Bigfoot

Lawrence Technological University
IGVC 2015

Faculty Advisor Statement

I, Dr. CJ Chung and Jonathan Ruszala of the Department of Math and Computer Science at Lawrence Technological University, certify that the design and development on the Bigfoot platform by the individuals on the design team is significant and is either for-credit or equivalent to what might be awarded credit in a senior design course.

ChanJin Chung 5-15-2015

Signature Date

J Ruszala 5-15-2015

Signature Date

DESIGN REPORT FOR THE 2015 INTELLIGENT GROUND VEHICLE COMPETITION

Lawrence Technological University
Students: Gordon Stein, Ícaro Gargione, Jimmy Tanamati Soares, Fan Wei
Faculty: Dr. CJ Chung cchung@ltu.edu, Jonathan Ruszala jruszala@ltu.edu

INTRODUCTION

Lawrence Technological University's entry into the 23rd annual Intelligent Ground Vehicle Competition is Bigfoot. Built on top of the Clearpath Husky platform, Bigfoot is ready to take on the course with its rugged, capable design.

TEAM ORGANIZATION

Due to the limited team size, only one member was assigned a management position, and two sub-teams were created. One sub-team focused on mechanical engineering and electronics, while the other focused on the software for the robot. By creating software simulations of the real environment, the software sub-team was able to work at the same time as the engineering sub-team, without limiting the availability of the robot to the other group.

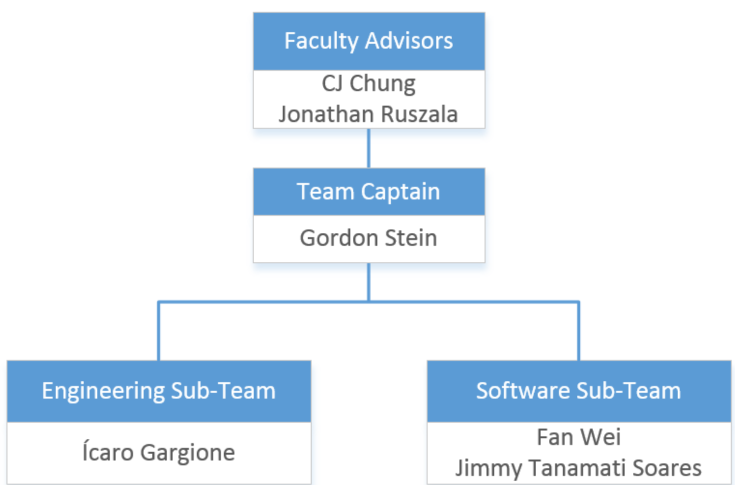

Figure 1. Bigfoot Team Organization

DESIGN PROCESS

Design Concepts and Goals

The main design goal for Bigfoot's hardware was to use low-cost, off the shelf components wherever possible. The reconfigurable top plate furthers this goal by making it simple to attach new parts to the robot. The sensors used were chosen because they were readily available and possible to replace if necessary.

The main design goal for the software was to greatly expand the use of simulations and software-based testing. In previous years, there was difficulty testing with the physical robot because software could not be run until the hardware was complete, and even when the hardware was complete, inclement weather or sunset would prevent the robot's operation outside. In addition, our university campus does not have an area where a proper practice course could be set up, so previous years' teams relied on testing with a small section of path. A simulation allows the team to test at any time and any location.

DESIGN INNOVATIONS

Redundant Motor Communications

Communications with the internal motor driver are received from two sources: the HP Z-Book laptop controlling the autonomous navigation for the robot, and a Raspberry Pi microcomputer receiving external commands. The selected source is controlled by a relay triggered by the Raspberry Pi, which transmits serial communications from the Raspberry Pi when open and from the laptop when closed.

Having two sources of commands for the motor controller allows the robot to be moved without the laptop present, so the robot is not immobilized if something disables the laptop. The Raspberry Pi is able to fully start up and send commands within a few seconds of receiving power, while the laptop could take several minutes depending on its current state.

Sliding Reconfigurable Top Panel

The top of the Clearpath Husky platform features slotted linear rails for mounting sensors and other equipment. To make this mounting system more extensible, two linear slides were installed on top of the rails running along the length of the robot. On top of the slides, two aluminium plates were attached, with machined holes for attaching other components. The top panels allow Bigfoot to be easily reconfigured with different components if necessary. The slides allow for the top panel to be moved without difficulty so that team members can access the inner compartment of the robot, but two transversal pins hold the plates in place so they do not move while the robot is in motion.

Updated Software Architecture

The software architecture for Bigfoot was completely redesigned from previous years. The new code was created with a focus on object-oriented, easily extensible design. Abstract base classes were created to allow for greater polymorphism and code reuse in the robot code. For example, the classes used for sensors all extend a base Sensor class which includes the common features of all sensors. Different motor controller classes (all extending a MotorController class) allow for different control systems to be used without having to change any code outside of which class is used. Navigation is handled through classes extending a base NavigationController class, allowing the robot's code to quickly change between control modes as needed, without requiring changes to other modules when new control modes are added.

Simulation/Testing Focus

Software simulations of the IGVC challenges were created early in the development process. This allows for the navigation features of the robot to be tested earlier in the development process, while also making the software sub-team less reliant on the hardware sub-team to evaluate their progress.

In addition, simulations were used to test the designs for structural components of the robot. This allowed the robot to be constructed using the most efficient use of space and materials, without sacrificing the structural stability of any part of it.

HARDWARE DESIGN

Chassis

Bigfoot is built on top of a Clearpath Husky platform. This platform includes four motors with rubber lug tread tires, a 24 volt rechargeable battery, and a power distribution system to provide 24 volt, 12 volt, and 5 volt DC current to internal electronics. On the top of the Husky platform are four t-slot rails to allow custom components to be easily attached. The Husky platform also includes an emergency stop button to shut off its motors. An inner compartment provides enough space for computers and other electronic parts.

On top of the Husky platform, the reconfigurable top panels and the sensor tower are installed. The plates are, together, 520 mm wide and 659 mm long with 858 x 5 mm(Ø) holes. This configuration makes it easy to mount and reconfigure the sensors on the robot, making this design extremely flexible.

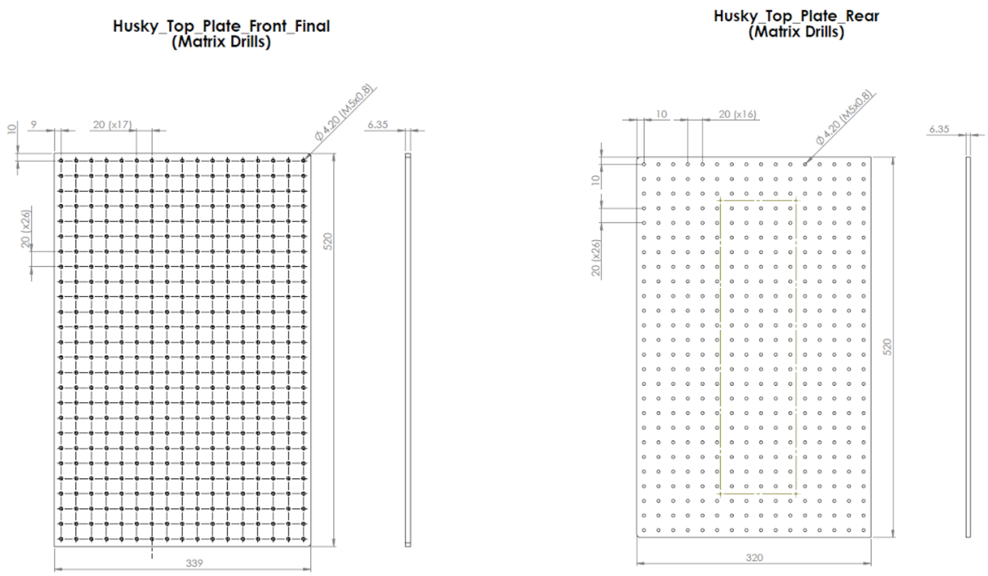

Figure 2. Top Plates Technical Drawings

The sensor tower reaches approximately 130 centimeters above the top panel and includes mounts for the camera, GPS, and digital compass. It is composed of rails similar to the ones on top of the main chassis where the slides are mounted. Computer simulations were performed on the tower structure in order to find the most efficient design and reduce vibrations which could disturb the sensor readings.

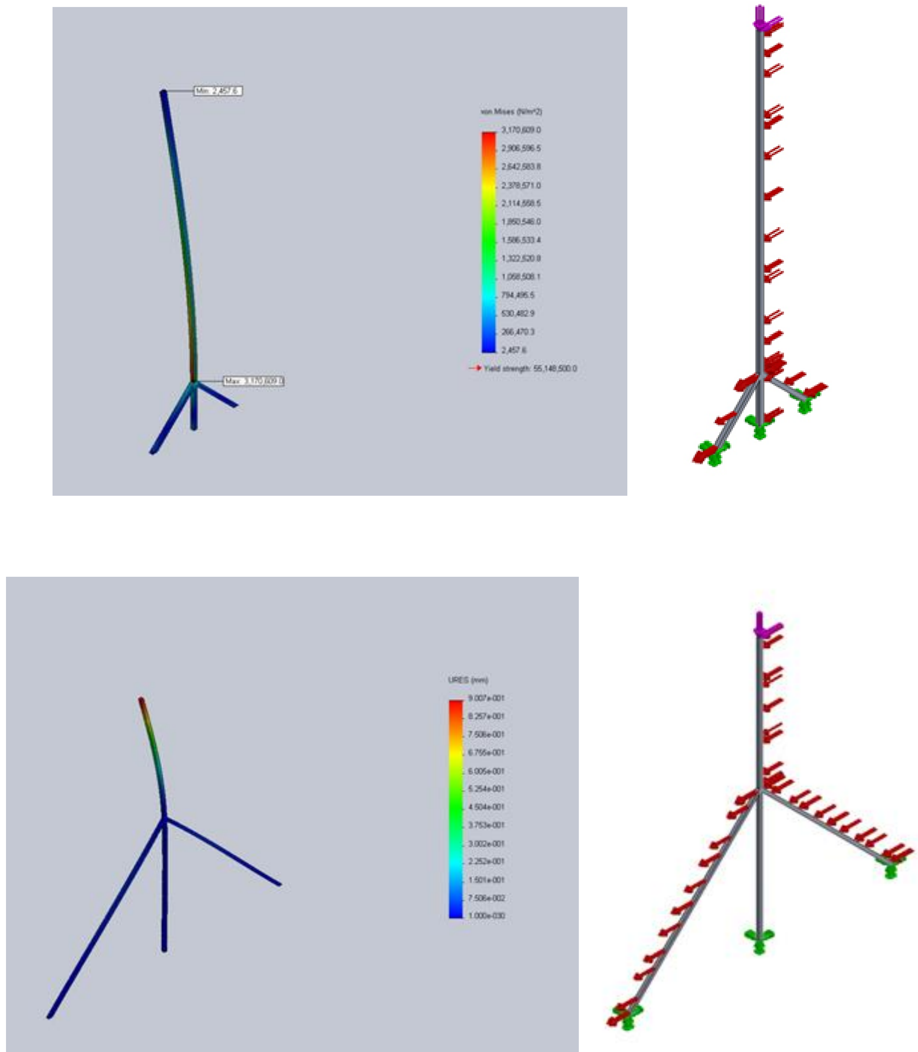

Figure 3. Simulations of forces on sensor tower (with exaggerated displacement)

Electrical Design

Primary power for all systems is provided by the integrated Husky power supply board. This board supplies 5, 12 and 24 volt rails at 5 amps per rail from the 24 volt 20 amp hour battery. Wheel speed is controlled using the integrated wheel encoders with 78,000 ticks per meter. Battery state of charge can be requested from the Husky controller board or read from the LED battery charge display located on the rear of the robot. Communication between the Husky platform, onboard PC, raspberry Pi and relay board is provided by USB and serial connections. A simplified system diagram of all electrical and communication connections is shown in Figure 4.

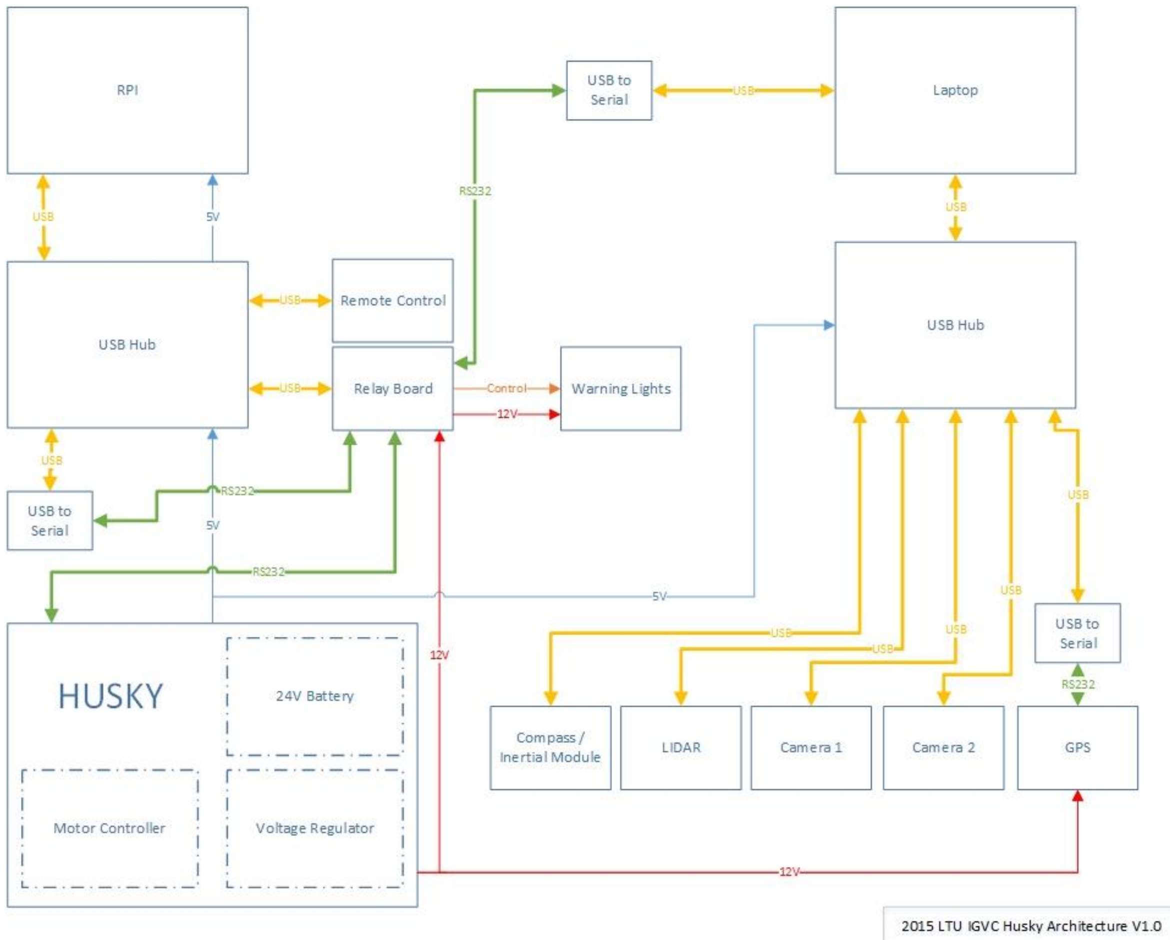

Figure 4. Electrical connections within Bigfoot

Sensors

Camera. Bigfoot's vision is provided by a Microsoft LifeCam Studio webcam. This webcam provides a low-cost, off-the-shelf solution with 1080p video at 30 frames per second.3 Focus and color balance controls are provided through a software interface. The LifeCam has been tested extensively in previous IGVC entries, and has been found to be very reliable.

LIDAR. Bigfoot uses a Hokuyo URG-04LX-UG01, a small, low-cost LIDAR able to detect objects within 5600 mm. A scan is completed at 10 Hz with an angular resolution of 0.36 degrees across a detection area of 240 degrees. The LIDAR is powered by 5V DC provided by the USB connection, and its maximum current consumption is 500mA.[1]

GPS. The latitude and longitude are calculated with the help of Novatel's ProPak GPS receiver. The ProPak provides reliable data with sub-meter accuracy using the help of differential GPS correction. An external antenna provides GPS signal reception.

Electronic Compass. Heading information is provided by a Sparton GEDC-6E electronic compass. The GEDC-6E provides very accurate and easy to configure readings for pitch, yaw, and roll. Although Bigfoot uses it only as a compass, the GEDC-6E contains a 3-axis gyroscope, magnetometer, and accelerometer, providing an entire suite of inertial measurement unit features.[4]

SOFTWARE DESIGN

Software Architecture

Bigfoot's software architecture was designed with a focus on object-oriented design and flexibility. After the design was conceived, abstract base classes for each object type were created. This included base classes for sensors, motor controllers, and navigation controllers.

System Integration

The robot's software relies on a navigation controller object to make decisions based on the input from the various sensors, and then send commands to the current motor controller object. The navigation controller can be replaced at runtime to change the robot's current mode of operation. Autonomous navigation, waypoint following, and IOP are all available as navigation controllers.

Autonomous Navigation

Vision Processing. Input from Bigfoot's camera is primarily used to detect the lines marking the edges of the course. White lines are easily found by transforming the BGR image we obtain from the webcam into an HSV color space. Thresholds are used to isolate the white lines, then filters are applied to eliminate noise from the image.

Sensor Fusion. The vision data provides half of the information used to make navigation decisions. The remaining data is received from the LIDAR. The LIDAR data provides distances to obstacles around the robot, but no information about the lines of the course. The vision supplies reliable information about the lines on the course, but poor information about the obstacles, especially outside of its limited field of view. To overcome the weaknesses of each sensor, their data is combined into a map of the robot's surroundings, the "local grid".

The LIDAR data can be easily converted into local coordinates using trigonometric functions. Visual data requires more processing, due to the perspective of the image received. An interpolation function is used to transform the image space coordinates to approximate local coordinates. With both sets of obstacles in the same coordinate space, the local grid can be assembled.

To prevent issues navigating while driving down a ramp, the lidar data may be discarded when the digital compass recognizes the robot is pitched significantly downward. Without this, the robot would see the ground at the bottom of the ramp as an encroaching wall and attempt to steer away, even if no obstacles are present. Vision will still be used, allowing the robot to avoid obstacles near the bottom of a ramp.

Local Grid Navigation. Navigation in the local grid is performed by testing multiple potential turns and observing the robot's progression for each possibility. Each turn is tested by drawing a curve in the local grid space, representing the turning curve of the robot for the potential input, and testing for known obstacles along the robot's positions along that path.

Potential turns are tested starting at the center and working outward. The distance before an obstacle of each possible path is noted and the navigation logic compares them to find the best path to use in the current situation. If there is a known waypoint ahead of the robot in the course, it will weigh turns leading in that direction slightly higher.

Long distance mapping outside of the local grid was decided to be not necessary for the tasks. A larger map, creating a "global grid", would be most useful in a scenario where the obstacles cannot move significantly between runs or the obstacles must be visited multiple times for completion.

Waypoint Navigation

The Waypoint navigation is done by combining the functionality of GPS and compass. The current heading is acquired by the compass and used to determine the target turning angle. Using conventional methods that are commonly used on automotive navigation GPS to calculate the distance between current position and desired position, the robot is able to vary its speed dependent on distance from a target waypoint.

IOP

The IOP component of this year's software was based on the previous year's JAUS component. Classes have been made to model the entire JAUS topology and JAUS messages sent over JUDP. The services have been rewritten to better comply with the JAUS standards, and a navigation controller class has been created to handle Local Waypoint Driver requests.

TESTING AND SIMULATION

Test Course Simulation

A simulation was created using the Unity game engine. The use of this engine accelerated development because it already included the physics and graphics capabilities required for our needs. This platform also allows the simulation course to be easily modified to test new scenarios. In prior years, testing the autonomous features of the robot at night or during poor weather was not possible, but the simulation is accurate enough that navigation can be tested at any time.

The virtual robot in the simulation consists of several components that match the real life robot. The motor controller, camera, LIDAR, GPS, and digital compass are all simulated. The simulation sensors are implemented into the existing software architecture by creating a SimulationSensor class that facilitates the communications with each sensor, and a SimulationMotorController class that sends commands to the simulated motor controller. The software architecture used allows for the simulation classes to be used with no changes to any other modules.

Once the robot software is connected to the simulation, commands are sent to the simulated motor controller identically to how they are sent to the real motor controller. The navigation controller makes decisions based on the simulated data and the simulation updates based on these decisions. The obstacles in the course are simulated with collision physics for greater realism.

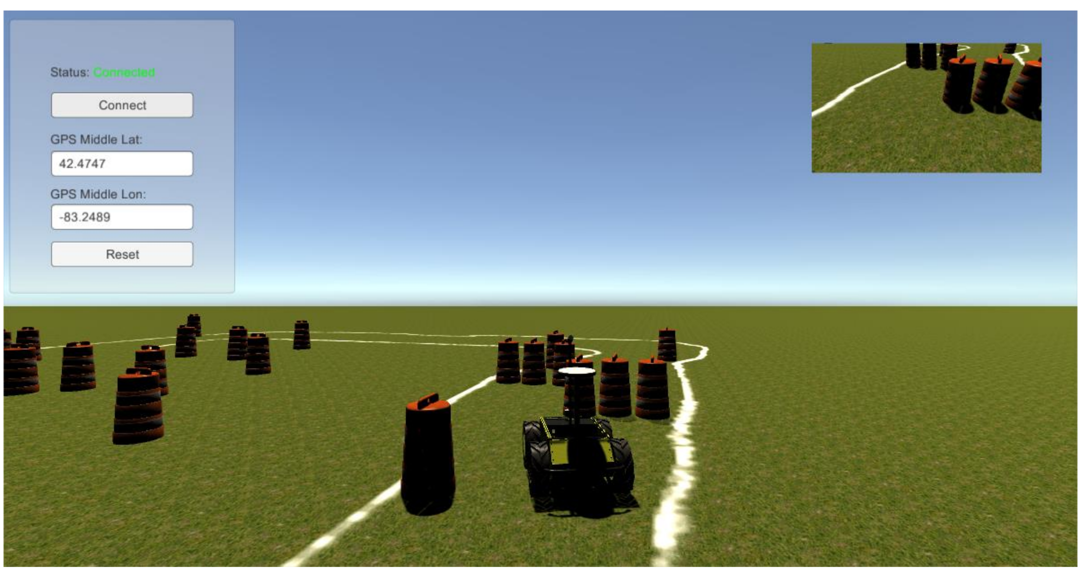

Figure 5. Screenshot of Auto-Nav course simulation connected to robot software.

IOP Testing

Testing software was also created for the IOP feature. A simulated Common Operating Picture (COP) communicates with the robot's computer and runs the known tasks for the competition challenge. Specific commands are available for testing as well. The result of the selected command or test is displayed for the user upon completion, or an error message is displayed if the task is not completed within an acceptable period of time.

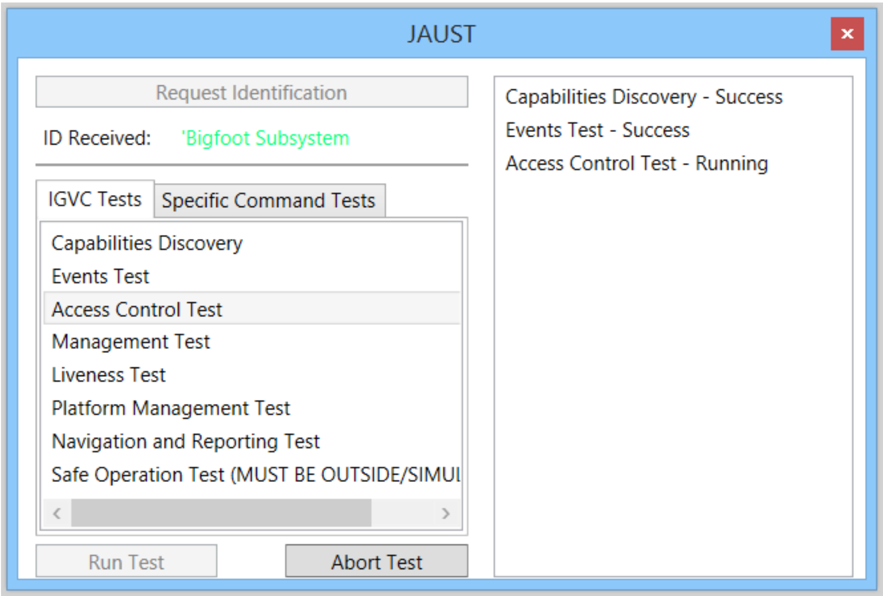

Figure 6. Screenshot of IOP testing tool connected to robot software

PERFORMANCE ANALYSIS

Travel Speed

The Husky robotic platform has a maximum speed of 1 meter per second.[2] However, because this limit is implemented through software instead of being a physical limitation of the motors or electrical systems, the platform is able to reach this speed even with the additional mass of the modifications.

Incline Climbing

The Husky robotics platform is specified as being able to climb a 45 degree incline and drive laterally on a 30 degree incline.[2] The modifications made to the platform have moved the center of gravity. Additional testing was required to ensure Bigfoot would be able to handle the approximately 10 degree inclines potentially included as obstacles on the course.

Trial Data. Using hills available on the Lawrence Tech campus, Bigfoot was tested on inclines exceeding 20 degrees. While the robot was able to both climb and descend the hills without issue, braking is required for the robot to stay at a position on the hill.

Ramps on campus, measured at approximately 10 degrees were also used. The ramps had no significant impact on the performance of the robot.

Reaction Time and High Speed Operations

The navigation controller is limited to running no faster than the camera updates, at approximately 30 frames per second. However, the processing required may take slightly longer than one thirtieth of a second, resulting in a new decision not being made for up to one fifteenth of a second. The motor controller updates the requested speed every 20 milliseconds.

The LIDAR completes a scan ten times a second, so obstacles not detectable with vision or seen in previous LIDAR data may not be reacted to for 100 milliseconds. However, this is not likely to be an issue for autonomous navigation because the obstacles not already seen are most likely at the maximum range of the lidar, giving the robot several seconds to react.

Battery Life

Clearpath specifies that the Husky platform's lead-acid battery is capable of providing power for 8 hours of basic use or 3 hours of heavy use.[2]

The computer used has been upgraded with an external extended battery, which will more than double its battery life. The extended battery is rechargeable and swappable, so it can be replaced with another battery after it runs out.

Trial Data. Tests of the Husky battery found that it could be used for approximately 5 hours for our use. The laptop battery is able to be used for over 10 hours of light use without replacing the external battery.

Obstacle Detection Distance

Vision. The angle of the camera allows Bigfoot to see approximately 5 meters in front of it, with a diagonal field of view of 75 degrees. There is a small blind spot very close to the robot, within the nearest 20 centimeters. However, this blind spot usually does not prevent obstacle detection because the robot avoids situations where the front of the robot is that close to a line, and the LIDAR would detect a physical obstacle within that range.

LIDAR. The LIDAR sensor we are using is Hokuyo URG-04LX-UG01, of which the detection area is 240°, with a 0.36° angular resolution. The specified detection range of the LIDAR is from 20 millimeters to 5600 millimeters. However, the sensor is designed for indoor use only and bright sunlight may have some influence on the maximum detection distance of the sensor. In addition, the detection distance of the LIDAR sensor may also vary with different objects. This bigger the object is, the easier the LIDAR will detect it at a long distance.

Complex Obstacles

When facing a switchback, the robot will treat it like any other turn on the course. It will attempt to find the best available path through the detected obstacles, which should lead it safely around the turn. Center islands will also be detected like any other obstacle, and the robot will navigate around the island. Potholes will be detected by the vision code looking for the lines, which will also place the pothole on the local grid. Dashed lines are detected by the software like other lines, and will definitely be avoided as long as the hole is not significantly larger than the robot.

Failure Point Identification

Most of the components used in Bigfoot are available off-the-shelf and compatible with alternative parts. The failure points found are largely total failures of each component, which would require a quick replacement with a compatible part.

Parts using USB to communicate were chosen for greater compatibility with different computers and to remove adaptors as a failure point. The main software is able to run on any Windows computer with the processing power required to make decisions in a timely matter.

Location Accuracy

The DGPS data available through the Novatel GPS receiver can determine its location with sub-meter accuracy.

Trial Data. Tests using known waypoints found that the GPS sensor could position the robot within 1 meter of the desired waypoint.

FAILURE POINTS AND MODES

Raspberry Pi

The Raspberry Pi controlling the relay board and non-autonomous motor control could fail. If this occurs, the board will need to be replaced. As it is available off-the-shelf, a new board would be easy to obtain quickly, and the code would be restored from a backup. In the event that a new board cannot be obtained in time, the relay board could be rewired to give the laptop control of the motors. The lights could also be re-wired to be controlled through an alternative method.

Relay board

Partial. In the event that a single relay becomes stuck or unresponsive, its function can be moved to a different relay. The code running on the Raspberry Pi would be modified to command the new relay.

Total. If the entire relay board fails, or too many relays fail to allow wires to be moved to working relays, the relay board will need to be replaced. The board is available off-the-shelf, and extra boards will be brought to the competition to be used in the event of a failure.

Motor

The Husky platform's motors are a potential failure point. The windings in the motor could become damaged, causing a cascade failure limiting the robot's mobility.[5] However, the only known case of this happening was due to modifications made to the drivetrain of another team's robot and is very unlikely to occur in Bigfoot. In the event that the motors do fail, the Husky platform will need to be returned to Clearpath for repairs and an alternate platform will need to be found. The modular design of the reconfigurable top panel means that it would be easy to separate the chassis from the other parts of the robot for repairs.

Sensor

All of Bigfoot's sensors are reliable and available off-the-shelf. If a sensor fails, it will simply be replaced with a compatible part. The reconfigurable design allows for any of the sensors to be easily detached and replaced.

Main Computer

Bigfoot's main computer is easily replaceable. The software is stored on a remote Git repository, allowing it to be transferred to any computer over the internet. In the event that the laptop fails completely, its USB ports fail, or it becomes damaged in some other way that prevents it from being used, the laptop can be switched out for any other laptop with sufficient USB ports to connect to all of Bigfoot's sensors and sufficient computing power to make decisions in time.

Battery

The battery used to power Bigfoot could become damaged and lose its ability to store a charge. If the battery becomes damaged, it will be replaced with the extra battery that was purchased with the Husky platform.

Wiring

In the event that the wiring inside Bigfoot becomes damaged, the hardware sub-team will focus on finding the damaged wire and replace it with new wiring.

Software

In Bigfoot's testing, it was found that dead or dormant grass can sometimes be light enough for the vision processing software to classify it as white area from a line or pothole. However, this is not expected to be a major issue during the competition due to the time of year and weather expected. Although making a stricter filter for finding visual obstacles could result in the robot not seeing a line, additional filtering will be added if deemed necessary at the competition.

BILL OF MATERIALS

Table 1. Costs of Bigfoot Components.

Component	Retail Cost	Cost to Team
Clearpath Husky robotics platform (with extra battery)	$10000	$6250
HP Z-Book Laptop with 2 extended batteries	$2200	$200
Raspberry Pi microcomputer	$25	$25
Sparton GEDC-6E electronic compass	$1,350[4]	$0
NovaTel ProPak-LB GPS receiver	$2700	$1000
Microsoft LifeCam Studio camera	$60	$60
Hokuyo URG-04LX-UG01 LIDAR	$1140	$1140
Relay board	$30	$30
Safety lights	$50	$50
Misc. Electrical	$100	$100
Misc. Hardware	$100	$100

CONCLUSION

Bigfoot was built on top of a very rugged and reliable platform. By focusing on using off-the-shelf hardware with an easily reconfigurable design, the team's extensions to that platform will also be reliable. The software focus on simulation and testing allows us to ensure that Bigfoot will continue to be reliable no matter what situation it is put into.

REFERENCES

[1] Hokuyo Automatic Co., Ltd. (2009). *Scanning range finder (SOKUIKI sensor)* [Online]. Available: https://www.hokuyo-aut.jp/02sensor/07scanner/urg_04lx_ug01.html

[2] Clearpath Robotics. (2015). *Husky Tech Specs*[Online]. Available: http://www.clearpathrobotics.com/husky/tech-specs/

[3] Microsoft Corporation. (2014). *LifeCam Studio Technical Data Sheet* [Online]. Available: http://download.microsoft.com/download/8/8/5/8854555F-8FF8-407A-8CD0-0B10D7FC4176/TDS_LifeCamStudio.pdf

[4] Sparton Corporation. (2015). *GEDC-6E - Gyro-Enhanced Inertial System* [Online]. Available: https://www.spartonnavex.com/product/gedc-6e/

[5] University of Detroit Mercy IGVC Team. (2013). *Crisis -- the plot thickens!*[Online]. Available: http://eecapstone.udmercy.edu/IGVC2013/the-road/husky-crisis----overheating/