Ball-thazar

The

Autonomous Vehicle

BOB JONES UNIVERSITY | IGVC 2007

1. INTRODUCTION

Ball-thazar is an autonomous robot designed by the electrical and computer engineering students of Bob Jones University. Despite the fact that this year’s team was composed entirely of new members, we built heavily on the work of the teams of the past two years, ADAM, and ADAM II. We named this year’s robot Ball-thazar in honor of its unique ball suspension. Balthazar is the traditional name of one of the three wise men in the Gospel of Matthew, an appropriate name for an intelligent vehicle. The introduction of a new suspension system represents just one of the ten innovative new systems, modules, and components that have been added to produce this year’s IGVC entry.

2. PHILOSOPHY

In contrast to many IGVC teams, the philosophy underlying our design decisions encourages minimal dependence on exotic or expensive technologies with a long term goal of creating an autonomous vehicle which is economical and which uses only passive sensors. Benefits to this approach include increased appeal in military applications where active sensors such as laser range finders could alert the enemy to the presence of the vehicle and where jammed GPS signals may result in an inoperable vehicle. In addition, reducing reliance on exotic sensors often makes the vehicle more cost effective and reliable. Note that the rule changes for IGVC 2007 propose obstacles that may be very difficult to identify visually, hence the addition of a backup sonar and IR systems. This anomaly to our otherwise passive philosophy is purely for IGVC competitiveness and can be disabled at will.

3. DESIGN PROCESS

As a team we followed a process of definition, innovation, development, integration, and testing. Since we started with a functional robot that had competed before at IGVC, our first step, the definition phase, was to analyze the new IGVC rules and the robot's performance in the previous competitions in order to find areas that most needed improvement. Next the team brainstormed ideas as a group in order to have a foundation from which to start the building process. We then narrowed our focus on the areas that promised to provide the most improvement and divided the tasks among the team members for development of several new subsystems. The various subsystems were then integrated into the robot and tested.

3.1 Definition Phase

Because the design team began with a functioning vehicle which competed at the IGVC in 2006, the design process began with a review of the robot's performance the previous year, as well as the new rules to which Ball-thazar would have to comply. Areas for improvement that were identified included enabling the robot to detect obstacles with colors other than orange and white, navigating more accurately in the navigational challenge, reprogramming in Python to allow rapid software changes, adding JAUS capability, and improving the vehicles handling in turns. In considering the proposed changes to Ball-thazar, feasible designs had to not only comply with IGVC specifications for the vehicle, but also had to lie within our budget constraints.

3.2 Innovation Phase

The improvement of the designated areas started with several meetings in which the team members discussed the feasibility of ways to reach the objective in each target area. The most important design change the robot needed was the capability of detecting obstacles based on a 3D ranging system instead of using only the “target” colors, orange, yellow and white, that the previous robot utilized. A stereo video approach was planned as the primary system, and sonar and infrared rangefinders were planned as backup systems. Because the vehicle had rough handling characteristics in situations where direction of travel was reversed, a freely rotating ball was conceptualized as well. Once fabrication and development actually began, the team met each week in order to monitor progress and continue redesign of areas with such needs.

3.3 Development and Integration Phases

Once the ideas were solidified, the tasks were divided among the team members according to capability, and interest. The three basic areas of required work were software, electronics hardware, and mechanical hardware. To effectively accomplish the proposed redesign of Ball-thazar, each member of the design team was assigned specific tasks. Though team members frequently worked together, each team member was expected to manage his assigned tasks. Task interdependencies were also determined, and a master schedule was created to facilitate on-schedule completion of the redesign. A website and online forum dedicated to the project helped team members stay updated on the project and any current issues.

3.4 Team Contributions

Name	**Contribution**	**Estimated hours**
Joseph Balanay Electrical Engineering Senior	GPS Receiver	75
Jonathan Baize Electronics Freshman	Ball Suspension, IR Ranging System	50
Benjamin Brame Computer Engineering Junior	Sonar System, Stereo Vision System, Windows Port, WiPort Module	200
Jordan Jueckstock Computer Science Sophomore	Odometry Module	100
Zach Mason Computer Engineering Junior	Python Rewrite, WiPort Module, JAUS	100
Joe McKirdy Electronics Senior	Motor Controller	75
Ryan Wentworth Electrical Engineering Senior	Sonar System, Ball Suspension	150
Phillip Woodhull Electrical engineering senior	Sonar System	150
TOTAL HOURS		**900**

4. INNOVATIONS

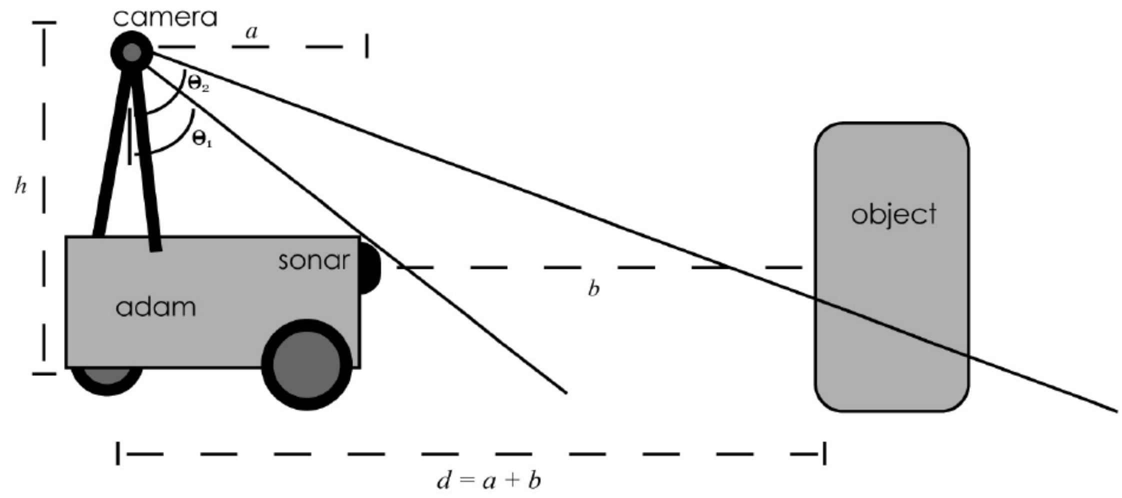
This year’s IGVC team has introduced several innovations to create Ball-thazar. While taking stock of our current robot at the beginning of the year it quickly became apparent that significant changes and additions were necessary to enable our robot to remain competitive.

4.1 Sensor Innovations

A change was introduced into the IGVC rules which allowed objects of arbitrary color to be placed on the course. Our previous obstacle detection strategy used a single camera to detect obstacles which were previously guaranteed to be red, orange, or white. The team developed three additional sensors to the robot to allow it to detect obstacles of arbitrary color.

4.1.1 Sonar

Our first addition was a sonar sensor array. Although our long term philosophy involves purely passive sensors, sonar was seen as an easy and inexpensive system that will allow us to remain competitive in case out vision systems prove to be inadequate at this year's IGVC. Five MaxSonar EZ1 sonar sensors are mounted on the front of the robot in 36˚ increments. This gives the sensors a 180˚ field of view. A PIC16F88 microcontroller is used to interface with the sonar sensors. The PIC receives distance measurements from the sensors and than transmits these measurements via RS232 back to robot using a serial port. Ball-thazar’s main software then interprets the measurements and projects the obstacles onto the robot’s primary camera image as virtual obstacles. This mapping is detailed in the figure on the following page.



h: height of the camera = 64.4 in.
d: distance from the camera to the object
a: distance from the camera to the sonar 29.9 in.
b: distance from the sonar to the object
θ_1: lowest angle that the camera can see = 40.2°
θ_2: highest angle that the camera can see = 80.2°
α_x: angle from the center of teh screen to the center of the field of view of sonar x

$$y = 240[1 - \frac{\tan^{-1}(\frac{d \cdot \cos\alpha_x}{h}{}_i) - \theta_1}{\theta_2 - \theta_1}]$$

$$x = 160[1 + \frac{\alpha_x}{\alpha_0}]$$

$\alpha_1 = -$
$\alpha_2 = -$
$\alpha_3 = 0^\circ = 0$
$\alpha_4 = -36^\circ = 0.62832$
$\alpha_5 = 72^\circ = 1.2566$
$\alpha_0 = 56.5^\circ = 0.995$

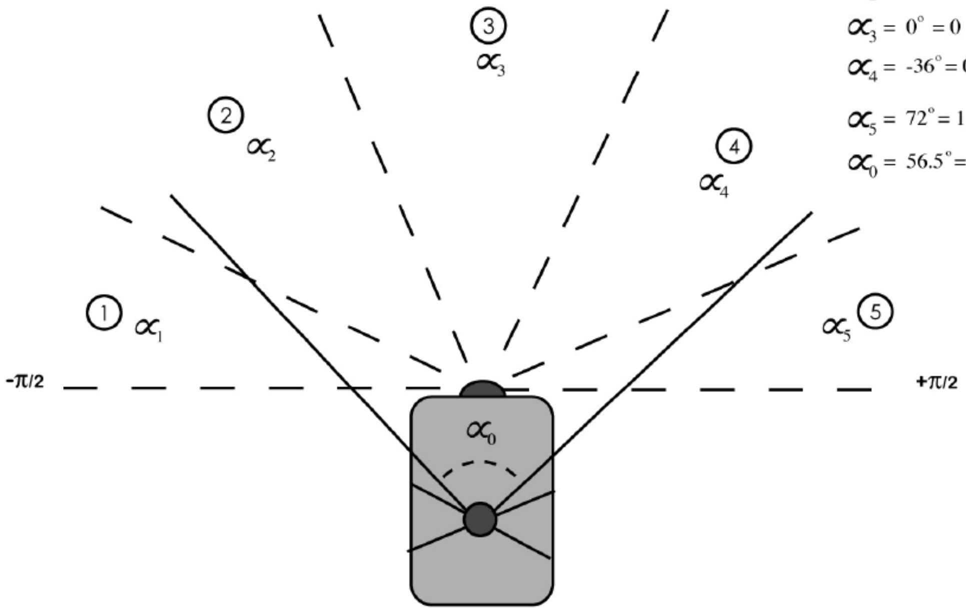


Figure 1 - Mapping of sonar obstacles into virtual objects in the camera image.

4.1.2 Single Camera Stereo Vision

Our second sensor innovation was a single-camera stereo vision system. Stereo vision is attractive for obstacle detection but is complex and expensive. We have developed a unique stereo vision system using a single camera. The vertical field of view is split by two stacked prisms (see diagram below). The two resulting fields of view are redirected back in front of the robot, thus giving Ball-thazar two parallel views of the world in front of him. The software then takes a corresponding slice from both views, identifies and matches obstacles, and is then able to perceive the distance to each obstacle.

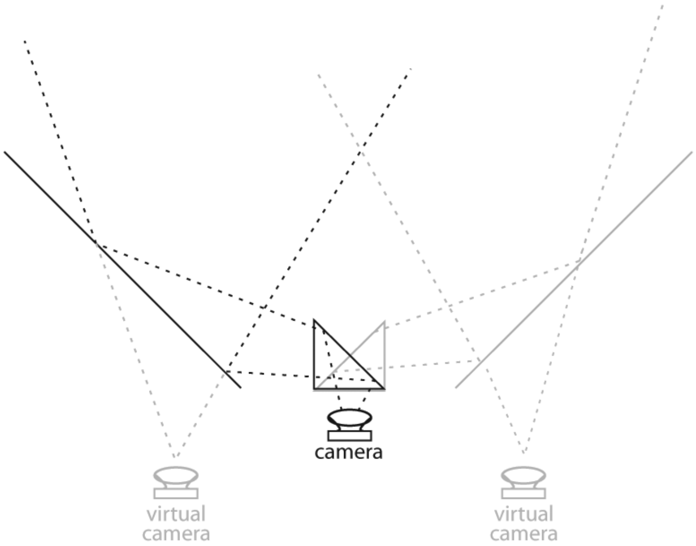


Figure 2 - Single-camera stereo

4.1.3 Infrared Rangefinder

As we gained experience with the weaknesses of sonar, we began researching IR as an alternative sensor system. The system is built on the Sharp 2Y3A003 IR range finder module. The module transmits a frequency modulated IR beam, and triangulates on the reflected light to calculate the range. Our tests show the system to be accurate to within inches and to have good sunlight capability, a traditional weakness of IR systems. The sensor outputs an analog signal which is digitized by a PIC18F1220 microcontroller which, in turn, sends the data to the main processor via USB. The IR system uses a servo to scan the sensor over a 150º field of view 4 times per second thus maintaining current readings. As with sonar, virtual obstacles are added to the robot's primary camera image.

4.2 Software Innovations

Our software was dramatically revised from previous years in five significant ways.

4.2.1 Port to Windows XP

The first change consisted of porting the existing code to Windows XP. The change mandated rewriting the serial port and video capture code using Windows APIs. It also required the code to be restructured around a Windows GUI event pump rather than a linear thread of execution.

4.2.2 Rewrite in Python

The second phase involved a complete rewrite of the majority of the code using the Python scripting language. Ball-thazar’s framework, sensors, and drive strategies were rewritten in Python to allow for rapid development and easy deployment, while the video capture and image processing algorithms continued to use the C language for efficiency.

4.2.3 Odometry

An odometry module was also added to the system to allow Ball-thazar to navigate in the absence of current GPS readings. Dead reckoning is achieved by combining the histories of compass bearing, wheel movement, and previous GPS readings.

4.2.4 Two-Computer Control

We continued implementation of the two-computer "horse and rider" control system, which was planned for Adam II but never fully implemented. The robot itself has primitive reactive navigation capability via the on-board embedded PC, while a rider notebook computer provides higher level deliberative navigation capability.

4.2.5 Course learning

Given that the same IGVC course is driven multiple times, the robot ought to be able to learn from previous course runs as any human driver would. A course map is constructed in memory, based on GPS, odometry, and obstacle detection inputs. This map is used in future runs of the same course to bias arc selection in reactive mode and as the basis for preferred direction changes in deliberative mode.

4.3 Hardware and Electronics Innovations

Building upon our software enhancements, we introduced two innovations to the robot’s hardware and electronics.

4.3.1 Replace Caster Wheel With Ball

We introduced a replacement for the rear caster wheel by building a unique ball suspension that provides smoother direction changes. Robots riding atop a ball is a novel new paradigm pioneered by Carnegie Mellon's Ballbot. Unlike Ballbot, our vehicle does not balance atop the ball but uses it as one point in a three-point suspension. In addition to smooth direction changes, we expect the ball to have superior performance on sand and other soft surfaces. The ball rides on three inverted double-sealed ball transfers. Prototyping was in Catia-CAD.

4.3.2 Wireless capability

In the past, on on-board keyboard and monitor was required for human interface to the on-board computer. We created remote access to Ball-thazar using a Lantronix WiPort embedded wireless module. This module is designed for embedded wireless and is compact and DC powered. This addition allows the team to control the robot remotely during testing, and is able to be disabled during competition as required by the rules.

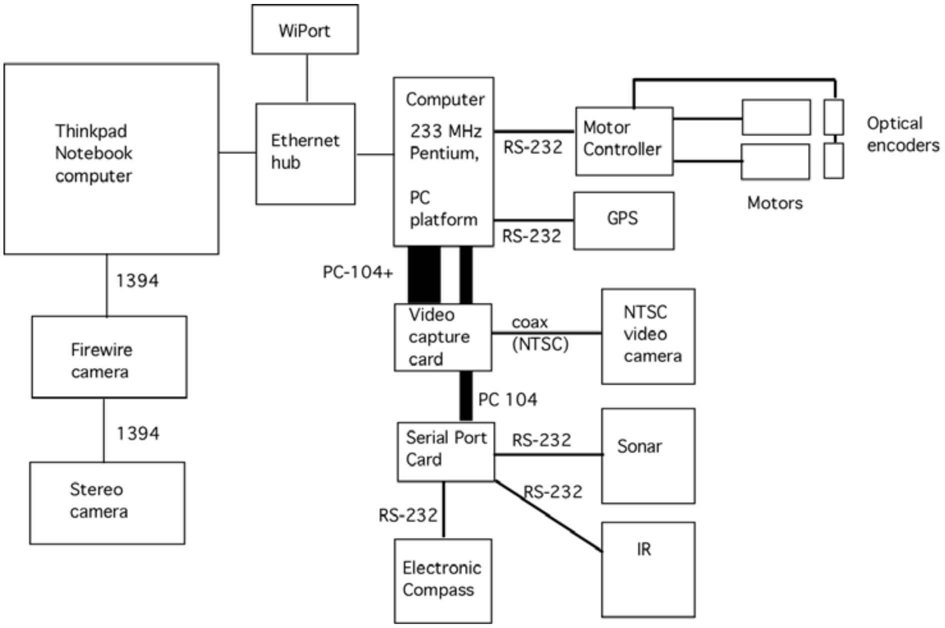


Figure 3 - System Block Diagram

5. ROBOT BODY

The robot's main frame is constructed from extruded aluminum T-slot material which was supplied by 80/20, Inc. This material provides ease of assembly as well as strength. The batteries and motor controller are mounted on a large piece of aluminum sheet metal. Also, the rear suspension is mounted on aluminum angle iron anchored on either side of the robot frame. Our front wheels and motors are fastened with custom mounting plates. Differential speed control of the front wheels provides drive and steering. The rear ball simply coasts.

The majority of the electronic equipment is housed in a rebuilt electronics case. Padding is used to prevent vibration problems with some of the heavier equipment, such as the batteries. A tower was built on top to provide better camera placement. The devices mounted on this tower include a camera, compass, GPS, and the wireless e-stop antenna.

6. ELECTRICAL SYSTEM

The electrical system of our robot consists of the following components: two motors, a motor controller, three batteries, a single board computer, emergency stop system, GPS receiver, compass, voltmeters, amp meters, video cameras, sonar module, infrared module, and wireless card. The single board computer is the central component of the whole electrical system. It coordinates the electrical devices required for the operation of the vehicle. Most of the electrical components are located in a main electronics case. A benefit of this is the overall neatness and accessibility of the electrical box. Two amp meters are located on the side of the robot to measure the load on the 12 and 24 volt batteries. Alongside the "ON" switch for the robot are two LCD displays which show the voltages across both the 12 and 24 volt battery packs. This enables us to easily determine the status of the battery packs during use or charging.

6.1 Power Supply

Three 12-volt gel-cell batteries (PS12180) are used to power the robot. Two of these batteries are connected in series to supply 24 volts to the two motors and the motor controller. The remaining 12 volt battery is used to power the computer and other electronic devices that are on the robot through a high-efficiency +5V DC-DC converter.

6.2 Embedded PC

The robot is controlled by an Ampro LittleBoard P5V PC-104 single board computer which comes equipped with an Intel Mobile Tillamook Pentium 266 CPU and 256MB of memory. This industrial-grade PC can run fanless and diskless and is temperature and vibration rated. The LittleBoard computer has a daughterboard that provides a PS/2 keyboard, mouse input connectors and two USB ports. The LittleBoard itself includes PC-104+ expansion slots, which are populated with the video capture card and a serial port expansion card.

6.3 Motors and Motor Controller

The motors for this robot are modified wheel chair motors with attached gearboxes (NPC-R82) from NPC robotics. For compatibility and ease of interfacing, the motor controller for the robot (NPCAX2550) is also from NPC robotics. This controller manages motor speed using PWM techniques and supports a closed-loop feedback through optical encoders. The controller communicates with a host computer via an RS-232 interface.

6.4 Video Cameras

A NTSC color video camera and an RTD Embedded technologies CM7326ER PC-104 video capture board are used for imaging and obstacle detection. The video capture board handles both the PAL and NTSC video standards with variable frame rates and capture resolutions. It also supports multiple video channels. In addition to the sonar mounted on the front of the robot, a hue-based obstacle detection system is also used. It is crucially dependent upon accurate color information in the video signal. A second iFire Firewire camera attaches to the rider computer.

6.5 GPS Receiver

The GPS receiver that we are using is the Garmin GPS18 LVC. It runs off of a required power supply of 4.0-5.5v DC @ 60mA. The GPS18 is capable of receiving a Differential GPS signal, and is WAAS enabled. While receiving a Standard Positioning Signal (SPS), we can obtain accuracies of <15m typical. With a differential signal (DGPS/WAAS) available, we can get accuracies of <3m typical. The receiver communicates to our computer board through a serial connection.

6.6 Digital Compass

The PNI-TCM2 is a very sophisticated electronic compass. Not only does this compass display heading within a tenth of a degree, but it also has a two-axis sensor to detect pitch and roll.

6.6 Wiport and network hub

An Ethernet hub connects the WiPort wireless access module, the on-board embedded computer, and the "rider" laptop computer.

6.6 System integration

The embedded CPU integrates all of the various electrical subsystems and performs the low-level communication and control. Standard busses and communication ports provide easy interfacing and readily available software drivers. Our unique two-computer approach allows the notebook computer to be dedicated to high level navigation, and standard Ethernet provides easy integration of the two computers and the WiPort.

7. SOFTWARE

The software runs in four modes: Autonomous, Navigation, Calibrate, and Follow.

7.1 Autonomous Mode

Autonomous mode is designed to guide the robot through the IGVC autonomous challenge. Processing begins with capture of an image from the main camera. This image is processed to detect red/orange/white obstacles based on a hue/brightness threshold scheme.

To this image are added virtual obstacles seen by the three obstacle detection systems (stereo camera, sonar, IR). This process of adding virtual objects to the camera image is our unique *sensor data integration approach*. Note that these three subsystems can be enabled or disabled in software, allowing us to adapt to the realities of the actual IGVC course. Thus our *obstacle detection approach* consists of four independent sensor systems.

The image is also pre-processed to detect lines using a Hough transform. These lines are then extend them to fill in any gaps, allowing the robot to perform in the presence of dashed lines. Thus our *lane-following approach* consists of extending lines to fill gaps and then treating them as obstacles.

7.1.1 Reactive algorithm

From this master image including obstacles, virtual obstacles, and lines, various arc paths are evaluated and the one that has the longest free path is the one chosen to drive, with a bias towards straight paths if a number of free paths are available. This *obstacle avoidance approach* can be described as "drive towards the free space" and works well when the path is relatively free of obstacles.

7.1.2 Deliberative algorithm

If no adequate paths are found, the robot drops into deliberative mode. Deliberate mode executes changes of direction and/or backs up to search for alternate routes, based on a global map. When a free path is detected, the software returns to the reactive algorithm. This is essentially a random search for a free path but constrained by the global map to avoid complete reversal of direction and traversal of the course in the wrong direction.

7.2 Navigation Mode

Navigation mode is designed for the IGVC navigation course. A series of waypoints are read from an initialization file when the program begins. The execute method reads the current estimated pose (from filtered GPS, compass, and odometry data) and determines which direction it must drive to reach the next waypoint. When the distance between the current location and the desired waypoint is below a specified threshold, the robot assumes it has reached the target and will advance to the next waypoint.

While driving, an image is captured and processed by the reactive mode code described under "autonomous mode" above. If this analysis indicates that the path to the waypoint is clear, the robot drives towards the waypoint. If the way is not clear, the reactive algorithm is used to drive around the obstacles, at which point the path to the waypoint is recomputed and the process begins again.

These steps constitute our *waypoint navigation approach*. Because the obstacles are relatively sparse in the navigation challenge, the deliberative algorithm is not used although it could be activated if necessary. However, the map building approach is optimized for the autonomous challenge.

7.3 Follow Mode

As an alternative to joystick manual control, follow mode uses the on-board camera to locate and drive towards a predefined optical target. This allows the robot to be "led around" by a human operator carrying the optical target. Loss of target in the image or distance below a predefined threshold causes the robot to stop, for safety.

7.4 Calibrate Mode

This mode was designed for testing purposes. Numerous test images and diagnostic information are displayed, including most significantly a cross-hair that displays RGB values from the main camera for in-the-field image analysis and algorithm tuning.

8. PRACTICALITY

8.1 Safety

Safety features include a long range wireless e-stop system with the receiver mounted to the top of the camera tower to increase range. The robot can be stopped by wireless link from well over 50 feet away. Along with the wireless e-stop, a red push button is located on the rear of the robot; a simple touch will immediately shut down the robot. The robot motor controller contains the needed integrated e-stop capability and provides the braking necessary to stop the robot within the required stopping distance. In addition to the e-stop systems, foam bumpers on the front and rear of the robot provide an additional measure of safety for bystanders and obstacles. Main power supplies are fused. Batteries are sealed lead-acid with valve-regulated sealed construction and therefore have limited emissions and may be inverted without danger. A keyed main power-up switch prevents non-qualified personnel from powering the robot.

8.2 Reliability and Durability

The motherboard is temperature resistant, vibration tested, and reliability rated by the manufacturer. Rugged off-road tires are highly puncture resistant and provide good performance on rough terrain. Batteries are sealed for long life. Major components including the processor and batteries are mounted with shock-absorbing materials. Easily-removable, completely enclosing body panels provide protection from the elements. The motor controller includes both overcurrent and overheating protection.

8.3 Efficiency

A high efficiency DC-DC converter provides the main 5V power supply. A high-efficiency motherboard includes a low-power Tillamook CPU, contains no fans, has no rotating media, and only 5V power requirements. Unused systems are powered down for further power savings. The system is run without a monitor except for setup and testing. The motor controller includes high-efficiency multiple-MOSFET power output stages and is rated at 120A per channel with no cooling fan. With a weight of 3.8 lb, it is the lightest in its class and perfect for a robot like Ball-thazar.

8.4 Performance

Ball-thazar weighs approximately 180 lbs. Weight distribution is an ideal 33% per wheel. Center of mass is a low 12 inches. Batteries are rated at 18 Ah, providing almost one hour of constant drive time at the maximum load, or four hours on typical flat terrain. The electronics package of this robot draws a calculated 2.67 amps of current. The 18 Ah batteries should provide a run time of approximately 6.7 hours.

We plan to run our software at a 1/15 second frame rate in reactive mode, providing a reaction time of 0.067 seconds. At 5 mph, the robot moves approximately .149 meters between frames. The camera views 9 meters ahead, providing adequate time to respond to obstacles even at full speed. Deliberative mode is expected to require a much slower frame rate, so we plan to operate at speeds much slower than 5 mph when in this mode.

Table 1 - Performance Details

Parameter	**Prediction**	**Actual**
Top Speed	8.38 mph	5 mph*
Weight	185 lb	190 lb
Power (level)	90 watts	96 watts
Power (15% slope)	351 watts	360 watts
Current (24v battery pack)	3.76 amps	4 amps
Current (12v battery)	2.67 amps	1.4 amps
Weight Distribution (front/rear)	70/30	67/33**
Battery Life (level Ground)	4.5hr	>2 hours***
Reaction time	67ms	143 ms****
Distance of Obstacle Detection	9m	8.5m
Waypoint accuracy	1.5 m	Varies with GPS conditions

* The drive train is capable of >8 mph on level ground but the closed loop speed controller limits the maximum speed to 5 mph.
** With two wheels on the front, this represents a distribution of almost exactly 33% per wheel.
*** Not tested all the way to dead batteries
**** Hardware is capable of 67ms frame period (15 Hz) but software has only achieved 7 Hz so far.

8.5 Bill of Materials

Table 2 - Cost of Components

Components	Quantity	Retail Price	Cost to Team
PS12180 18Ah batteries	3	$144	$144
Ampo Littleboard Pv5 single board computer	1	$995	$995
NPC-R82 motors	2	$570	$570
Hubs	2	$40	$40
NPC-AX2550 motor controller	1	$94	$94
Wheels and tires	2	$495	$495
GPS receiver	1	$89	$89
PNI-TCM2-20 Digital Compass	1	$699	$0
PC-104 4-port serial board	1	$149	$19
T-slot and angle bracket frame	N/A	$180	$180
Emergency stop system components	N/A	$70	$70
CM7326ER PC/104 video capture card	1	$455	$455
NTSC video camera	1	$15	$15
Firewire camera	1	$120	$120
Thinkpad computer	1	$1,500	$0
Stereo Camera (research project)	1	N/A	$220
IR System	1	N/A	$100
WiPORT wireless module	1	$199	$99
Ball mount	1	$350	$350
Miscellaneous	N/A	$170	$170
Total		**$6,335**	**$4,316**

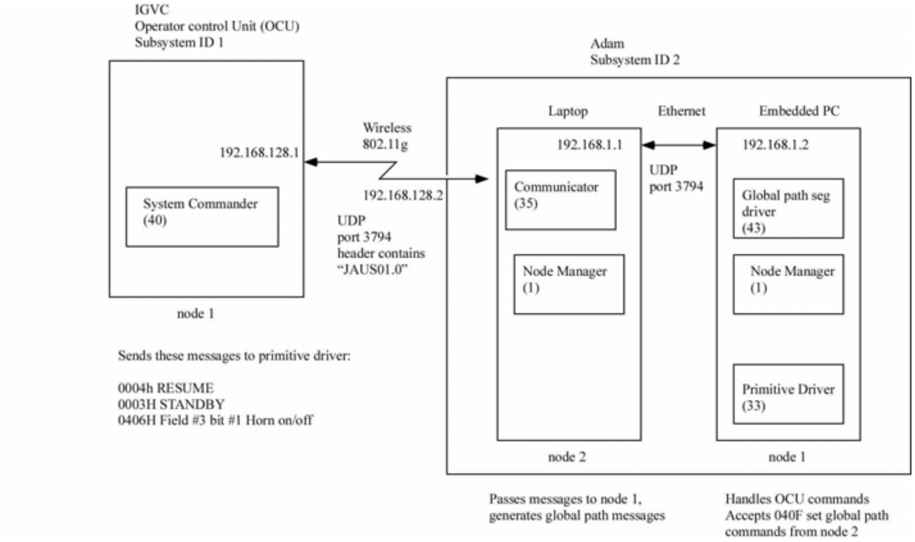
8. JAUS

8.1 Learning Process

JAUS was researched by the 2006 team and software developed to implement it. The JAUS specifications, the IGVC JAUS guidelines, and last year's code and documentation were all used by this year's team to gain an understanding of JAUS.

8.2 Integration

A block diagram of our JAUS implementation is below. Note that our robot is a subsystem consisting of two computers. These two computers use JAUS messages to communicate. The wireless WiPort used for development and testing is not shown in this picture, as it is disabled during runs and is not part of the JAUS system.



8.3 Challenges

The 2006 team failed to complete the JAUS challenge due to a software bug based on a misunderstanding of the message structure. A significant obstacle to JAUS implementation is the lack of an easily accessible JAUS message generation system to test against. We think we understand and can fix this bug, but again are unable to test completely until we arrive at IGVC.

Fortunately, this year's IGVC interface control document was released earlier and is more complete, making the learning task easier for this year's team.

Certification of design work performed by the Bob Jones University design team

I, Dr. Bill Lovegrove, Professor of Electrical Engineering at Bob Jones University, certify that the members of this engineering design have done significant engineering design work on the robot that is equivalent to the work that is awarded credit in a senior design course.

Signed:

Date: