Big Blue

University at Buffalo

Intelligent Ground Vehicle Competition 2011

Dominic Baratta, Sean Bicknell, *Brett Bowman, Ben Deuell,*

Colin Lea, Christian Nugent, Bich Vu, Willem Rohl-Hill

I certify that the engineering design of the vehicle described in this report was done by the current student team and has been significant and equivalent to what might be awarded in a senior design class.

Dr. Jennifer Zirnheld
Department of Electrical Engineering
University at Buffalo

1 Overview

UB Robotics, an undergraduate student-run organization at the University at Buffalo, presents substantial revisions to **Big Blue**, a robot that was first introduced in the 2009 Intelligent Ground Vehicle Competition. Significant efforts have been made to the software and electrical components of our current unmanned ground vehicle. At the 2009 competition, Big Blue placed 12th overall and successfully completed the Interoperability Challenge. In 2010, they were 7th in their design group but were only able to qualify due to hardware issues at competition. The goal for 2011 was to become more competitive in the Autonomous Challenge.

The aim for the 2010-2011 school year stems from feedback received from the previous competition as well as problems seen in the exhaustive analysis and review process. Notable changes have been made to the electronics, software algorithms, and safety mechanisms for operation. The entire platform is documented and major changes are noted with a ★.

1.1 Team Structure

Current members range from freshman to seniors, all of whom are pursuing their undergraduate education. Many new subtopics within vehicle autonomy and circuit design were investigated and the club's recent accomplishments represent a comprehensive understanding of mobile robotics. The IGVC team structure is as follows:

Project Leader

Ben Deuell, ME '12

Hardware Leader	**Software Leader**
Ben Deuell, ME '12	*Dominic Baratta, CS '12*
Christian Nugent, EE '12	Colin Lea, ME '11
Brett Bowman, EE '12	Bich Vu, CSE '13
Willem Rohl-Hill, EE '14	Sean Bicknell, ME '14

ME = Mechanical Engineering, EE = Electrical Engineering

CSE = Computer Engineering, CS = Computer Science

1.2 Design Process

Figure 1 represents the three-year design process that UB Robotics has employed. The flow represents an iterative approach emphasizing simulation and testing. When possible, physical prototypes are tested before spending large amounts of time and resources manufacturing full-scale components. Simulation is used in all domains whenever possible: CAD for mechanical design and software simulation for algorithm development.

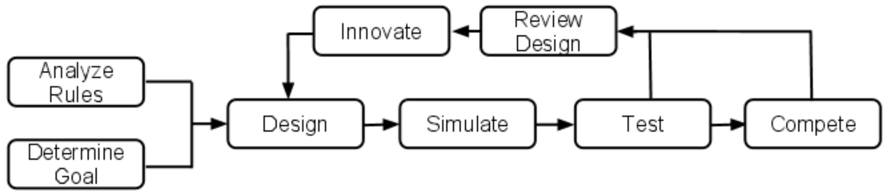

Figure 1 **Overview of the Design Flow Implemented by UB Robotics**

UB Robotics feels that outreach and dissemination of information is important for promoting the field of robotics as well as self-reflection. By developing tutorials and workshops on tools useful to the competition, students not only develop a deeper understanding of the content they are teaching but are able to help others learn valuable skills. This also leaves a legacy, which aids in documentation and assists new members in climbing the learning curve. Tutorials are available in both written and video format on the UB Robotics website [1]. Additional demonstrations have been done this year at the Buffalo Public Library and regularly at University at Buffalo.

1.3 Focus Areas

At the 2010 competition, Big Blue faced critical hardware problems that almost stopped it from competing. The custom motor controllers had problems due to large current draws causing some of the transistors to overheat and malfunction. To combat this issue, the electrical team added new speed controllers that would be able to withstand these high currents. In addition, the mechanical team worked to implement drive mechanisms to maximize mobility while minimizing power and current consumption.

The previous UB Robotics software team built a hardware abstraction layer with many of the features necessary for robust autonomous navigation. Significant work for the 2011 competition focused on designing a new model-based software platform and a more advanced computer vision system used for lane detection and recognizing objects.

IGVC 2011 Focus Areas

Hardware	**Software**
• Circuit Diagnostics • Motor Controllers	• Model-based Software • Lane Detection • Object Recognition

It is estimated that over 2500 voluntary man-hours have been put into Big Blue over the past year without class credit or monetary compensation. Weekly meetings are held to discuss updates and open hours are hosted regularly to facilitate active membership. Integrating the hardware and software teams is important for physical development and implementation, thus joint weekly meetings were held.

2 Mechanical Design

An in-depth background of Big Blue's chassis and general hardware design can be found in the 2010 IGVC Technical Report [2]. In this document, focus is on technical details and recent innovative efforts. All hardware designs were first developed using Computer Aided Engineering tools such as Autodesk Inventor and PCB Artist. A complete test platform was developed to prototype the new additions before the full-scale models were manufactured.

2.1 Chassis and Drive Train

The design goal of Big Blue's chassis and drive train was to establish a rugged, reusable platform capable of navigating diverse outdoor terrain. A four-wheel direct-drive scheme was used to increase speed capabilities and provide zero point turning. Zero point turn is especially important for software control in order to simplify the motion planning process. Additional consideration was placed on keeping a low center of mass and making components easily accessible.

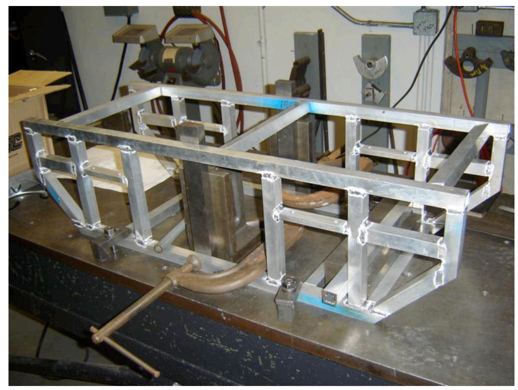

Figure 2 Bottom Portion of the Chassis

The chassis was developed with an upper and lower half. Heavy parts such as motors and batteries are placed in the bottom half, and circuit boards, sensors, and the computer are placed in the top portion. The welded frame was manufactured using 1" square tubing. Finite element analysis within Autodesk Inventor was used to confirm structural integrity [2]. Big blue uses four NPC Robotics T64 brushed DC motors running on 24V with an output of over 0.7 horsepower. Experimental results show the vehicle can travel at speeds up to 10 miles per hour.

2.2 Mecanum Wheels

In order to navigate a curve, a four-motor differential drive system requires wheels to slip. This causes localization issues, puts added stress on the motors, and requires greater amounts of electrical current to navigate. Problems such as these were not fully taken into account during the original design of Big Blue. These were resolved in 2010 with the creation of custom Mecanum wheels.

Mecanum wheels have a series of rollers that are placed along a wheel hub at 45 degree angles, which allow the vehicle to move forward and laterally [3]. Recent publications demonstrate vehicles with Mecanum wheels attached to all four motors allowing movement in any direction [4]. Note that the goal of using these wheels was not to develop a non-holonomic vehicle, but to turn with greater efficiency and control. Putting them only on the front motors increases mobility and decreases current draws on the motor controllers.

In the previous drive system, wheel slippage was highly unpredictable which made encoder data unreliable while turning. It was also difficult to calculate how far the wheels must rotate to turn the robot, so sensor feedback was crucial to controlling the robot. The relation of the Mecanum wheel rotation to the robot movement is highly predictable so encoder data and localization are greatly improved.

The Mecanum wheels are much more efficient than the previous drive system because they eliminated the need for the wheels to drag across the ground while the robot is turning. Previously, dragging of the rigid wheels was found to be a large waste of energy. The improved efficiency allowed the robot to operate 150% longer on the same batteries.

Figure 3 Mecanum Wheels (a) CAD Design (b) Final Product

The size and ruggedness of wheels required for Big Blue are unavailable through commercial-off-the-shelf (COTS) solutions, thus the design was developed and manufactured in-house by UB Robotics. Considerations were placed on ruggedness and durability. The rollers on the COTS wheels investigated are continuous and are meant for smooth, indoor surfaces. Grooves in the UB Robotics design provide greater traction for the competition's outdoor environment. Each wheel has twelve rollers equally distributed on 8.5 inch rims. Rollers are made of a two part urethane cast in a silicone mold that was molded around a custom aluminum master part created on a CNC lathe. The design was developed in Autodesk Inventor (figure 3a) and then imported into FeatureCAM to generate the tool paths and code for the lathe.

With a year of testing, we have determined the following in regards to our Mecanum wheels. The high roller per wheel count allows the wheels to roll smoothly on flat the ground. However, it forces the diameter of the rollers to be relatively small and thus decreases their ability to climb over obstacles. The Mecanum wheels have some difficulty climbing over obstacles when a wheel is moving sideways. This occurs when performing a zero point turn where the robot is alongside a vertical step. If the robot was also moving forward, as is the case in an arcing turn, the large diameter of the wheel helps it climb over obstacles. The Mecanum wheels have no problems moving over the terrain presented in the competition, however, larger rollers would improve the robots mobility in rougher terrain.

2.3 Sensors

Big Blue houses a suite of differential and absolute sensors used to determine its location, vehicle motion, and objects on the course. A Novatel ProPak-V3 differential GPS is used to track the global position. The GPS is WAAS-enabled and outputs positional data with three standard deviations of 10 centimeters using the support of an Omnistar HP subscription. A PNI 3-axis digital compass with pitch/roll compensation is used to determine the current heading with resolution of 0.1 degrees

Objects are detected on a 2D plane using a SICK PLS101 laser rangefinder (LIDAR), which outputs range data to targets up to 50 meters over a field of 180 degrees.

2.4 Power Supply

A custom power supply board was created in 2009 to supply power to each component. Four rails distribute power at 24, 12, and 5 Volts. The 24V rail is an unregulated source connected to the motor controllers. The rest of the rails are regulated and are enabled with individual channel switches. A soft-start circuit charges the capacitors before starting the main system. A keep-alive circuit was created for the GPS to eliminate the need to reconnect to satellites every time Big Blue restarts.

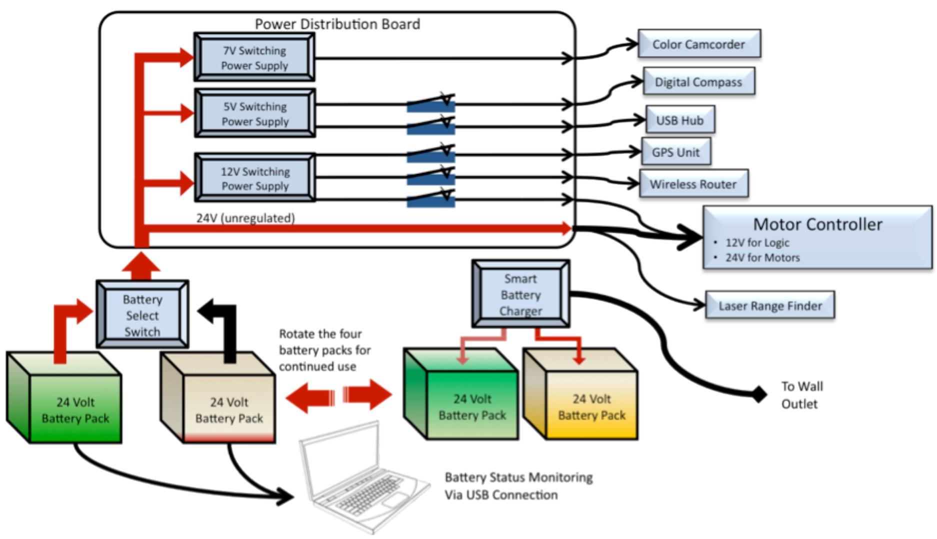

Figure 4 Power Distribution

2.5 Batteries

Two 24V battery packs are installed in the vehicle at a time. Each pack contains two 12V lead acid batteries in series. Power can be switched between packs without shutting the system down. A total of four battery packs are rotated successively in order to age them at the same rate. A battery monitor was custom designed to monitor the voltage levels in order to maximize performance.

2.6 Motor Controllers

New for this year are RoboteQ speed controllers rated to withstand 120 amps of peak current, and realistically 60 amps for over one hour. Each controller has two channels capable of driving two separate motors. The decision to implement these controllers instead of the custom-made H-bridges was due to several reasons. The first being that the high current drawn by the motors in previous years led to a high amount of stress on the previous controllers which were only rated to 30 amps. This high current often resulted in blown MOSFET chips. Next, these controllers have a built in PID function to adjust wheel velocity to the desired speeds, which prevents the motors from stalling and drawing too much power. While our previous custom controllers had this function, using PID that is built into the device helps reduce the amount of calculations for the microprocessor. Motor commands are received from the computer and encoder data is transmitted back over a USB interface. As a safety measure, the emergency stop cuts power to the motor drivers via logic gates rather than through firmware.

2.7 Remote

A custom rapid-prototyped remote was developed in 2009 for wireless communication with Big Blue. All information including the emergency stop, joystick positions, and menu buttons are sent as packets to a UART interface on the motor controller which uses a 418MHz RF Transceiver. The packets are filtered to check for errors in transmission resulting in a more reliable method of communication.

2.8 Power Consumption

The majority of power consumption comes from the motors. Figure 5 details the breakdown for components in Big Blue.

	Voltage (V)	**Power (Watts)**
Motors	24	480
LIDAR	24	17
GPS	12	2.8
Compass	5	0.1
Camera	8	5.5
Circuitry	5	0.6
USB Hub	5	2.5
	Total:	**508.5**

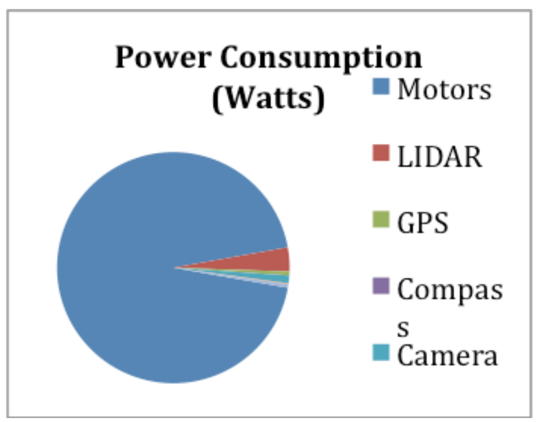

Figure 5 Power Consumption

3 Software Design

Significant work has been put into redesigning the underlying software architecture that runs Big Blue this year. For the previous two years, UB Robotics has been using the same base structure, which uses a client-server model. The decision was made to replace this aging system for the 2011 competition.

RobOS2 features a system design similar to that of a "Model View Controller" which is often utilized in web applications. This type of design gives the software suite an excellent ability to adapt to new challenges easily. If a new type of data is encountered, a new "Model" can be created to store the data in a way that allows all of the necessary controllers access to it. Should a new computation challenge present itself, all that has to be implemented is a different controller that takes input from the standardized data in the models. The "view" portion of the system is used to output information from either the controllers or the models to some form of human- readable output (e.g. Graphical User Interface, System Logs, etc.).

As in previous years, the software was developed targeting both the clubs dual core Dell laptop and Java SE 6. This was done to control system costs, as well as provide an environment that facilitates bringing new members onto the project easily. The only exception to this is the computer vision component. This is written in Python and communicates to RobOS 2 via TCP/IP sockets. All of the software was developed utilizing locally hosted SVN repositories to enable multiple team members to collaborate on the project at once.

3.1 Autonomous Challenge

Since the 2010 competition, Vector Polar Histogram Plus (VPH+) has been used for path planning in the Autonomous Challenge. This relies on a local map from the vehicle and attempts to find the safest and farthest direction in which to move. The algorithm is based on a published implementation [5] and is modified to suit the needs of this challenge. Our method has changed slightly this year to accommodate traveling to various waypoints within the challenge. The output from our global planner, A*, is input into VPH+ to direct Big Blue to the next waypoint. In previous years, the biggest problem was lane detection, so significant effort in 2011 was directed towards computer vision.

3.2 VPH+

VPH+ functions by transforming a set of data in polar form into a binary histogram. Data is merged from the current LIDAR data and line boundaries marked by the camera. A function is used to find a "safe" distance in each direction, as determined by equation 3.2.1. This value is compared to the distance from the vehicle to the nearest object at each angle. Parameters V, θ_i, a, and *Dsafe* are used for velocity, target angle, deceleration rate, and safety distance.

$$D(\theta) = \frac{V^2 \cdot cos^2(\theta_i - \frac{\pi}{2})}{2a} + V + D_{safe}$$ (eqn. 3.2.1)

Target directions are determined by free spaces indicated by a "1" in the binary histogram. Targets are filtered based on an angular safety distance, eliminating choices that are too close to hazardous objects. In situations where there are less than a nominal number of targets an artificial point behind the robot is chosen. This requires the robot to turn around and search its environment.

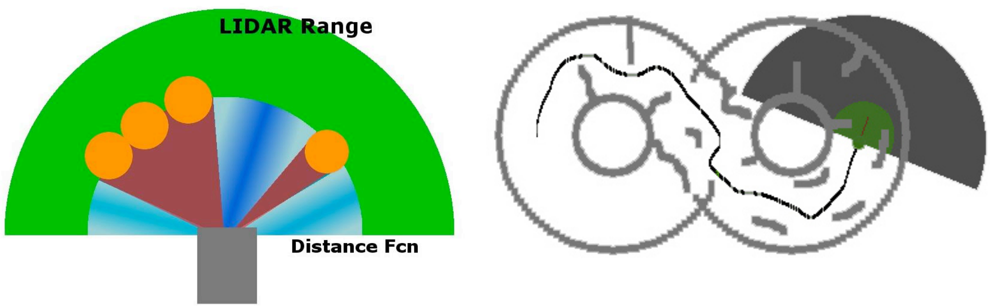

Figure 6 **(a) VPH+ Diagram (b) Simulation using VPH+**
Shades of blue indicate cost where darker blue is larger in value. Red denotes an invalid direction.

Points are grouped into different objects based on their proximity to other nearby points. If the distance between two sequential angles is less than a certain value it is concluded that they both belong to the same object. Directions encompassed by closer objects are eliminated from the target directions. This is the main advantage to the VPH+ algorithm over its predecessors Vector Polar Histogram and Vector Field Histogram.

A cost function determines the final direction in which to move. Cost is developed with the idea that there is not necessarily a predetermined goal. Note, however, we are able to guide the vehicle towards each of the waypoints for the 2011 Autonomous challenge. Big Blue should move forward or towards its goal as far as it can while minimizing turning and maximizing safety. The safety factor is based on the angular distance to the closest "closed" angle. The cost function is shown by equation 3.2.2. The final direction is based on the maximum cost. Parameter θ_i refers to the target angle and K_s and $K\theta$ are tunable coefficients reflecting the weighting of the safety and heading factors. Figure 6 depicts the target directions and their calculated cost.

$$Cost(\theta_i) = K_s D_s - K_\theta \cdot (\theta_i - \frac{\pi}{2})$$ (eqn. 3.2.2)

This algorithm has proven to provide safer navigation over Big Blue's previous A* based method. Simulation in figure 6 shows the robot navigating a course without hitting anything. The closest obstacle comes 0.4 meters away from the robot on its side.

3.3 Computer Vision

This year UB Robotics presents a novel vision system that is capable of detecting and classifying features in the Autonomous Challenge. We are capable of finding lane markers and objects such as cones and barrels. Classification is important this year in order to navigate to the left and right of colored flags that will be placed on the course. There are three stages in our vision system: preprocessing, detection, and classification. The detection phase uses a novel graphical model technique that has been submitted to the International Conference on Intelligent Robots and Systems (IROS) [6]. Video footage was taking from the practice course at the 2010 IGVC competition for testing our system.

Preprocessing

Initial processing of the input image plays in important role in forming our model. The foreground and background must be differentiated using the camera's color image for use in feature detection. The saturation component of the Hue-Saturation-Luminance (HSL) color-space was chosen as a basis for its superior ability to differentiate the lanes and objects from the background. However, there is a problem where shadows sometimes are too close in color to the features and trigger false positives. By combining information from both the saturation and luminance channels of HSL we are able to define a better initial image. A morphological opening filter is then applied to eliminate noisy pixels.

$$I_{hybrid} = \max(I_{sat}, I_{lum} < \alpha)$$

Figure 7 shows the saturation and hybrid channels and their respective preprocessed binary images. Both channels are thresholded at separate values, $\alpha_{\{S,L\}}$, which are based on the peaks in the histograms of each channel.

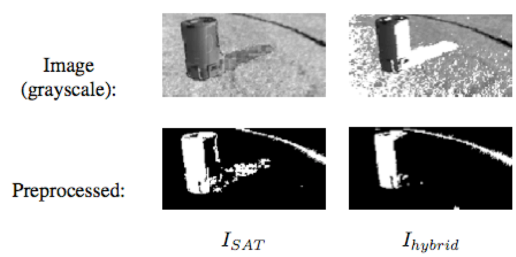

Figure 7 Preprocessing Comparison

Detection

The preprocessed image outputs binary values that do not differentiate individual objects in the scene. This is a problem because in our application the white lane markers are often faded or muddy and have the same texture as grass. Furthermore, using binary labels prevents the detection of multiple overlapping objects. Classification thus becomes a problem because features from two or more objects may be combined into one segment. Using Hierarchical Markov Random Fields we are able to implement a more accurate and robust system for segmentation and classification.

Our model, shown in Figure 8, performs two operations: denoising and inference. We have developed a Hierarchical Markov Random Field

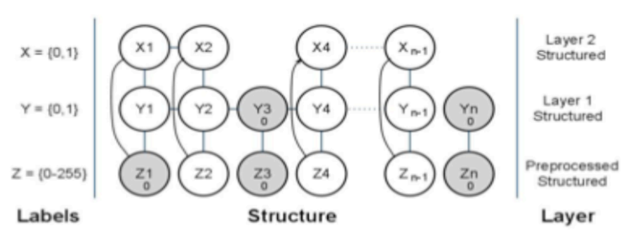

Figure 8 Hierarchical Markov Random Field Model

(MRF) using two fully-connected layers. An MRF is a graphical model used to find an "ideal" image from an input image. They are often used for applications such as image restoration and image segmentation. The first layer denoises the image and the second layer infers a label for each pixel. This is used to differentiate different objects in the scene to aid with classification. Outputs of the HMRF are seen in Figure 9 using test footage from the practice course at IGVC 2010.

Figure 9 A Hierarchical Markov Random Field model is used to perform multi-object detection in near real-time in order to classify course features.

Classification

Objects are classified based on their identifying characteristics. A decision tree takes in features from each object and designates a class. Through analysis of pixel count, area, placement of the centroid, and other features shown in Figure 10a we empirically constructed the tree seen in Figure 10b.

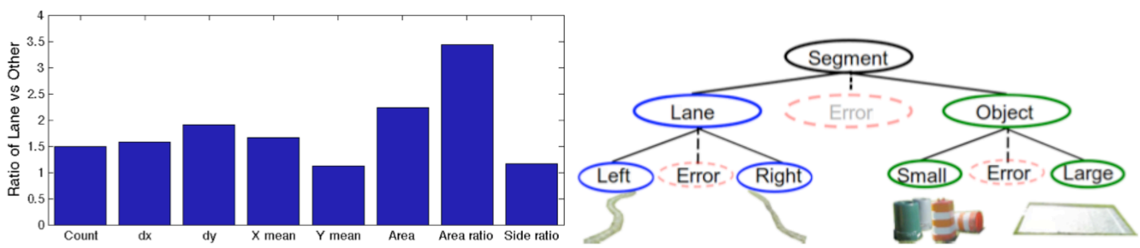

Figure 10 (a) Comparison of 'Lane' segment features versus 'Barrel' segment features (b) the decision tree designed for object classification.

In many areas of the practice course the system achieves 93% accuracy, however, problems in areas such as the switchback decrease the overall accuracy to approximately 70%. Methods to fuse video and LIDAR data in troubled areas are still being investigated.

3.4 Navigation Challenge

Planning for the navigation challenge is done using both local and global path planning techniques. On a global level the A* algorithm creates paths connecting each of the waypoints while avoiding known obstacles. A* provides an optimal closed set path from the robot to the goal [6]. The path is augmented using VPH+ as described in section 3.2. For this challenge, the cost function in VPH+ is adjusted to follow the direction of the line provided by A* rather than attempting to keep the current heading. Combining A* and VPH+ provides a safer way to avoid obstacles and generates a smoother path.

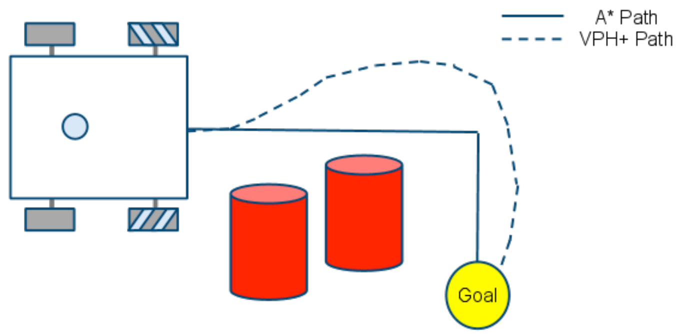

Figure 11 **Comparison of A* and VPH+ using A* for guidance**

3.5 Mapping

In both challenges, the world is only partially observable. This means that we can not see everything on the course at the same time. This is a greater problem in the Navigation challenge because Big Blue has to remember obstacles it has previously seen. At every time step, a local map replaces information in a global map. This replacement technique is used to reduce map smearing due to localization issues. The global map uses a quad-tree custom-implemented to store object's Big Blue has discovered. The Autonomous challenge relies on a local map, which includes lane boundaries from the on-board camera, as well as LIDAR data.

3.6 Localization

An Extended Kalman Filter (EKF) is used to provide refined localization using GPS, odometry, and compass sensor data. The EKF is a Gaussian-based filter that linearizes the vehicular model with Taylor series expansion using the state model seen in equation 3.6.1 [7]. Covariance is calculated with respect to sensor measurements and the predicted state, which is used to weight each input differently during the update phase. Redundancy in sensors by means of differential and absolute measurements provides more accurate localization data. For example, when the vehicle is not moving higher weighting is put on the encoders due to random deviations in GPS data. However, when the vehicle is turning the GPS is weighted more heavily because the encoders provide a less accurate motion model.

$$X = [x \; y \; \theta \; \dot{x} \; \dot{y} \; \dot{\theta}]^T$$

3.7 Control Feedback

A Proportional-Integral-Derivative (PID) controller is used to govern Big Blue's wheel speed. It is assumed the motor varies linearly with voltage input. The output is dependent on the current error, rate of change in error, and accumulation of error as calculated by equation 3.7.1. PID guarantees the wheels are actually spinning at the speed specified by the software.

$$u(t) = k_d \dot{e}(t) + k_p e(t) + \int k_i e(t)$$ (equation 3.7.1)

Closed loop control allows the vehicle to follow a trajectory with greater accuracy. It also prevents the motors from stalling and improves response time. Figure 12a overlays images of Big Blue tracing a circle over time. The red line is superimposed for visualization.

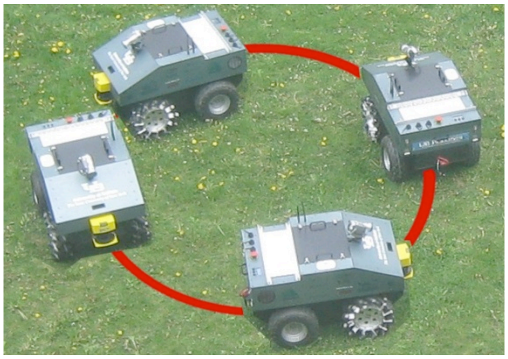

Figure 12 (a) Trajectory Tracing (b) Big Blue Test Platform

3.9 Testing and Simulation

Before completion of Big Blue's new hardware, testing was done in simulation and on a prototype robot. A test platform was constructed and evaluated with off-the-shelf Mecanum wheels. The kinematic equations were evaluated to test correctness and a LIDAR was employed to test path planning algorithms. A laptop running RobOS is placed on top of the vehicle seen in figure 12b.

3.10 Interoperability Challenge

Big Blue completed the Interoperability Challenge during the 2008 and 2009 competitions by implementing the JAUS communicaiton protocol. In both occasions, JAUS was tested with simulation software developed in-house before competition. The way that JAUS is implemented in RobOS2 is slightly different than in the original version of the system. The JAUS subsystem is now implemented as a "model" for the UDP packets as well as a "controller" which interprets the messages received from the COP.

4 Performance

Big Blue has exceeded expectations in regards to ruggedness and response. The vehicle can travel at upwards of 10 miles per hour and has ascended hills with an angle of over 55 degrees and about 0.5 seconds to go from active to stopped. Big Blue's response and speed come with drawbacks. Each battery pack lasts about 30 minutes. Thus, with its two on-board packs the total battery life is 60 minutes. Note that the introduction of Mecanum wheels has increased battery life by a factor of 1.5.

Performance Results	
Speed	10mph
Reaction Time	Near Instant
Battery Life	30 minutes/pack (2 packs onboard)
Ramp climbing	55° +
Object Detection Distance	5 meters for lines/20 meters for objects
Waypoint accuracy	30 cm

Figure 13 **Performance Results**

4.1 Course Complexities

The implementation of VPH+ running on Big Blue compensates for dead ends. In safe situations the algorithm can always find multiple target travel directions. If there are less than a small specified number of targets then the vehicle turns around and detects an open path. Using a local path planning algorithm eliminates the goal seeking problem that global planners have with switchbacks. Because VPH+ resists turning (while optimizing for safety) it does not have this problem. Simulation (figure 6) shows that Big Blue successfully traverses switchbacks.

4.2 Cost

Big Blue is considered a research vehicle, thus its cost is substantiated by its high-accuracy sensors, well-manufactured parts, and custom electronics. A cost breakdown is shown in figure 14.

Component	Retail Cost	Team Cost
Dell Latitude D830 Laptop	$1,200	$0
Novatel Propak V3 DGPS	$8,000	$3,900
SICK PLS-101	$5,000	$215
NPC Motors	$1,144	$572
Batteries	$250	$250
PNI TCM-2.6 Digital Compass	$850	$0
Panasonic 3CCD color camera	$800	$0
Custom Electronics		
Motor Controller	$725	$525
Remote Board	$250	$250
Power Supply	$260	$260
US Digital E4 optical encoders	$150	$150
Mechanical Parts (Metal, hardware)	$1,250	$1,250
Anodizing	$100	$100
Total	$19,980	$7,472

Figure 14 Cost breakdown for Big Blue

5 Conclusion

With the addition of new motor controllers and a more sophisticated software system - including better lane detection and object recognition – this year's Big Blue represents substantial change for the 2011 IGVC. UB Robotics is confident in their efforts and believes that these additions will ensure success in the 2011 Intelligent Ground Vehicle Competition.

Acknowledgments

UB Robotics would like to thank all its supporters who provided sponsorship and donations in the form of hardware, services, and financial support making Big Blue possible: Novatel, Omnistar, PNI, Advanced Circuits, Sunstone Circuits, Seneca Platers, Inc., UB Student Association, Sub Board-I, UB Engineering Machine Shop, and the Energy Systems Institute (ESI) at the University at Buffalo. Special thanks go to the club adviser, Dr. Jennifer Zirnheld, for her continued support, as well as the rest of the ESI staff and UB Robotics alumni for all of their help.

References

[1] *UB Robotics*. Available: http://www.eng.buffalo.edu/ubr

[2] D. Baratta, B. Deuell, C. Lea, C. Nugent, et. al. , "Big Blue: University at Buffalo's Entry into the 2010 Intelligent Ground Vehicle Competition " May 2010.

[3] B. Ilon, "Wheels for a course stable self propelling vehicle movable in any desired direction on the ground or some other base ." U.S. Patent 3 876 255, 1975.

[4] S. Y. A. Akira, P. Viboonchaicheep, and K. Samura, "Mecanum-wheel Vehicle Systems Based on Position Corrective Control," in Proceedings of the Annual Conference of the IEEE Industrial Electronics Society, 2005.

[5] Y. D. J. Gong, Y. Man and G. Xiong. , "VPH+: An Enhanced Vector Polar Histogram Method for Mobile Robot Obstacle Avoidance," presented at the Proceedings of the IEEE International Conference on Mechatronics and Automation, 2007.

[6] (In Review) C. Lea, J. Corso. "Efficient Hierarchical Markov Random Fields for Object Detection on a Mobile Robot." International Conference on Intelligent Robots and Systems (IROS), 2011.

[7] S. Koenig, Likhachev, M., Liu, Y. , Furcy, D., "Incremental heuristic search in AI," AI Magazine, 2004.

[8] W. B. S. Thrun, and D. Fox, Probabilistic Robotics. Cambridge, MA: The MIT Press, 2005.