Bob Jones University's

Fortunatus

Richard Armstrong, Lucas Buchmoyer, Anthony Ceder, Benjamin Cole, Benjamin Hancock, Sarah Ishida, Ryan Kawakami, James Moreno, Seth Noe, John Pobuk, James Wagner

Faculty Statement:
The design and engineering of this vehicle by the current student team has been significant, representing senior design credit for this project for some of the team members.

____

William P. Lovegrove, PhD.
Chairman, Dept. of Physics and Engineering
Bob Jones University

TABLE OF CONTENTS

I. Introduction

II. Obstacle Avoidance Innovation

III. Design Process

- III.1 Strategy
- III.2 Safety
    - III.2a Durability
    - III.2b Reliability

IV. Software

- IV.1 Strategy
- IV.2 Signal Processing
- IV.3 Control Decisions
    - IV.3a Line Detection and Path Following
    - IV.3b Obstacle Avoidance
    - IV.3c Waypoint Navigation

V. Hardware

- V.1 Electronics
- V.2 Electrical System
- V.3 Computer
- V.4 Body

VI. Robot Capabilities

- VI.1 Speed
- VI.2 Slopes
- VI.3 Battery Life
- VI.4 Reaction Time
- VI.5 Navigation Accuracy

VII. The Team

VIII. Costs

IX. Conclusion

FORTUNATUS

I. INTRODUCTION

This year the Bob Jones University robotics team has implemented a design philosophy which differs from that of most of its competition. Of the two approaches to robot navigation, our team opted for the reactive approach, in which the robot merely reacts to its present situation, instead of the more popular deliberative approach, in which the robot constructs a map of its environment. With the exception of retaining a memory of the course direction, our robot Fortunatus possesses no history and no memory—we are exploring the limits of purely reactive navigation.

II. OBSTACLE AVOIDANCE INNOVATION

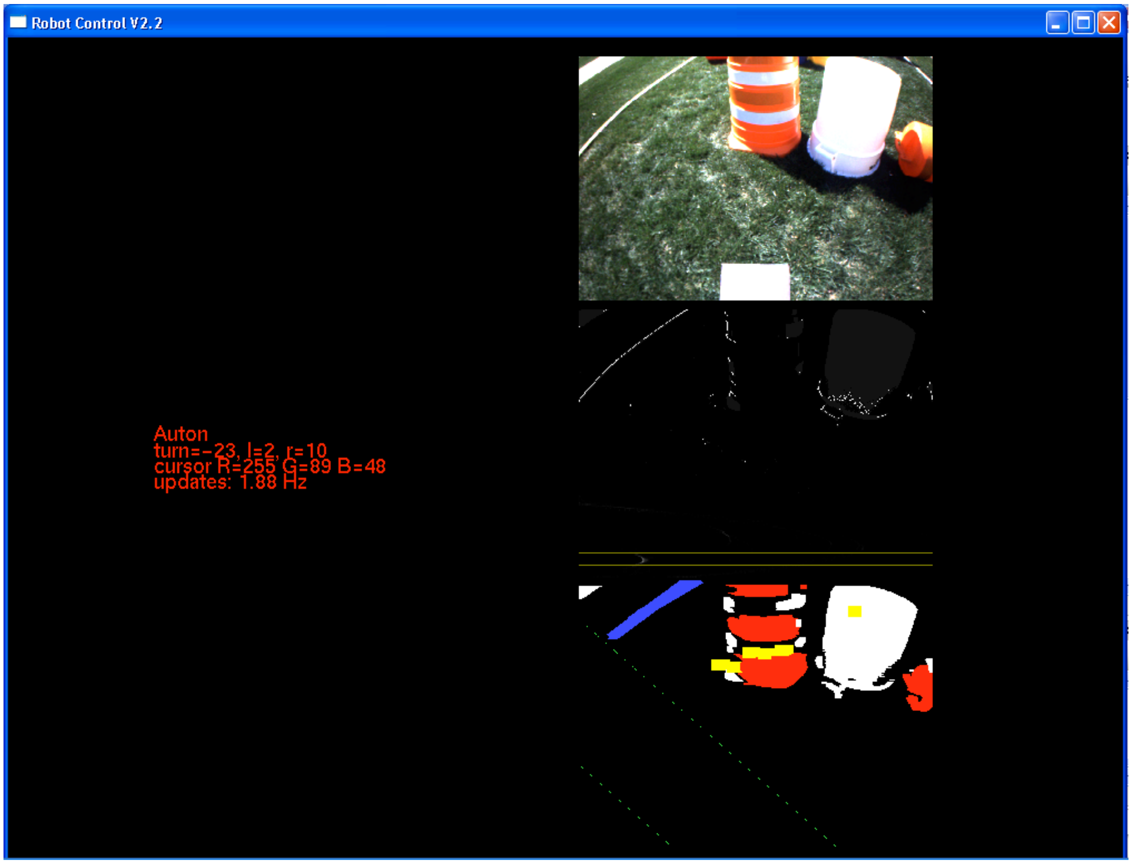
Consistent with our reactive approach design philosophy, Fortunatus's innovative and enhanced obstacle detection program enables the robot to use texture analysis to identify obstacles. The program from last year's team looked at the basic color spectrum of red, green, and blue and used the RBG values to avoid any particular color using rules written for specific colors. Because this program resulted in somewhat troublesome code, we now program Fortunatus to avoid any surfaces that have a color variation less than the texture of grass. Our change requires the computer to search for fewer conditions, allowing it to process information faster.

Fortunatus calculates hue using the camera's RGB (red, green, blue) values of individual pixels. The team used standard formulas to convert RGB values to HSL (hue, saturation, lightness) and HSV (hue, saturation, value) for the following analysis.

Fortunatus perceives varying textures by looking for changes in the hues of its environment. How often the hues change determines the texture the robot detects. The computer receives input from Fortunatus's digital camera, and the program looks at textures to determine which parts of the camera's picture are obstacles and which parts are safe to drive on. As the program scrutinizes the picture from the digital camera, the code searches for hue changes from pixel to pixel. If the difference in hues is above a threshold, the pixel contributes to the number of hue changes in an array of regions. Because differences among hue calculations are numerous on grass (due to shadows, varying shades of grass, and objects on the grass, such as leaves), the program opens only regions that contain enough hue variation, allowing Fortunatus to drive only on grassy textures. In contrast to grass, which consists of differing hues, the texture of obstacles (such as cones and barrels) consists of uniform hues. If the differences of a region's hues are not great enough, the program fills in that region and does not allow the robot to drive there because it assumes the region contains an obstacle. Fortunatus constantly performs this obstacle detection process as it navigates.

A programmer can access a configuration file to adjust the obstacle detection program's sensitivity to differences in hues. The programmer can increase or decrease the amount of hue variation needed for Fortunatus to decide what has a grassy texture and what has the smooth texture of an obstacle. Access to the configuration file makes Fortunatus more versatile. For example, if we find that obstacles in one environment are smoother than the obstacles in a previous environment, we don't have to change the entire obstacle detection program to enable Fortunatus to navigate in the new environment. Instead, we can simply alter the numbers in the configuration file.

Although Fortunatus primarily relies on texture analysis and could navigate around obstacles using only its camera, infrared sensors are a secondary input verifying that obstacles are present. In particular, infrared allows Fortunatus to detect textured obstacles that would otherwise fool the texture analysis program.



```
Auton
turn=-23, l=2, r=10
cursor R=255 G=89 B=48
updates: 1.88 Hz
```

Figure 1—A screenshot from Fortunatus's computer depicting the image captured by the camera (top) and an image of what Fortunatus's obstacle detection software detects (bottom)

III. DESIGN PROCESS

III.1 Strategy

This year the team sought to improve on a previously designed version of Fortunatus, which came in twelfth place in the 2010 IGVC Autonomous Challenge. The team began designing this year's robot by looking at the weaknesses of the 2010 robot and then finding ways to remedy these flaws.

This year's team made changes to both the hardware and software of last year's robot. We added many new features and improved the existing code. We also changed the robot from rear-wheel drive to front-wheel drive and designed a new body to accommodate this change.

Each first-semester team member was assigned a role in the project. The tasks were broken up into eight fields in order to maximize both lab and personal time spent on the robot. Four different people worked on the programming. One person was the overall software manager, and three additional team members worked separately on the Navigation code, the Autonomous code, and the JAUS programming. Despite work done on the JAUS programming, however, we decided not to attempt the optional JAUS competition this year.

Our design process consisted of a sequence of intermediate goals to be attained throughout the semester. First, we tested all systems to make sure that the robot still functioned according to the basic requirements that last year's team had met. Another goal in the sequence required the robot to successfully maneuver through a series of switchbacks.

Next, the team brainstormed for ideas for major changes and implemented those changes. Because the team needed to make significant changes to the robot's coding in order for it to qualify for the competition, we decided to make Fortunatus front-wheel drive. Making Fortunatus front-wheel driven was obviously a significant mechanical change because turning the robot around required body modifications and a new wiring harness. But this mechanical change also required a major change in coding because the current robot would move backwards if last year's code were still used. Without a change in coding, the camera, computer, and other components would face the wrong way when the robot made an attempt to move forward.

Last year's robot had a forward caster wheel, but this year we decided to place the two motor-driven wheels in the front. Turning the robot around improves it by giving it a narrower back end. When it had two wheels in the back, the rear of the robot would touch obstacles during sharp turns. But now that Fortunatus has a single wheel in the back, it is less likely to touch obstacles during these turns.

Fortunatus's improved turning ability is also suited to its improved ability to recognize when it is stuck at a dead-end. This year's team implemented logic in the code, allowing Fortunatus to swivel and turn around in place. The robot can now maneuver out from obstacles that are blocking it.

Except for the time we dismantled the robot to change it to front-wheel drive, Fortunatus has been fully operational the entire time we worked on it, allowing us to implement new ideas directly to the robot as we progressed on a certain aspect of the design. All of the systems were fully integrated during the entire design process.

III.2 Safety

As in the previous year, this year Fortunatus is equipped with the mandatory wired and wireless e-stops. We ran into a problem with the wireless e-stop, however, when we discovered that the range of the wireless wasn't long enough. To solve this issue, we modified the transmitter antenna.

In addition to the e-stops, this year's robot features improved fuses, a definite safety upgrade. Last year's robot had fuses that were incorrectly positioned, resulting in a fire. Thankfully, the fire was small and was put out quickly enough to save everything except a small wire. The problem was caused because Fortunatus needed a fuse immediately after the wire that connected the battery to the electrical system. Now that this problem has been fixed, the fuse will blow when a problem occurs instead of starting a fire.

A final hardware safety feature is Fortunatus's sealed lead acid batteries. Unlike car batteries, Fortunatus's batteries will not spill when tipped over.

In addition to hardware safety considerations, Fortunatus detects obstacles at ample distance to avoid running into them. The robot reacts to an obstacle at a minimum of 9.8 ft., a distance which gives the robot enough time and distance to turn or stop instead of crashing into the obstacle.

Our robot is capable of driving at 10 mph but the image processing and vehicle dynamics are difficult at this speed.

III.2a Durability. Fortunatus's frame is constructed a 8020 T-slot aluminum frame and heavy gauge aluminum sheet metal.

III.2b Reliability. Fortunatus's durable frame also contributes to its reliability. In addition to the frame, Fortunatus has heavy duty motors and a heavy gear box. The body and motors of our robot are not likely to wear out or burn out from insufficient strength.

IV. SOFTWARE

IV.1 Strategy

Fortunatus's software was written in C++ using the Visual Studio development environment. Other than device drivers and the open GL graphics library, all of the software was written by the team.

IV.2 Signal Processing

Fortunatus's signal processing consists of a custom interface to the infrared sensors that last year's team designed. The remaining robot systems are all integrated and require no signal processing. This robot relies heavily on the camera and image processing, so the image processing is a large part of the design process.

IV.3 Control Decisions

Fortunatus has two front wheels that are driven by two separate DC motors. By simultaneously adjusting the direction and speed of both motors, the robot can turn left or right and can turn quickly or slowly.

Fortunatus uses its obstacle detection code to determine when to turn. The program analyzes the current image and identifies obstacles. The algorithm determines either a desired steering direction or an indication that the robot is stuck and needs to pivot and go in a new direction. The code calls a subroutine to deal with pivoting or driving in reverse to try to remove the robot from the tight space. The two drives wheels are commanded to drive at the desired speed to accomplish the chosen path.

IV.3a Line Detection and Path Following. The robot progresses through several functions for detecting solid lines. The program looks for any near-white pixels. The pixels are grouped together with similar pixels, and line thickness tests are passed to see which pixel groups are wide enough to be considered a line. All of the pixels that are determined to be both white and part of a line are analyzed using the standard Hough Transform, which indentifies the lines in the current image. When the robot encounters dashed lines, the software identifies the line fragment and then fills the gap by extending the line forward in the image**.**

IV.3b Obstacle Avoidance. As explained earlier in the Obstacle Avoidance Innovation section, Fortunatus avoids obstacles by analyzing the textures in its environment. Fortunatus understands that grassy textures are safe to drive on and that smoothly textured surfaces should be avoided. Fortunatus is also equipped with infrared sensors as a secondary means of obstacle avoidance. These sensors particularly help to detect textured obstacles.

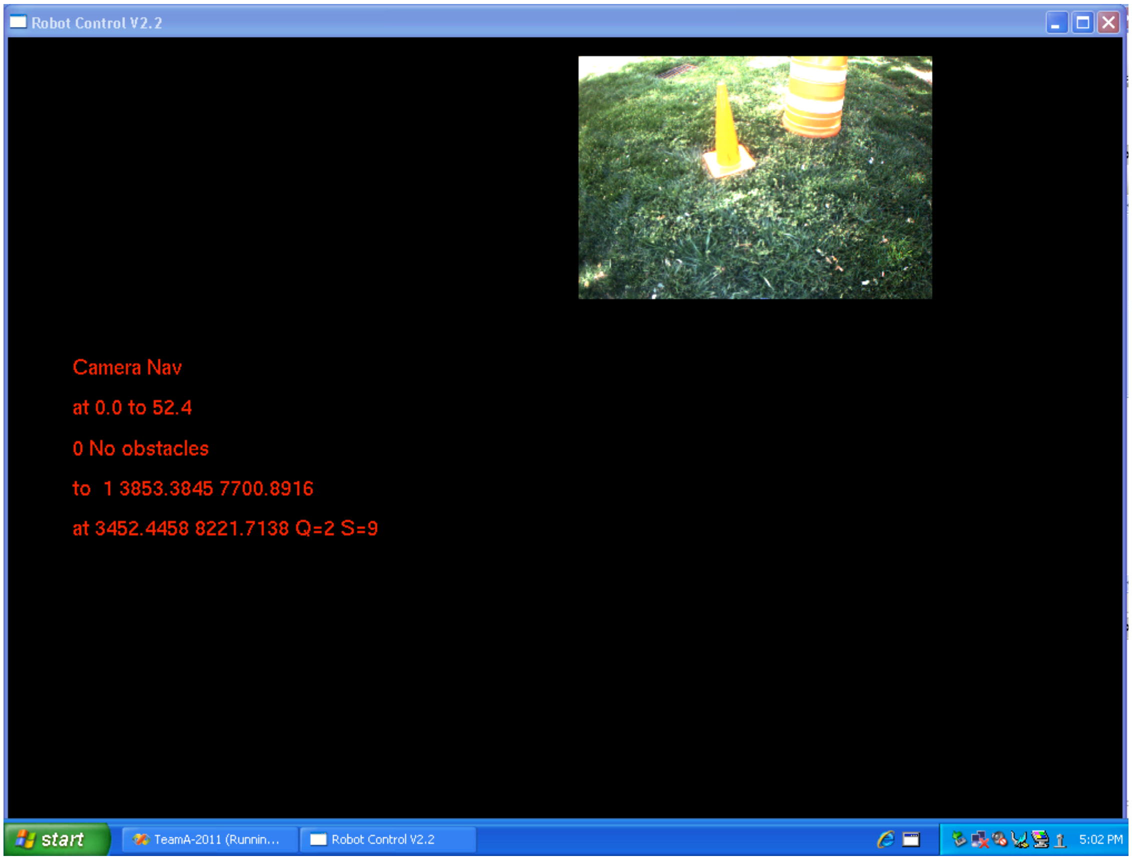
Like our plan for general obstacle avoidance, the plan for dealing with complex obstacles (such as switchbacks, center islands, dead ends, traps, and potholes) is related to the reactive approach design philosophy. Fortunatus searches for any available free path, enabling the robot to find a path through complex obstacles but occasionally allowing the robot to turn around and travel the course backwards. Although our robot never gets stuck, it may continue to turn around until it's in the wrong direction. To prevent Fortunatus from going in the wrong direction, the robot retains a memory of the course direction and compares this direction to its current compass heading.

IV.3c Waypoint Navigation. Fortunatus's mode of navigation conforms to our reactive approach design philosophy. While robots following the deliberative approach reach waypoints using a planned path, Fortunatus's means of reaching waypoints consists of steps which are repeated for each waypoint destination.

First, Fortunatus's navigation program uses the GPS received to obtain the current location. The current location is compared to the first target waypoint.

After Fortunatus compares its heading and orientation to that of the desired heading, Fortunatus steers in a direction that will align it with the first waypoint. The radius of turn is proportional to the discrepancy in compass heading. If no obstacles are encountered, Fortunatus visits the waypoints one by one in order, repeating the steps of orienting itself to the waypoint and then turning and driving to each waypoint.

However, if at any time Fortunatus detects an obstacle in its path, the robot navigates around the obstacle even though doing so diverts the robot from pursuing the direction of the target. Fortunatus enters avoidance mode and ignores the GPS until it once again detects clear ground, at which time it resumes steering toward to the waypoint. This reactive approach does not result in optimum paths but is simple and quite robust.



```
Camera Nav
at 0.0 to 52.4
0 No obstacles
to  1 3853.3845 7700.8916
at 3452.4458 8221.7138 Q=2 S=9
```

Figure 2— A screenshot from Fortunatus's computer while the robot is operating in navigation mode

V. HARDWARE

Fortunatus's frame was made using 8020 T-Slot aluminum. The base of the machine houses an enclosed space, the RoboteQ DC motor controller, the power converter, and all three of the lead-acid batteries used to power the vehicle. An IBM ThinkPad is used to control the robot.

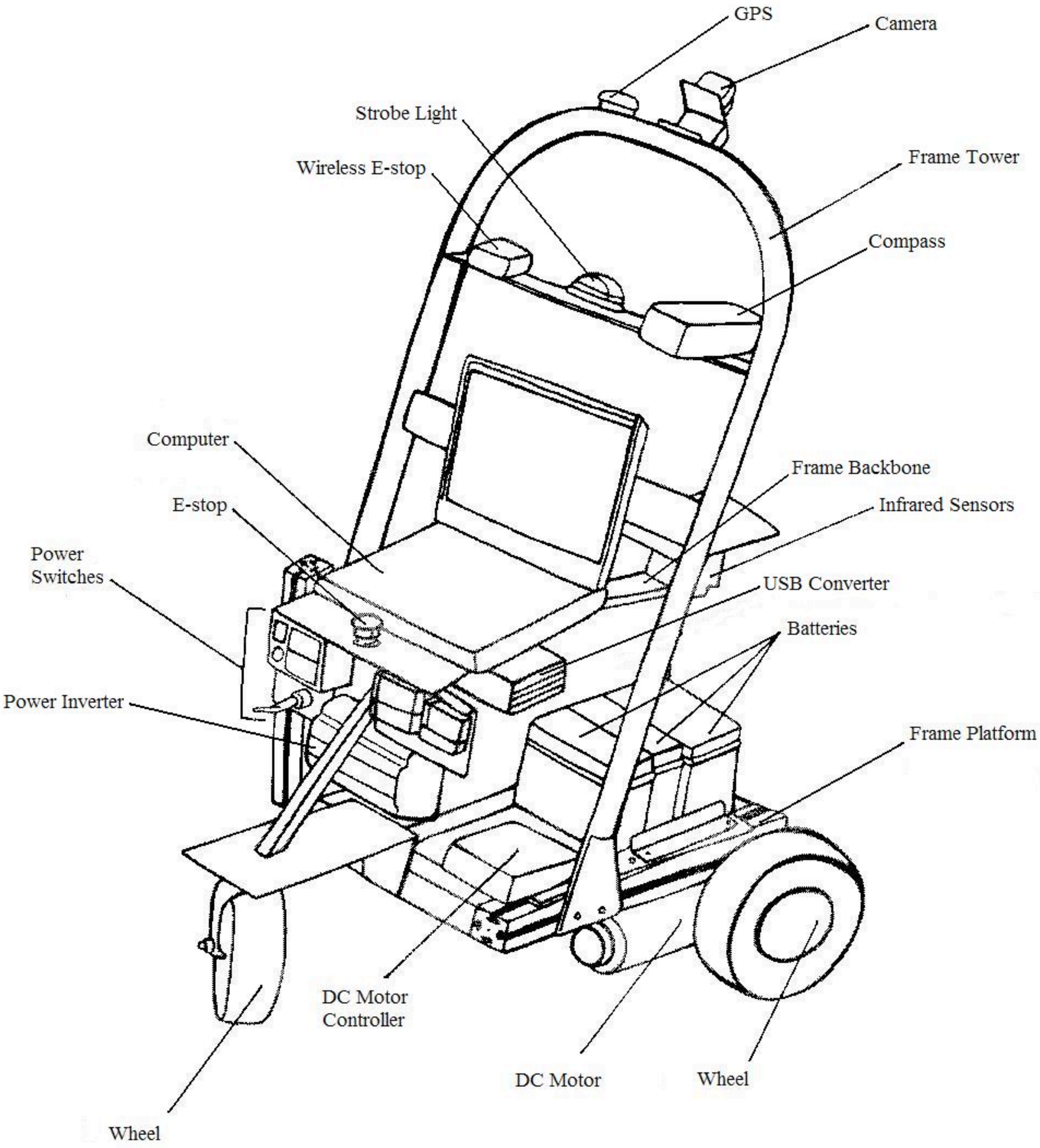


Figure 3—Line drawing of Fortunatus identifying key components of its hardware

V.1 Electronics

The electronics system is composed of the computer through which all electronic signals are processed, the Point Grey camera, the Ublox GPS antenna and receiver, the PNI digital compass separated from any electrical interference from the other equipment, the orange strobe light, the infrared sensors located high enough off the ground so that a possible incline is not detected as an obstacle, and the RoboteQ motor controller behind the batteries. All of these components were present on the robot before this year, but this year's team rewired large parts of the robot and repositioned wireless e-stop antenna where it could receive stronger signals. The electrons are powered by a large 12V sealed lead acid battery. Note that to avoid interference the drive motors are powered by a separate battery.

V.2 Electrical System

The electrical system contains two DC motors to power the front wheels, one 24V drive motor battery pack consisting of two 12V batteries in series, the RoboteQ motor controller, a power converter, and a power supply for the computer.

The DC motors behind each front wheel are Fortunatus's only actuators. These two motors can turn the wheels at different speeds, an advantage allowing both front wheels to both drive and steer, unlike other robots that have one motor to turn the back wheels and another motor to steer the front wheels. Fortunatus's simplicity enhances its durability and reliability because there is less equipment in this area with the potential to malfunction.

V.3 Computer

Fortunatus uses an IBM ThinkPad to operate all software and additional support programs. The ThinkPad has a Pentium 4 2.6 GHz processor and 512 MB of RAM. The team considered implementing a second computer to speed up the processes, but the added weight of a second computer and the space necessary to accommodate the computer did not make it worthwhile. Also, the battery life would be significantly shorter with a second computer on board.

V.4 Body

The body is constructed of transparent Plexiglas. The entire body can be taken off the robot, allowing easy access to the interior components. We enclosed the lower level of the robot completely so that the batteries, motor controller, and power supply are shielded from the elements. The computer sits on top of the rest of the robot body with a separate Plexiglas rain shield.

VI. ROBOT CAPABILITIES

VI.1 Speed

Fortunatus has a measured top speed of approximately 8 mph, but as of April 2011 the robot was limited to 5 mph due to image processing and vehicle dynamics limitations.

VI.2 Slopes

Fortunatus requires a measured 360 watts when climbing a 15% slope, but the robot requires approximately 96 watts on flat ground. Predictions for the slope equaled 351 watts, and predictions for flat ground were 90 watts.

VI.3 Battery Life

The lifespan of the 12V battery is primarily determined by the computer's power needs, and at full load the computer draws a measured 5 amps (60 watts). The batteries are rated for 35 amp-hours, providing seven hours of operation, which is enough for a full day of competition.

Operating conditions and terrain heavily influence the length of the 24V battery's life. Yet even when experiencing hard driving more severe than what is required of it at the competition, the battery was tested to last for 45 minutes, which is enough battery life to last through the minimum of nine 5-minute runs during a full day of competition.

Even though Fortunatus's batteries will allow it to function for a full day of competition without having to be charged during the competition, if the batteries are charged between runs, battery life can be increased significantly.

VI.4 Reaction Time

Fortunatus has a frame rate measured at 3.5 Hz and a reaction time of 285 ms. At 10 mph this corresponds to a distance traveled of 4.2 ft. The robot usually begins to see an obstacle at 14 ft., but it will react to an obstacle at no less than 9.8 ft., which gives the robot a distance of 9.8 ft. and a time of 0.67s to turn or stop, requiring a deceleration of -3.3 ft/s^2 or approximately -0.1g, which the robot can easily perform.

At the maximum speed that we run the robot (approximately 5 mph), Fortunatus sees obstacles on its screen at a distance of 10.5 ft. At this speed, 4 ft. is the average distance between the robot and the obstacle when the robot reacts and adjusts to avoid the obstacle. Preventing the robot from traveling faster than 5 mph keeps it from getting too close to the obstacle before it can react or from making an attempt to halt abruptly, which could result in its tipping forward and falling over.

VI.5 Navigation Accuracy

The robot was predicted to get within 1.5 meters of the desired GPS waypoint, but the accuracy of arrival varies slightly with the weather conditions affecting the GPS. Most of the time, however, Fortunatus is able to come within 1.5 meters or less of the waypoint.

VII. THE TEAM

Name	Academic Classification	Major	Designated Responsibility	Hours Spent on Project
First Semester				
Lucas Buchmoyer	Senior	Electronics and Computer Technology	Navigation Specialist	76
Anthony Ceder	Senior	Electronics and Computer Technology	Project Manager	84
Benjamin Cole	Senior	Electrical Engineering	Author of first Fortunatus operating manual	83
Ryan Kawakami	Senior	Electronics and Computer Technology	Schedule Manager	51
Seth Noe	Senior	Electronics and Computer Technology	Design Report Specialist	66
James Wagner	Senior	Electrical Engineering	Software Manager	71
Second Semester				
Richard Armstrong	Freshman	Engineering		20
Sarah Ishida	Junior	Humanities	Technical Writer	25
James Moreno	Sophomore	Engineering		15
Both Semesters				
Benjamin Hancock	Senior	Electrical Engineering	JAUS Specialist	73
John Pobuk	Senior	Electrical Engineering	Autonomous Specialist	93
Total hours: 657				

VIII. COSTS

Item	Manufacturer	Model	Cost (US dollars)
Batteries	Power-Sonic	PS12180 18 Ah	144
USB 2.0 digital camera	Point Grey	Chameleon	
Compass	PNI Corporation	TCM2	
Computer	IBM	Thinkpad G40	1,000
DC motors	NPC	NPC-R82	570
Emergency stop system components			70
GPS receiver	U-blox	EVK-5H	89
Hubs	NPC		40
Infrared system	Custom by the team		100
Motor controller	RoboteQ	AX2550	94
Power inverter	XPower	700Plus	
T-slot and angle bracket frame			180
USB Converter	Digi	Edgeport 4	198
Wheels and tires	NPC		495
New caster wheel	Northern Tool		30
Total:			3,010

IX. CONCLUSION

Fortunatus is a quality robot based on a purely reactive design philosophy, and its performance validates its design. The robot was tested at the AUVSI demonstration in Washington DC, and it performed successfully, easily avoiding obstacles and effectively traversing the switchback on the course. The Bob Jones University robotics team has made Fortunatus a worthy competitor in the 2011 IGVC.