challenge

Using the O$^3$ method path planning is done in two steps: explicit (before motion) and implicit (during motion). By sorting the GPS points through traditional discrete algorithms an optimal order can be achieved in an ideal environment without obstacles. Furthermore, when the introduction of obstacle occurs – in real time discovery – the Wunderbot 4 is capable of implicitly changing its path to adapt to its environment.

*Explicit path planning.* Upon receiving the GPS coordinates from the IGVC judges a custom script (written in Matlab) is used to sort the points in an optimal order based on the distance matrix. Since the number of possible paths is on the order of

$$!(n-1) \qquad (8)$$

a method was developed using the Delaunay triangulation as outlined in Appendix B. This method has shown to bring the number of permutations from the complete set shown in equation (8) down to a reasonable set of size *n*. Additionally, it has also been proven that by using the Delaunay sub-graph the globally optimal solution has been preserved and the integrity of the solution has not been compromised.

*Implicit path planning.* As outlined in Appendix B, a non-traditional use of the Voronoi polygons has allows for more efficient traversal of the arena. An obstacle that expands neighbor polygons can allow for local points of opportunity that through testing have shown to be globally optimal.

### 4.3.3. Path Planning

The global path planning is done through the explicit and implicit defined above. The path planner in this challenge incorporates the data from three systems: 1) GPS, 2) digital compass, and 3) LIDAR. The vision system has been deactivated since its primary ability is detecting white lines on grass. In this challenge that would pose as threat more than an aid since the GPS coordinates are outlined in white lines. Extending from this issue is the case of boundary points. However, using the explicit graph developed with the original coordinates a convex hull can be developed and as long as the implicit path is within the convex hull the path is valid and the robot will execute the proper commands.

## 5. JAUS

After the 2006 IGVC competition JAUS research was started in order to participate in the 2008 JAUS challenge. Since the team did not participate in the 2006 JAUS challenge we had to start from the beginning.

The Wunderbot 4 can read and execute the JAUS message commands from the operator control unit through the 802.11g data link. During the JAUS challenge the Wunderbot 4 will be set to monitor for JAUS messages and check for incoming JAUS commands. At this level of implementation these messages will start the Wunderbot 4 moving forward in autonomous mode, stop the Wunderbot 4 from moving in autonomous mode, and activate a warning device (sound file output to speaker) and report position.

In the JAUS software the UDP Ethernet connection is opened to listen for JAUS messages coming in through that port and IP address. When a message is received it is checked for the UDP header information containing the ASCII equivalent of "JAUS01.0" then parsed out of the UDP header. The incoming IP address and port are rechecked in code to ensure that they are correct for receiving data and carrying out commands. Next the message properties are parsed out and output to the front panel of the LabVIEW program. The next piece of information that is in the JAUS header is the command code followed by the destination ID, the source ID, the data control, and sequence number. For the competition there is no data control information being sent or sequence number. After the command code is parsed out of the byte array the command string is sent to the JAUS Command VI to carry out the command.

## 5. Performance

The Wunderbot 4 system maintained the same specifications as the previous platform; however, the results were retested and verified.  These are shown in Table 5.

| Category | Required Result | Expected Result | Confirmed Result | Unit |
|---|---|---|---|---|
| Speed | 5 | 5 | 5 | Mph |
| Ramp Climbing | 15 | 45 | 30 | Degree include |
| Stopping Distance | 6 feet, 15% incline | Immediate | Immediate | -- |
| E-stop range | 50 | 1000 | 100 | Feet |
| Payload | 20 | >20 | 100 | Lbs |
| Battery life | 30 | 240 | 120 | Minutes |

**Table 5 – Performance specification on Wunderbot 4.**

## 6. Cost

The budget for this year was minimal as many of the changes were software. Therefore, we have supplied the estimate in total robot costs from last year plus the additional hardware changes made by the current team to the Wunderbot platform.  The updated budget is shown in Table 6 and a full budget breakdown is available in Appendix G.

| Item | Price |
|---|---|
| Remote control | $200 |
| Tower parts | $150 |
| Plexiglass | $50 |
| Electrical blocks | $60 |
| Wiring conduit | $50 |
| Laptops (3) | $1200 |
| **Subtotal** | **$1710** |
| Estimated project 2006 net worth | $31,000 |
| **Total Cost** | **$32,700** |

**Table 6 – Budget for the Wunderbot 4 team. This figure does not include additional costs incurred by the club including but not limited to IGVC travel costs.**

## 7. Social Contributions

The Wunderbot 4 team has provided many social contributions both technical and non-technical. The technical avenue includes publishing an IEEE paper, meeting with industry sponsors, and structuring learning opportunities for other students within the department on topic such as control theory, signal processing, and quality assurance. In the non-technical venue the team has worked with high school students sparking their interests in robotic design. They have been featured in numerous media outlets including television, newspaper, and Internet publications. It is always on the team's agenda to further the discussion and developing of robots society to aid in all aspects in the home, classroom, and/or in space.

## 8. Conclusion

The Wunderbot continues to be a platform for undergraduate student research in the RMI Lab of Elizabethtown College and has provided many opportunities educationally and professionally to its students. The team would like to take this opportunity to thank the judges and organizers of the IGVC for their hospitality and the team looks forward to a successful competition this year!

## 9. References

[1] . Siegwart and I.R. Nourbaksh, Introduction to Autonomous Mobile Robots, The MIT Press, Cambridge, Massachusetts, 2004

[2] I. Ulrich and J. Borenstein, "VFH+: reliable obstacle avoidance for fast mobile robots", IEEE International Conference on Robotics and Automation, Leuven, Belgium, May 16-21, 1998, pp. 1572-1577

## Appendix Listing

Appendix A – Software flow diagrams

Appendix B – O$^3$: An Optimal and Opportunistic Path Planner (with Obstacle Avoidance) using Voronoi Polygons (ISBN:

Appendix C – Vision System for Wunderbot 4 Autonomous Robot

Appendix D – IGVC Way Point Navigation Solution; Case Study: Wunderbot 4

Appendix E – The Joint Architecture for Unmanned Systems: A Subsystem of the Wunderbot 4

Appendix F – Simulations results of Wunderbot 4 path planning

Appendix G – Complete budget of Wunderbot platform starting with Wunderbot 0.