5-15-2017

iWheels 3

Lawrence Technological University

Team Captain:	Devson Butani	dbutani@ltu.edu
Faculty Advisors:	CJ Chung	cchung@ltu.edu
	Jonathan Ruszala	jonathanruszala@gmail.com
	Gordon Stein	gstein@ltu.edu
Team Members:	Sean Bleicher	sbleicher@ltu.edu
	Kevin Cox	kcox@ltu.edu
	Nirmit Changani	nchangani@ltu.edu
	Muhammad Aftab	maftab@ltu.edu

Faculty Advisor Statement

We, Dr. CJ Chung, Jonathan Ruszala, and Gordon Stein of the Department of Math and Computer Science at Lawrence Technological University, certify that the design and development on the iWheels 3 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.

Chan Jin Chung

5/15/2017

Signature Date

J Ruszala

5/15/2017

Signature Date

Gordon Stein

5/15/2017

Signature Date

DESIGN REPORT FOR THE LAWRENCE TECHNOLOGICAL UNIVERSITY “iWHEELS 3” 2017 INTELLIGENT GROUND VEHICLE COMPETITION

Devson Butani, Sean Bleicher, Kevin Cox, Nirmit Changani, Muhammad Aftab, Dr. CJ Chung, Jonathan Ruszala, Gordon Stein

INTRODUCTION

The Lawrence Technological University (LTU) Intelligent Ground Vehicle Competition (IGVC) Team has developed upon the design of the previous year, to create a new entry for 2017, iWheels 3. The team observed the performance and structure of the previous year’s entry and determined the gaps for improvement. iWheels 3 features a completely new drivetrain from an old robot wheelchair, improved organisation and waterproofing, new electronics, improved simulation, new vision code, and new navigation code. The hardware and software design both began in spring.

TEAM ORGANIZATION

LTU’s IGVC team is split into two task oriented groups. One group of the team is focused on hardware, and the other group is focused on software. Due to the small size of the team, a decision was made to not break these groups into specific task based responsibilities, but to instead give each member of the software team at least one focus. These focuses included: navigation and perception, IOP, and simulation. All members participated in creating the design report.

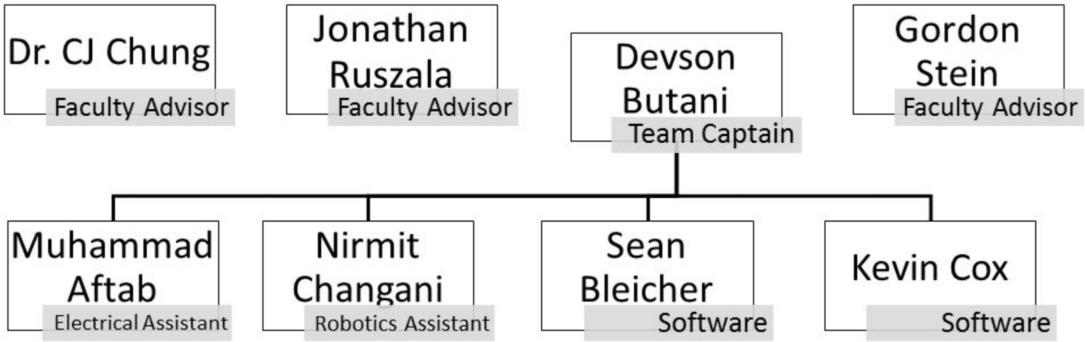

Figure 1. LTU 2017 IGVC Team Organization Chart

Design Concepts and Goals

The main design goal for iWheels' hardware was to economise and organise components wherever possible. The electronics enclosure furthers this goal by waterproofing circuits and simplifying robot servicing. The sensors used were readily available and possible to replace from old robots if need be. We designed the robot to have a seat so that it can be used for the development of intelligent self-driving wheel chair system after the IGVC competition.

The main design goal for the software was to implement new sensors and improve the reliability of current sensors. Use multiple sensors to more accurately measure speed, distance, and gps coordinates helps with navigation. Also software improvements to IOP communications were made to better follow IOP standards.

DESIGN INNOVATIONS

Simple frame design

From previous year's designs, it was observed that using aluminium extrusion frames was cheap and highly customizable for the required sturdiness. Although making frames that carry weight require reinforcements therefore, major mass of the batteries and the payload was accommodated as low as possible. As the drivetrain was designed to be a wheelchair by the manufacturer, a small seat was added to carry the payload. This seat would further be used as an autonomous patient transport application after the competition. Using the Aluminium frame solely for sensors makes the design simple and lightweight. For structural stability, the frame and the seat both were mounted directly to the vehicle chassis.

Space optimization

A waterproof enclosure for electronics simplifies cable management through the frame and reduces time required for repairs by locating all boards in a central location. Keeping the batteries inaccessible below the seat allowed for safety from unintentional electrical hazards. Extra space under and behind the seats was used for laptop storage and cooling. Overall, the vehicle maintains the compact the structure and safety without losing its primary function as a wheelchair.

Simulation/Testing/Monitoring Focus

A laptop was mounted on the robot such that in case anything in the program goes wrong then instead of connecting the laptop and finding a comfortable place to keep it on, the laptop can be pulled out right away while seating on the robot.

MECHANICAL DESIGN

Overview

iWheels 3's mechanical design was redesigned based on the performance of the last competition robot Bigfoot 2. The drivetrain on Bigfoot 2 had caused difficulty turning on many surfaces due to the high traction output without a differential, so it was replaced by a simpler two wheel drive drivetrain available. The previous entry was not as fast as the rules changed to allow for faster robots, so new motor controller and motion controller were selected to increase the robot's speed with precise control. The sensor tower from the previous year had issues with vibration as the robot had no suspensions unlike this year's drivetrain. Previous years' entries were not water resistant, preventing them from being used during inclement weather at the competition, so steps were taken to weatherproof the robot.

Frame and Chassis Structure

iWheels 3 is built on top of a Arrow Storm Series wheelchair platform. This platform includes two motors with rubber tread tires and a rigid chassis for carrying weight. On the top of the frame are four t-slot rails to allow custom components to be easily attached. An inner battery space provides enough space for two 12V lead acid batteries and all the electronic boards.

On top of the chassis the seat, extrusion frame and the sensors are installed. The entire platform is, 638 mm wide, 1130 mm long and 1770 mm tall. This configuration makes it easy to mount the camera right below the height limit for wider vision and reconfigure the sensors' positions on the robot, making this design extremely flexible. The frame includes mounts for the seat, footrest, camera, GPS, LIDAR, laptop, and digital compass.

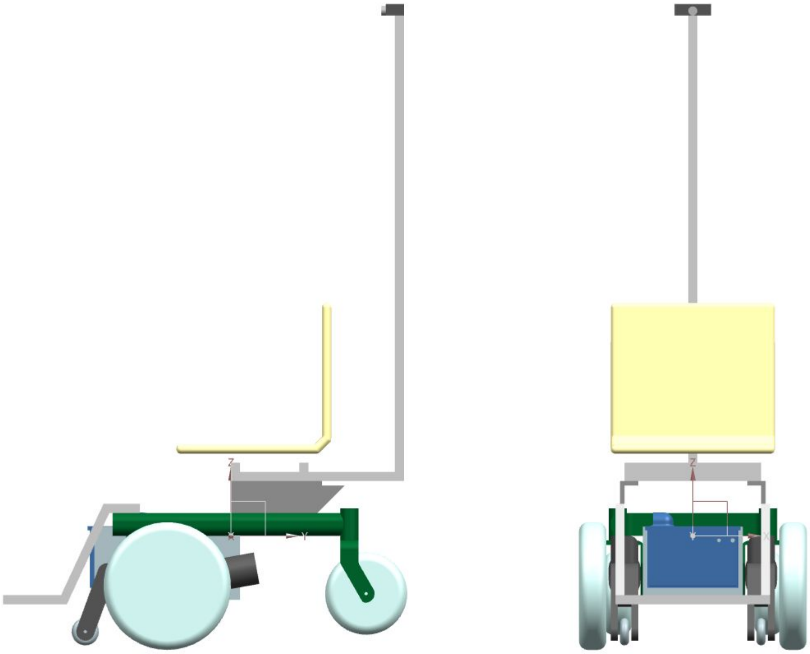

Figure 2. iWheels 3 Left Side and Front View

Electronics Enclosure

A 3D printed electronics enclosure was made to avoid making individual mounts for all the boards and terminals only to waterproof them all separately then after. The enclosure allows for electrical isolation from the chassis and metal frames with added organisation of components and easy access to boards that need servicing for testing. The enclosure also houses the terminal bus connectors from the battery to isolate the battery terminals from any metal exposure. It also has cable outlets for more secured harnessing and mounting space for switches and indicator lights. All the wires inside would be fastened to the walls such that the wires don't damage while the vehicle is moving through rough terrain. Also 3D printing allows for mass customization of mounting angles and holes.

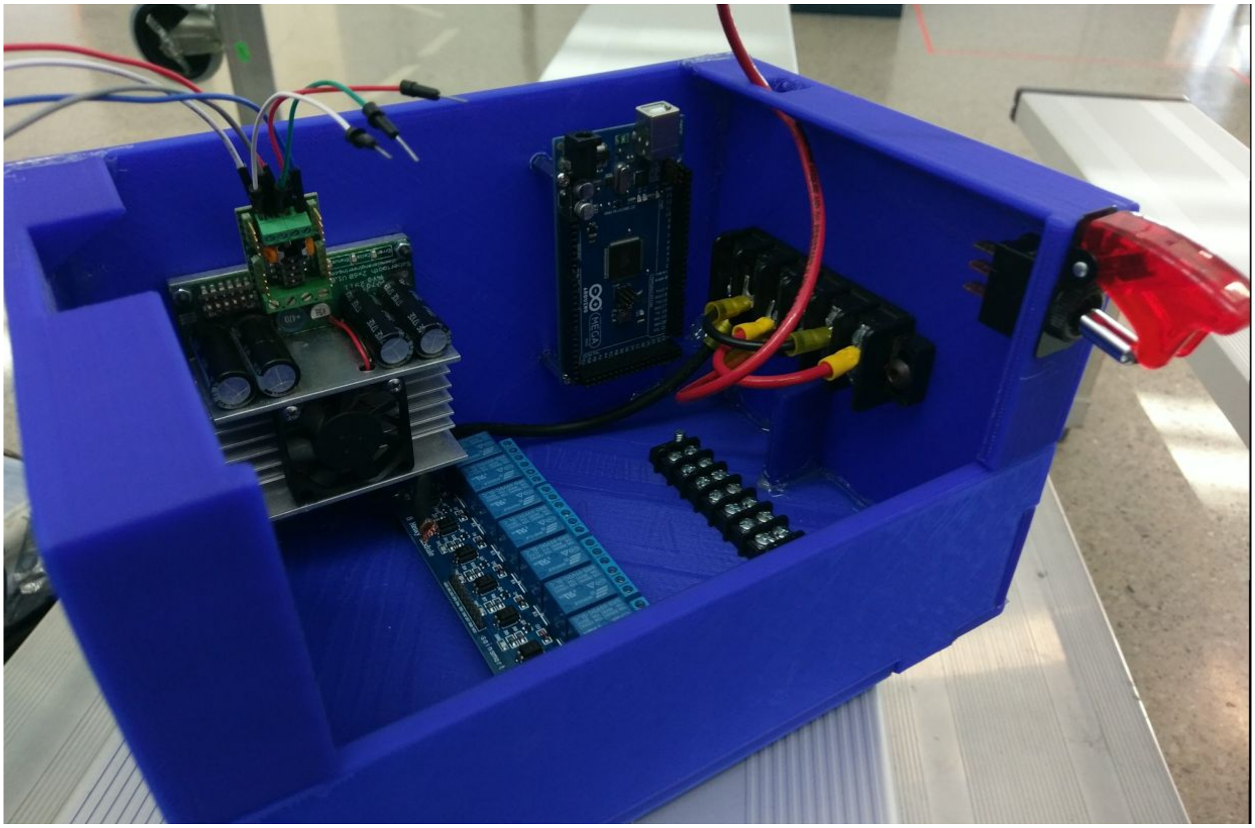

Figure 3. Electronics Enclosure

Adjustable Camera Mount

Looking at previous years, to maintain a camera angle and to keep the camera aligned with the robot’s horizontal was done by trial and error. To save time and maintain the same accuracy throughout the life, a new camera mount was designed and 3D printed to fit the Genius WideCam F100.

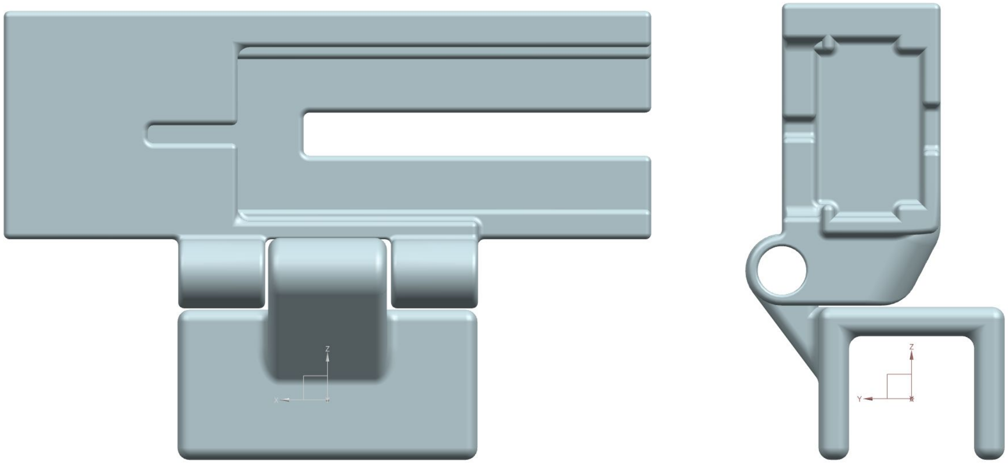

Figure 4. Camera Mount

Suspension

The Husky platform for Bigfoot 2 from previous year lacked a suspension system. Modifying it to add suspension would not be possible without creating an entirely new platform, so instead using the new wheelchair chassis with in-built suspensions was an added benefit. In the previous year, it was noted that the lack of suspension in the chassis caused significant vibration in the tall sensor tower to slow down processing. The new tower is designed to be sturdier and more rigid when the robot is driving on uneven terrain.

Weatherproofing

To reduce the risk of damage due to weather, the electronics are covered under a polycarbonate sheet such that water or dirt does not go into the electronics enclosure but there is free flow of air for cooling. The sheets used are clear so all the component indicators can be seen by the team as the robot runs in the rain.

ELECTRICAL DESIGN

Overview

iWheels 3's electrical design was simplified by two factors: the direct power distribution from the batteries, and the decision to use off-the-shelf modules and components where possible. The communications throughout the robot are handled by one Arduino Mega 2560 as a communication hub talking between the laptop, the Sabertooth 2x60 motor controller and the Kangaroo X2 motion controller over Serial, and the Turnigy IA6B wireless receiver and relay board over PWM and IO respectively; this allowed for standard USB hubs and cables to be used for data acquisition from the sensors connected directly to the computer. This simplified design allowed the team to make the most out of the small number of non-software students available.

Power Distribution

Primary power for all electronics is provided directly by one of the two 12V 35Ah lead acid (permanently sealed) batteries. The two batteries are connected in series and parallel simultaneously with isolated grounds to give 24V power supply to the motors and 12V power supply to rest of the electronics. Considering budget, time and electrical experience constraints, this setup was a fair option requiring no additional components. The battery eliminator circuit in-built in the motor controller powers the motion controller, wireless receiver and Arduino with 5V. The laptop used as an onboard computer has its own internal battery, which is augmented with an additional external battery, neither of which rely on the robot's power supply. Wheel speed is controlled using the self tuning Kangaroo X2 motion controller which is connected to wheel encoders with 9,000 lines per meter. Battery state of charge for the 24V system can be requested from the Sabertooth motor controller board however, the 12V and 5V system charge is not monitored as their current draw is directly reflected onto the 24V system. A simplified system diagram of all communication connections is shown in Figure 5.

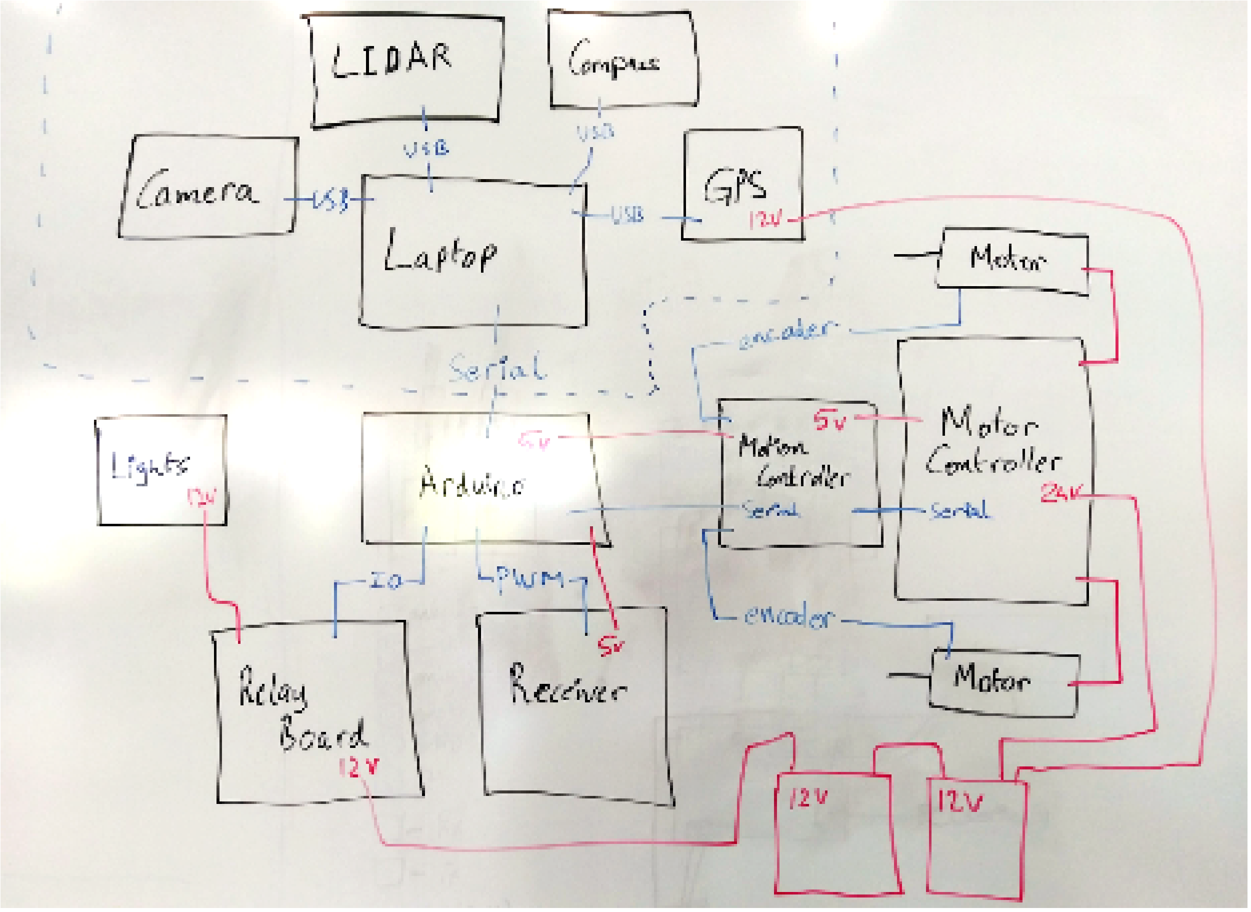

Figure 5. Communication Connections Within iWheels 3

Electronics Suite

CPU

iWheels 3’s computational power is provided by a HP Z-Book laptop. This laptop has an Intel i7 processor and 16 GB of RAM, along with an Nvidia Quadro GPU with 4 GB of VRAM. The power available in this computer allows for more complex vision and navigation code. The primary storage drive of the laptop is a solid state drive, allowing faster access to files.

Sensors

Camera. iWheels 3’s vision is provided by a Genius WideCam F100 webcam with a 120 degree viewing angle. This webcam provides a low-cost, off-the-shelf solution with 1080p video at 30 frames per second and 30 Mega pixel images. Manual Focus feature allows for more accurate vision capabilities and color balance controls are provided through a software interface. The WideCam has been tested to be very reliable.

LIDAR. A Hokuyo URG-04LX-UG01, a small, low-cost LIDAR able to detect objects within 5600 mm was used. 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.

GPS. The latitude and longitude are calculated with the help of Novatel 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 iWheels 3 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.

Quadrature Encoders. The motors used for the chassis of iWheels 3 are equipped with encoders, allowing the robot to determine its current speed on most ground conditions.

Safety Devices

Safety Light. A light has been attached to the body of the robot to be used as a safety light. The lights are turned on when the robot is power on, and stay in solid mode when the robot is driven manually. When the robot is in autonomous mode, the lights will start to flash.

Mechanical Emergency Stop. One emergency stop button has been attached to the rear of the robot. The E-stop is wired (12V) to a normally open relay that drives two 24V capable normally closed relays connected to the motor power cable. This setup allows for minimum power consumption and safety if the E-stop malfunctions.

Wireless Emergency Stop. The wireless remote control transmitter used for manual driving itself has an independent communication channel for an E-stop switch. The wireless receiver is connected to a receiver controlled switch which operates the two 24V relays with 12V. The additional switch is required for voltage isolation and direct relay control from receiver. Moreover the receiver has in-built fail safe code to engage the switch if the connection to the transmitter is lost.

Multiple Relays. Using three layers of relays connected to different levels of faults to control the 24V relay for stopping the motors ensures that the robot will immobilize as soon as a fault is detected.

SOFTWARE DESIGN

Overview

The 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.

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.

Obstacle Avoidance and Detection

The camera is used for lane detection. A multi-channel adaptive thresholding algorithm is applied to isolate the white boundary lines of the course. In order to avoid small amounts of noise from being detected as obstacles, this image data is processed with an erosion operation. A dilation operation is then used to restore the original size of the genuine obstacles.

LIDAR has been used to implement obstacle detection and avoidance. The LIDAR is a laser sensor used for area scanning. Under the most ideal condition, the scan area is 240º semicircle with 0.36 degree angular resolution and radius range of 20mm to 5600mm. The LIDAR can detect obstacles by processing the data reflected from the obstacles. The LIDAR can be used to collect the distances to obstacles around the robot within the detection range, but cannot help with the course lines detection.

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 assemble.

Sensor Fusion Improvements

While previous years' entries only utilized vision, compass, and LIDAR data, iWheels 3 incorporates data from additional sensors. The GPS and encoders are now combined using a Kalman filter to overcome the limitations of each sensor to obtain a more accurate coordinate for waypoint following and mapping of the course. In previous years, the inaccuracy and drift of the GPS used would occasionally cause the platform to steer away from the correct position of a waypoint. With this new filter, the robot is aware of the distance it has travelled and can better determine its longitude and latitude when starting from a known point.

Software Strategy

The software was created with a modular architecture to allow for easier simulation and modification. There are superclasses for navigation algorithms, motor controllers, and sensors. The sensor and motor controller classes have variations for both the simulated and physical versions, allowing for the software to switch to either the real robot or a simulated one with the click of a button. This switch is not visible to the navigation controller, so it behaves the same in a simulated environment or a real one. Making the code more modular also allows for easier team collaboration using a version control system.

Map Generation

The vision data provides half of the information used to make navigation decisions. The remaining data is received from the LIDAR. 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".

Goal Selection and Path Generation

A path for the robot to follow is chosen by testing multiple potential paths against the known obstacles on the local grid. 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. These possible paths are created using the known turning characteristics of the chassis.

The goal is selected using a system of waypoints. Approximate waypoints of the corners of the field (and the provided waypoints for the competition field) are given to the navigation code to find an initial direction. If the turn that would lead the robot in the direction of that waypoint is available, it will move in that direction. The current goal waypoint is switched to the next waypoint when the robot comes within a certain threshold distance. This waypoint system is also used during the qualification run.

Interoperability Profiles and JAUS

The JAUS framework from the previous year was expanded and improved. The existing code base had serious limitations preventing proper operation during the performance tasks. To alleviate this issue, the mobility code was recreated so that the robot can properly complete the task. The new sensor data from the Kalman filter allows for greater accuracy in position and velocity measurements, which are required to complete these tasks.

FAILURE POINTS AND MODES

Hardware failure points

Arduino Mega 2560

The Arduino controlling the relay board and non-autonomous motor controller 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 using an USB to IO output module to give the laptop control of the relay board and USB to TTL Serial cable to connect to the motor controller.

Relay Board

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

If the entire relay board fails, or too many relays fail to not 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.

Motors

The in-built motors are a potential failure point. The windings in the motor could become damaged, causing a cascade failure limiting the robot's mobility. 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 iWheels 3. In the event that the motors do fail, an alternate platform will need to be sourced. The modular design of the reconfigurable electronics means that it would be easy to migrate the chassis using the same parts.

Sensors

All sensors sensors are reliable and commercial off-the-shelf (COTS). 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

iWheels 3'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 iWheels 3's sensors and sufficient computing power to perform the necessary calculations.

Battery

The battery used to power iWheels 3 could become damaged and lose its ability to store a charge. If the battery becomes damaged, it will be replaced with the two exactly same spare batteries.

Wiring

In the event that the wiring inside iWheels 3 becomes damaged, the team will focus on finding the damaged wire in the easy to access electrical enclosure and replace it with new wiring.

Software Failure Points

In iWheels 3'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.

Failure Prevention Strategy

Our failure prevention strategy is to keep the robot's design as simple and as modular as possible, while also relying on COTS parts which have been extensively tested by not only the companies providing them, but also many other users of the same part. Fewer parts on the robot mean that there are fewer parts to break and troubleshoot. A modular design means that failures in one portion of the robot are less likely to affect other parts.

Testing

Testing work has been separated to two major parts, hardware and software. For hardware part, each component has been checked before being assembled to the robot. And then, functional test will be conducted to check if the vehicle can perform all the required functions. For software part, unit test has been conducted first, and after being integrated into the main program, integration test will be conducted to make sure nothing will be incompatible. After the combination of hardware and software parts, run and test the vehicle in real environment and check it follow the competition requirements.

Safety Design Concepts

The Safety Design Concepts have been kept from last year. All safety devices that are based on hardware including: Seat belts, which secure the payload or the passenger; one Safety Light, which is turned on while the vehicle is running; Mechanical Emergency Stop, which is activated by a red button; and Wireless E-Stop Device, which can stop the robot remotely within the controllable distance with a fail safe; Software controlled E-stop, which works when any communication link is broken; sensitive motor controller E-stop, which detects over-voltage and over-current situations. By adding these safety equipments, people can get alert while the vehicle is running, and prevent danger from happening. Once a failure occurs, the shutdown circuit will activate immediately.

TESTING AND SIMULATION

Simulation for the platform was done using an upgraded version of the simulator created for the previous year. This simulator communicates over a network with the robot software, sending updates of the robot's state to simulated sensors, and receiving commands to be sent to a simulation of the real platform's motor controller.

Fewer changes to the simulator were made this year than the previous year. It was noted that using the preliminary design of the robot for the simulation model was not worth the additional time spent, as that design changed in the final platform. The selection of obstacles available in the simulation test course were sufficient for the new IGVC competition course. Some enhancements were made to make the simulated GPS more accurate in reflecting the behavior of the real GPS sensor on the platform.

This simulation was used for both the auto-nav code and the IOP code. This allowed the team to have access to a robot and a field to test in, even when the weather was not good or the robot was not possible to use. In addition, using a simulation reduces the risk of testing new code, as the robot cannot possibly damage itself in a simulation.

PERFORMANCE TESTING

Travel Speed

The robotic platform has a maximum speed of 5 miles per hour. 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 robotics platform is specified as being able to climb a 30 degree incline and drive laterally on a 30 degree incline. The modifications made to the platform have moved the center of gravity. Additional testing was required to ensure the platform would be able to handle the approximately 10 degree inclines potentially included as obstacles on the course.

Using the terrain available on the Lawrence Tech campus, iWheels 3 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

The 24V battery module used is estimated to last 2 hours under continuous load with a full charge. 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. Tests of the iWheels 3 battery module found that it could be used for approximately 3 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

The angle of the camera allows iWheels 3 to see approximately 4 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 10 centimeters, due to the body of the robot blocking the camera's view. 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.

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. Using the waypoint system, the robot is very unlikely to end up turned around by a switchback. Center islands will also be detected like any other obstacle, and the robot will navigate around the island. Simulated 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 iWheels 3 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. Tests using known waypoints found that the GPS sensor could position the robot within 1 meter of the desired waypoint.

INITIAL PERFORMANCE ASSESSMENTS

In our initial tests, the robot was able to navigate easily around barrels and other obstacles while navigating. We noticed that the turning and stability of the tower are greatly improved from the previous year. In addition, the new organisation of the components allows for easy access for servicing electronics, which proved to be very useful during our tests.