UNIVERSITY AT BUFFALO

BIG BLUE

INTELLIGENT GROUND VEHICLE COMPETITION 2012

Dominic Baratta, Brett Bowman, Ben Deuel, 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 Big Blue. At the 2009 competition, Big Blue placed 12th overall and successfully completed the Interoperability Challenge. In 2010, Big Blue placed 7th in the design group but was only able to qualify due to hardware issues at competition. In 2011 Big Blue placed 19th in the autonomous competition and 8th in design group B.

The aim for the 2011 - 2012 school year stems from feedback received from the previous competitions 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 $\star$.

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:

Table 1: Team Structure

Project Leader Dominic Baratta, CS, '12	
Hardware Leader Brett Bowman, EE 12	Software Leader Dominic Baratta, CS, '12
Christian Nugent, EE 12 Ben Deuell, ME 12 Willem Rohl-Hill, EE 14	Bich Vu, CSE 13

ME = Mechanical Engineering, EE = Electrical Engineering
CSE = Computer Engineering, CS = Computer Science

1.2 Design Process

Figure 1 represents the four-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.

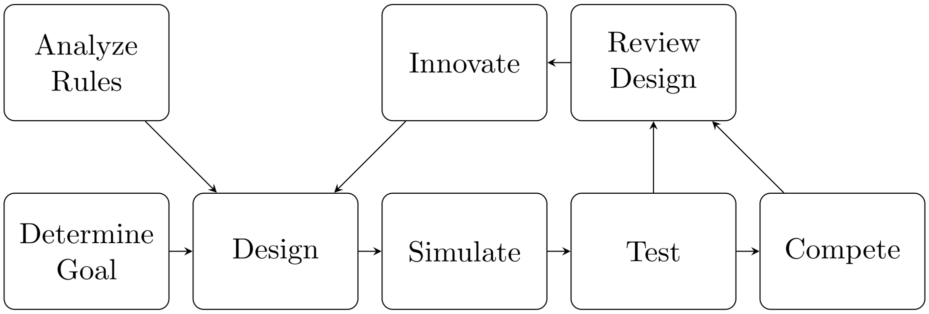

Figure 1: Overview of the Design Flow Implemented by UB Robotics

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 Museum of Science and regularly at the University at Buffalo.

1.3 Focus Areas ⋆

While critical hardware issues which plagued Big Blue during the 2010 competition were fixed for the 2011 competition there were still many areas of the system which suffered minor malfunctions.

The computer vision subsystem was rewritten for the 2011 competition and is the only component which is not implemented in the Java programming language. This subsystem which is implemented in Python utilizing OpenCV communicates with the rest of the system via a local TCP socket. During Big Blue's competition runs the socket crashed several times causing the main system to not be able to receive updates on where lines and obstacles were on the course. In order to prevent this from happening the socket layer has been rewritten for the 2012 competition to include threading, error checking and automatic reconnects in the event the socket should close due to an error.

Due to changes to the autonomous course and the removal of the navigation course at the 2012 competition significant effort was put into developing more efficient data structures to store maps and improving path planning and navigation algorithms. These changes allow Big Blue to utilize both global path planners to navigate to the way points as well as local path planners to navigate around local obstacles such as switchbacks.

Table 2: IGVC 2011 Focus Areas

Software Stability	Map Storage Data structures
Radio Control	Navigation & Path Planning

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 capability for zero point turning. The ability to perform a Zero point turn is especially important for software control in order to simplify motion planning algorithms. Additional consideration is placed on keeping a low center of gravity as well as keeping system components easily accessible.

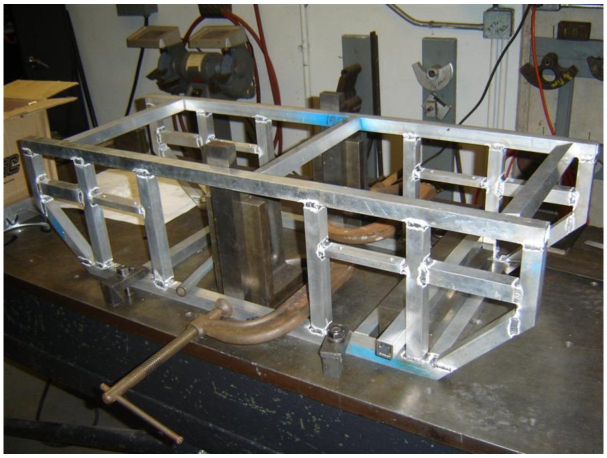
Figure 2: Bottom Portion of the Chassis

The chassis was developed with an upper and lower half. Heavy parts such as motors and batteries are placed in the bottom half, and control boards, sensors, and the system computer are placed in the top portion. The welded frame was manufactured using 1″ square tubing. Finite Element Analysis (FEA) 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 system.

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 since they eliminate 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 allows the robot to operate 150% longer on the same batteries.

Figure 3: Mecanum Wheels CAD Design [left] Final Product [right]

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 which were molded around a custom aluminum master part created on a CNC lathe.

The design was developed in Autodesk Inventor (figure 3) and then imported into FeatureCAM to generate the tool paths and code for the lathe.

After two years of real world testing with the Mecanum wheels on the platform they have provided a few key observations for future designs. 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 range finder (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. Dual 12 volt rails and a single 5 volt rail are regulated and are enabled with individual channel switches by the power supply. A soft-start circuit charges high capacity capacitors in the power supply before starting the main system. Additionally, a keep-alive circuit was created for the GPS to eliminate the need to reconnect to satellites every time Big Blue restarts.

2.5 Batteries

Big Blue accepts dual 24 volt battery packs in order to enable a longer overall runtime. These battery packs are constructed from two 12 volt sealed lead acid battery packs which are connected in series inside of a custom enclosure. The enclosures designed for the battery packs also feature a voltage monitoring circuit which is capable of showing the current voltage of the battery packs on seven segment displays as well as sending voltage information to a computer over a serial connection.

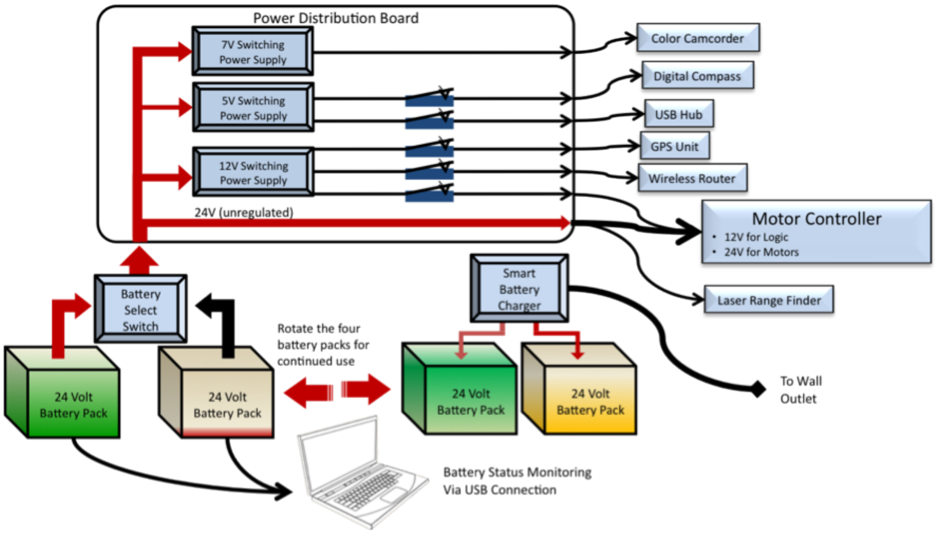

Figure 4: Power Distribution

Four battery packs were constructed and are cycled into and out of the system in order to age them evenly and prolong their lives.

2.6 Motor Controllers

Big Blue features two RoboteQ dual channel electronic speed controllers designed to withstand 120 amps of current per channel for 30 seconds. These controllers can handle a more realistic current of 60 amps for over one hour, as well as surge currents above 250 amps. After testing these controllers for over a year they have proven themselves as an excellent fit for the platform, eliminating issues which plagued previous solutions. Additionally, the RoboteQ controllers include a built in closed-loop PID subroutine to ensure the vehicle moves at the requested speed even if it is going up or down a hill. This prevents excessive current draws from motors which may be stalled in situations where a bit more power is needed to navigate over difficult terrain. The motor controllers receive commands over a RS-232 link either from the main control computer on board the platform or from the 900mhz XBee based remote. Furthermore, as a safety precaution the system's emergency stop circuitry cuts power 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 which utilized 418MHz radio modules due to their low cost. Packet filtering and checksums were manually calculated with these radio modules to ensure data integrity of the transmitted commands. The remote hardware has been resigned for 2012 and now utilizes 900MHz XBee Pro radios. While these radios do not have high data thruput they do have a extremely long range compared to other industry solutions. These radio modules also support hardware based AES encryption as well as packet filtering and error checking.

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

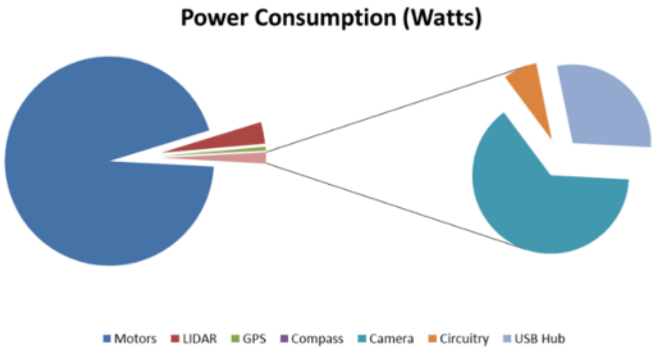

Figure 5: Power Consumption

3 Software Design

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 Java SE 7 and the clubs dual core Dell laptop. 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 module. This is implemented in Python and communicates to RobOS2 via TCP/IP sockets. All of the software is developed utilizing locally hosted Subversion (SVN) repositories enabling multiple team members to collaborate on the project as well as providing versioning history in case changes need to be reverted.

3.1 Auto-Nav Challenge ⋆

While the modifications to the autonomous challenge in 2011 required a few minor changes to the operation of the platform the changes presented for 2012 are much more in depth. RobOS2 still utilizes the Vector Polar Histogram Plus (VPH+) algorithm for planning "local" paths around objects such as cones and barrels. VPH+ attempts to find the fastest path through a situation while minimizing turning by utilizing a cost function, which has been found to produce a reliable path through tight areas. The implementation of VPH+ in RobOS2 is based off of a published implementation [5] it has been modified to better suit the auto-nav course. The A* path planner has also been leveraged from the navigation suite in order to plan an efficient and logical route between multiple way points. The A* planner is not as critical in the areas of the course where lines are present to guide Big Blue in the correct direction, however is indispensable for the way point navigation area. By combing both VPH+ and A* path planners efficient paths can be generated both on a global scale to navigate to each way point as well as quickly on a local scale to navigate around obstacles.

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 scan and line boundaries determined by the camera. A function is used to find a "safe" distance in each direction, as determined by equation 1. This value is compared to the distance from the vehicle to the nearest object at each angle. Parameters V, Θ_i, a, and D_{safe} are used for velocity, target angle, deceleration rate, and safety distance.

$$D\left(\Theta\right) = \frac{V^2 \cos^2\left(\Theta_i - \frac{\pi}{2}\right)}{2a} + V + D_{safe} \tag{1}$$

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 forces the robot to turn around and search its environment for an alternate path.

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.

Additionally, 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, it is possible to

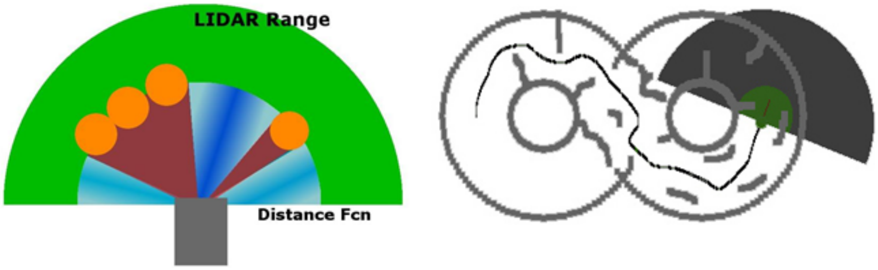

Figure 6: VPH+ Diagram [left] Simulation using VPH+ [right]

guide the vehicle towards each of the way points for the 2012 Auto-Nav 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 nearest "closed" angle. The cost function is shown by equation 2. The final direction is chosen by minimizing the cost of the path. Parameter Θ_i refers to the target angle and K_s and K_Θ are tunable coefficients reflecting the weighting of the safety and heading factors. Figure 6 depicts the target directions and their calculated cost.

$$Cost\left(\Theta_i\right) = K_s D_s - K_\Theta \left(\Theta_i - \frac{\pi}{2}\right) \tag{2}$$

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. In this simulation the nearest obstacle is 0.4 meters away from the robot on its side.

3.3 Computer Vision

The computer vision module is capable of detecting and classifying various features in a real-world environment. This system is capable of detecting driving lanes, detecting obstacles such as cones and barrels as well as performing classification in near real time. Classification is a critical part of the system in order to navigate around obstacles correctly as well as ensure correct placement on local maps which are kept for the duration of the current run. The vision module uses a three phase approach which consist of preprocessing, detection and classification. First, the image obtained from the camera is preprocessed to reduce noise and create a black and white image based on the image histogram. A graphical modeling technique published in the International Conference on Intelligent Robots and Systems (IROS) [6] is utilized for the detection phase. Furthermore, a decision tree is then utilized to classify objects accurately.

3.3.1 Preprocessing

Initial processing of the input image plays a critical role in forming the image model. The foreground and background are differentiated using the camera's color image for later 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, occasionally there is a problem when shadows 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.

$$I_{hybrid} = max\left(I_{sat}, I_{lum}, < \alpha\right) \tag{3}$$

Figure 7 shows the saturation and hybrid channels along with 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.

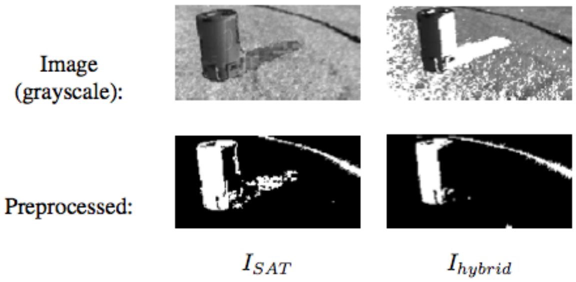

Figure 7: Preprocessing Comparison

3.3.2 Detection

The preprocessed image outputs binary values that do not differentiate individual objects in the scene. This is a problem in many applications when markers are often faded or muddy and have the same texture as the background. Furthermore, using binary labels prevents the detection of multiple overlapping objects. Classification thus becomes a problem since features from two or more objects may be combined into one segment. Utilizing a Hierarchical Markov Random Fields a accurate and robust system can be created for segmentation and classification.

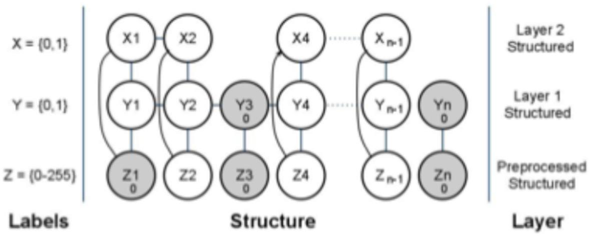

Figure 8: Hierarchical Markov Random Field Model

The model, shown in Figure 8, performs two operations: denoising and inference. We have developed a Hierarchical Markov Random Field (MRF) using two fully-connected layers. A 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 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 the 2010 IGVC.

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.

3.3.3 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 10 the tree seen in figure 10 can be empirically constructed.

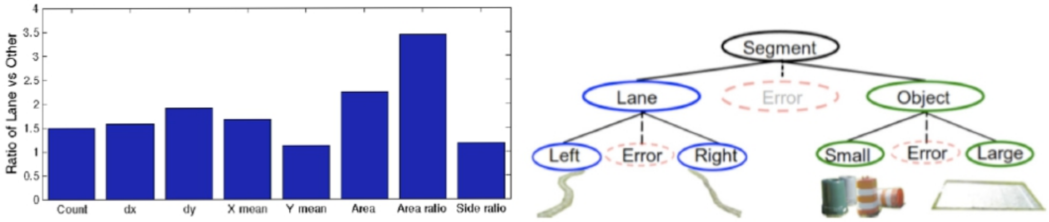

Figure 10: Comparison of 'Lane' segment features versus 'Barrel' segment features (left) the decision tree designed for object classification.(right)

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 Mapping ⋆

Even though BigBlue cannot store maps between runs due to the rules of the competition mapping techniques are still critical to it's successful operation. At any given time only part of the overall world is visible to Big Blue's sensors which can cause issues when attempting to plan logical and efficient paths around obstacles. To combat this issue RobOS2 keeps two maps simultaneously, a local map and a global map. Both of these maps are stored in quad-tree data structures for speed and efficiency. As Big Blue progresses through the world the features in the local map replace features in the global map as they are encountered. This accounts for and reduces map smearing issues that can arise due to poor localization that is occasionally seen.

3.5 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 4 [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 since the encoders provide a less accurate motion model.

$$X = \left[x y \Theta \dot{x} \dot{y} \dot{\Theta} \right]^T \tag{4}$$

3.6 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 5. PID guarantees the wheels are actually spinning at the speed specified by the software.

$$u(t) = k_d \dot{e}\left(t\right) + k_p e\left(t\right) + \int k_i e\left(t\right) \tag{5}$$

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 11 overlays images of Big Blue tracing a circle over time. The red line is superimposed for visualization.

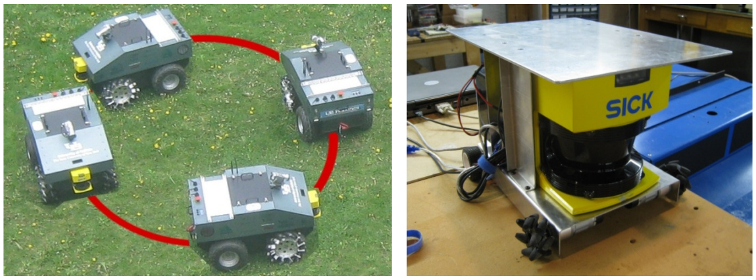

Figure 11: Trajectory Tracing [left] Big Blue Test Platform [right]

3.7 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 RobOS2 is placed on top of the vehicle seen in figure 11.

3.8 Interoperability Challenge

Big Blue completed the Interoperability Challenge during the 2008, 2009 and 2010 competitions by implementing the JAUS communication protocol. In all 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.

Table 3: Performance Results

Speed	10 MPH
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
Way point Accuracy	20 cm

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

Table 4: Cost breakdown for Big Blue

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	$7,742

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 table 4.

5 Conclusion

With upgraded radio equipment, interface boards and many software optimizations and improvements Big Blue represents a substantial change for the 2012 competition. UB Robotics is confident in the platform and believes the new additions will ensure success in the 2012 Intelligent Ground Vehicle Competition.

5.1 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] U. Robotics. `http://www.eng.buffalo.edu/ubr`.

[2] D. Baratta, B. Bowman, D. Conway, B. Deuell, C. Lea, C. Nugent, N. Ohmit, and B. Vu, "Big blue: University at buffalo's entry into the 2010 intelligent ground vehicle competition," May 2010.

[3] B. Ilon, "Wheels for a course stable selfpropelling vehicle movable in any desired direction on the ground or some other base," April 1975.

[4] A. Shimada, S. Yajima, P. Viboonchaicheep, and K. Samura, "Mecanum-wheel vehicle systems based on position corrective control," in *Industrial Electronics Society, 2005. IECON 2005. 31st Annual Conference of IEEE*, p. 6 pp., nov. 2005.

[5] J. Gong, Y. Duan, Y. Man, and G. Xiong, "Vph+: An enhanced vector polar histogram method for mobile robot obstacle avoidance," in *Mechatronics and Automation, 2007. ICMA 2007. International Conference on*, pp. 2784 –2788, aug. 2007.

[6] C. S. Lea and J. J. Corso, "Efficient hierarchical markov random fields for object detection on a mobile robot," *CoRR*, vol. abs/1111.1599, 2011.

[7] S. Koeni, M. Likhachev, Y. Liu, and D. Furcy, "Incremental heuristic search in ai," *AI Magazine*, vol. 25, no. 2, 2004.