Dokalman MK2

University of Cincinnati

27th Annual Intelligent Ground Vehicle Competition

Evan Baumann ✦Clay Curran ✦ Sam Heyl ✦ Joe Hirschfeld ✦ Jordan Jacob ✦ Emily Kiehl
Joseph Knight ✦ Amanda Mitchell ✦ Sam Oakes ✦ Michael Shuster ✦ Alex Suer

Contact information for team members provided in Team Organization section

Submitted 5/14/19

CERTIFICATION:
I certify that the engineering design in the vehicle Dokalman (original and changes) by the current student team identified in this Design Report has been significant and equivalent to what might be awarded credit in a senior design course.

Janet Dong

Doctor Janet Dong, Advisor

Introduction

The University of Cincinnati Robotics Team's proud tradition of competing in the Intelligent Ground Vehicle Competition started with the Bearcat Cub in the first Intelligent Ground Vehicle Competition. This year the team brings a completely new robot that was built from the ground up to compete in the 27th Intelligent Ground Vehicle Competition.

Team Organization

Role	Name	Major	Email*	Grad Year
President/Lead Engineer	Evan Baumann	Mechanical Engineering	baumanea	2020
Software/Electrical Lead	Joe Hirschfeld	Computer Science	hirschjb	2019
Mechanical Lead	Clay Curran	Mechanical Engineering	curranct	2022
Vision Lead	Sam Oakes	Computer Science	oakessa	2022
Treasurer	Amanda Mitchell	Electrical Engineering	mitch2an	2022
Member	Sam Heyl	Computer Engineering	heylse	2022
Member	Jordan Jacob	Computer Engineering	jacobjr	2020
Member	Emily Kiehl	Information Technology	kiehley	2023
Member	Joseph Knight	Computer Engineering	knightjp	2019
Member	Michael Shuster	Computer Engineering	shustema	2020
Member	Alex Suer	Mechanical Engineering	suerad	2022

*Note: Emails are suffixed @mail.uc.edu

Design Process

The Dokalman MK2 design process began with the goal to capitalize on the successes and lessons learned from previous competitions. The team evaluated the competition requirements to identify opportunities for improvements.

On the mechanical side, a singular design was created by a group of team members. A CAD model was then developed. Once it was finalized, the parts were sent off to be manufactured and eventually assembled in-house. The entire mechanical design was validated via an RC drive test.

The electrical design was done in a similar manner, where multiple members created the PCB design then assembled the finished PCB in-house. The PCB was tested independently of the robot before later being tested as part of the entire robot system.

Software development was a much more iterative process. Tasks were defined in an issues list where members would work on specific issues and submit code for a peer review before they were included in the main codebase. This allowed many members to work on separate parts in parallel.

Innovative Concepts and Technologies

Dokalman MK2 features a completely redesigned frame and gearboxes. Redesigning the robot frame and the interior of the robot allowed a more effective use of space. The robot meets the competition's minimum-size requirements while still allowing room for the needed computers and sensors. Custom gearboxes allow the robot to have the necessary power and speed that is required for competitive physical performance. The robot as a whole is safe from dust and water, giving the robot a IP53 rating. This allows the robot to operate normally even in conditions that could be hazardous for electronics.

In previous years, the competition robots had several simple circuits scattered about the robot. This year, the team decided to create custom circuit boards which simplified much of this circuitry down to a small, easy-to-manufacture circuit board. This circuit board sits atop of a Raspberry Pi and communicates with it via a packet-based UART protocol. This new circuit board handles emergency-stop circuitry, voltage monitoring, temperature feedback, LED status lights, and joystick inputs from a Spektrum radio receiver. This is a major improvement over previous models because not only does it make the robot easier to diagnose and assemble, but also makes the robot safer with its fault-tolerant, real-time, emergency-stop design.

This year the team also focused on improving the robot's vision-processing capabilities. The team recognized that vision is an incredibly parallelizable task which is performed much better by a computer with a graphics processing unit (GPU). The team specifically selected the NVIDIA Jetson TX2 GPU module for its impressive performance as a standalone system at its price point. All of our vision code now utilizes a CUDA specific processing pipeline for its increased performance.

The line and lane detection algorithm received a large update this year as well. In the past, the robot could only detect lines as they were marked on the grass. This year, additional filtering and logic were applied to the line points. The vision algorithm detects these to filter out invalid lines which would have otherwise been impossible to see, such as those under the robot or behind an obstacle, as well as apply lane detection. This improvement will address issues we have had in previous years regarding blind spots and returning to lanes from no-man's land.

Mechanical Design

Overview

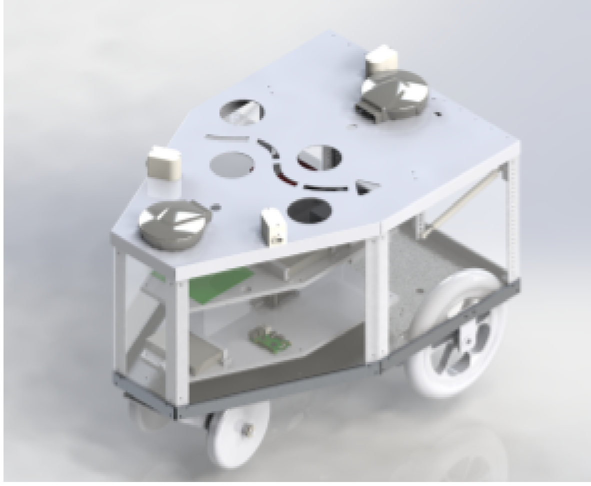

Figure 1: Dokalman MK2 Render

Dokalman MK2 underwent a complete mechanical redesign from last year's submission. This new design's footprint is the minimum allowable size which will allow for easier navigation of the course. Many of the large, metal parts of Dokalman were CNC laser cut and bent, while smaller more complex parts that could be made with plastic were 3D-printed.

Dokalman MK2 consists of three main sections: the drive platform;the frame/top panel; and the electronic panels. The drive platform consists of the bottom tray, two gearboxes, and the front caster wheel assembly. The frame/top panel acts as a weatherproofing shield around the drive platform and electronics panels. Mounted on the top panel are two wide-angle cameras, a LiDAR sensor, and GPS antennas. The electronics panels include two MDF panels for power management, networking, computers, and an IMU sensor . The status lights are mounted to the bottom pan of the robot.

Frame

The drive platform consists of both a bottom pan and a top rim, both of which were laser cut and bent to shape. The bottom pan and top rim are separated by laser cut aluminum struts attached by rivets to the pan and rim. The top panel was also laser cut and bent and then attached to the top rim by a rubber hinge. The all-aluminum construction of the frame allows for a robust and lightweight frame. Acrylic panels line the outside and include weather-sealing foam to create a water and dust-resistant barrier. The top panel is hinged and locked down with an internal latch that allows access to the electronics while also being secure from unwanted access. To prevent overheating, Dokalman MK2 includes an air intake and exhaust fan at the rear of the robot, circulating air around the robot. Due to the excess heat the LiDAR creates, the LiDAR has its own active cooling system that includes a small fan and heatsink combination.

Drive Train

Dokalman MK2 is propelled by a differential drive system that powers the two back drive wheels. A custom caster wheel assembly is mounted at the front of the robot, allowing for better maneuverability. Power is provided to each drive wheel by two CIM motors with a custom gearbox assembly. Each motor is able to provide 2.41Nm of torque at 5330 RPM.

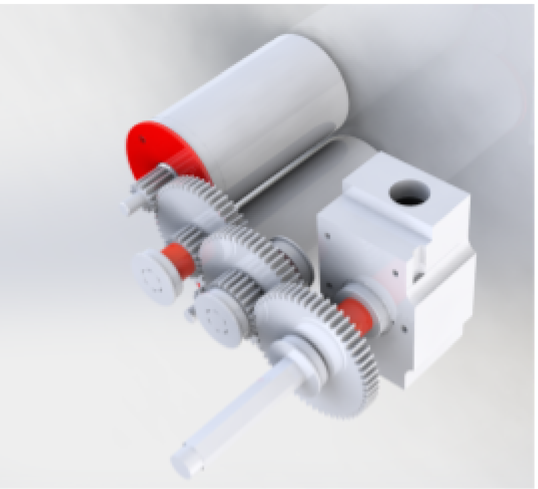

Figure 2 Gearbox

Each gearbox is made up of gears inset within a two part 3D-printed gear enclosure filled with lithium grease surrounded by two one-quarter-inch aluminum plates that hold the bearings and the print in place. The print was designed in a way to minimize the empty space inside the gearbox, keeping the lithium grease from coming off the gears. This fully-enclosed gearbox design provides a higher level of safety as obstructions such as debris or human limbs are unable to be caught in the gears. The enclosed gearbox also keeps out all water and dust from the inside of the gearbox.

By designing and fabricating our own gear boxes, we were able to select an optimal gear ratio for the target application given the motors we had selected. We chose a ratio of 30.38:1; the motor and gearbox combination allows the robot to reach a top speed of 5.07 mph and climb the required 15 degree gradient ramp with ease. All calculations assume an efficiency of 81%.

Suspension

This year’s model does not include a suspension on the drivetrain. Past experiences at IGVC have shown that the terrain does not warrant the extra design complication or safety concern a suspension would cause. While a suspension may smooth out a LiDAR reading or IMU readings, we decided the benefit was too marginal as it has never been a problem in the past.

Weather Proofing

Dokalman MK2 can operate in all but the most severe weather conditions. With an assumed rating of IP53, Dokalman will be able to operate as any negative conditions we expect to experience at the competition that will come from dust and pollen or heavy rain storms. The tight fitting acrylic panels lining the sides of the robot are fitted with weather proofing foam to keep dust and water out of the inside of the robot. Around the top rim of the robot, the top panel is sealed using a rubber weatherstrip which is compressed forming a seal when the panel is in the closed position. The hinge of the panel is made of solid plastic to prevent water from entering any cracks a conventional hinge may have. The top and bottom panels themselves are bent and seamlessly welded, preventing ingress of water or dust. Any holes in the top or bottom panels are covered with a grommet or a 3D- printed part that has a rubber sealing.

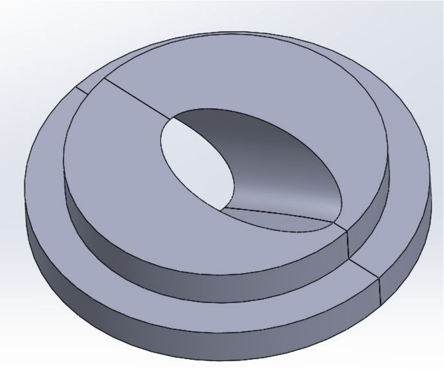

Figure 3 sample of custom grommet allow weatherproof wire passthrough

As an additional precaution, the electrical panels within the robot are raised off the bottom panel. If any water was to enter the robot, it would collect at the bottom of the robot and hopefully avoid the rest of the electronics.

Parts outside of the panels are protected by enclosures or are water-resistant on their own. The GPS antennas are IP69K rated, LIDAR is IP67 rated, wireless antenna is IP64 rated, and Ethernet and power jacks are IP65 rated. As IP ratings for systems are defined by their lowest rated item, and given we have assumed the rest of our assembly has a rating of IP53, we conclude that the entire robot as a system has a rating of IP53.

Electronic and Power Design

Overview

Dokalman MK2's primary power source is a single 100 amp hour 12 volt battery that provides 3.3 hours of navigation. Power for the robot is controlled by a 120A resettable fuse which will trip in the case of a short or other abnormally high power event. Off this fuse, the Sabertooth motor controller power is gated by a 120A relay controlled by e-stop circuitry. The e-stop is controlled both by the hardware button as well as a custom coprocessor circuit board. Auxiliary power for the robot for sensors, computers, and other electrical circuits is first passed through a networked RIGrunner, then either regulated or boosted to the required voltage.

Power Distribution

The power distribution for the robot is simpler than in the past. The entire robot is driven from a single 12V 100AH deep-cycle lithium iron phosphate battery. As mentioned in the overview, power is passed through a 120A resettable fuse for the entire robot. Power for the motor controllers is passed through the 120A emergency stop relay, which is powered closed by the emergency stop circuit. Power for the rest of the robot is provided by a network controlled RIGrunner power distribution unit. The RIGrunner provides resettable fusing and monitorable voltage and amperage on each channel. A voltage rail of 5V via a voltage step-down regulator is provided for the Raspberry Pi and coprocessor circuitry to use. A voltage rail of 19V is provided by a step-up regulator for the Intel NUC and the Jetson TX2. A voltage rail of 9V is provided by a step-down regulator for the network switch. A voltage rail of 24V is provided by a step-up regulator for the PoE injector, required for the wireless access point mounted at the back of the robot.

Electronics Layout

The electronics for the robot can be segmented into three logical subsystems: the hardware interface subsystem; the vision subsystem; and the control subsystem.

The hardware interface subsystem focuses on providing a bridge from the ROS controlled logic of the robot to actual hardware interaction. This subsystem is controlled by a Raspberry Pi with a custom PCB mounted on its GPIO expansion header with an Arduino based Adafruit LoRa Feather M0 (referred to as the coprocessor) mounted on top of the PCB. The Raspberry Pi conveys commands to the coprocessor over a shared UART channel on the Pi's GPIO header. The coprocessor handles many aspects of the robot, including e-stop logic, remote e-stop management, driving the status LEDs, and collecting joystick input from the Spektrum receiver also mounted in this subsystem. The Raspberry Pi also communicates with the Kangaroo x2 motor controller over a similar packet UART system. The Kangaroo x2 receives encoder feedback from the wheels and in turn, controls the Sabertooth 2x60A motor controller.

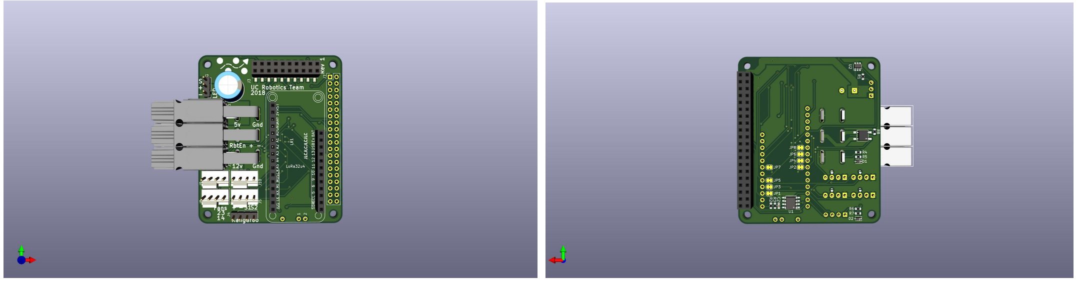

Figure 4 Raspberry Pi Coprocessor Board

The vision subsystem handles processing the bulk vision data and distilling it down to information that is more usable by the control subsystem. It is made up of an NVIDIA Jetson TX2 module along with two wide angle, wide dynamic range USB3.0 cameras. Wide dynamic range increases the camera's ability to both capture as well as process images with high amounts of lighting differences within the same frame. Combined with the wide-angle lens, the cameras are able to generate images which cover lots of ground but also are better at dealing with both the sunny and shadowy sections of the course. The NVIDIA Jetson TX2 is able to handle the full bandwidth of both of these cameras, processing the image data using CUDA pipelines.

The control subsystem communicates with both the hardware interface subsystem as well as the vision subsystem. The control subsystem is centralized by an Intel NUC which performs the major navigation processing for the robot. Over USB, the NUC processes data from a Razor 9-DOF IMU unit and differential Novatel GPS system. The differential GPS system not only provides the robot's latitude and longitude, but also its heading with a positional accuracy of less than 30 cm. The additional heading information is more accurate than a magnetometer's north reading, giving the robot a more accurate heading. The LiDAR on the top of the robot is attached to the NUC via network.

The NVIDIA Jetson TX2, Raspberry Pi, and Intel NUC are all connected to each other over the robot's shared 1000BASE-T network allowing for lots of information to be quickly shared between them.

Safety

The robot is designed with safety in mind. All electronics are connected through a 120A circuit breaker which will trip if a short or other high-amperage event occurs. This will disconnect the entire robot from the battery, preventing further damage to the battery or the robot.

The coprocessor receives emergency-stop information from both the emergency-stop remote as well as the state management software. Both emergency-stop sources need to agree to allow the robot to move in order for the coprocessor to complete the emergency stop circuit, powering the motor controller relay.

The coprocessor waits for an authorization from the state management software to reset the emergency stop condition. This authorization may be revoked later by the state management software when movement is no longer required. This prevents the robot from moving erroneously, even if all other emergency stop sources indicate authorization to move.

The coprocessor communicates with the remote emergency stop device through beacon and response transactions - first, the coprocessor sends out a beacon that it is still alive and waiting for emergency stop authorizations. The remote then, in turn, responds with its current state. If the remote emergency stop responds with an emergency stop condition or does not respond within 500ms, the coprocessor issues an emergency stop, disconnecting the motor controller from power.

In series with the coprocessor emergency stop relay is the hardware emergency stop button. When pressed, the button will open the emergency stop circuit, which in turn opens the 120A emergency stop relay powering the motor controller. Even if the coprocessor for whatever reason fails, the hardware emergency stop will always stop the robot.

Software Design

Overview

All processing nodes on the robot are managed and connect through the Robot Operating System (ROS) framework. This allows us to modularize different software components as well as borrowing proven components from other authors. These nodes communicate with each other via ROS communication primitives, consisting of publish-subscribe (ROS topics) and remote procedure calls (ROS services). These nodes run on either the Intel NUC, NVIDIA Jetson TX2, or the Raspberry Pi. The nuc handles localization and path planning, and the Jetson TX2 handles vision processing. Finally, interfacing between the hardware and coprocessor is handled by the Raspberry Pi.

Localization

The robot uses the Razor 9-DOF IMU, the Novatel differential GPS, and the wheel encoder odometry feedback information to localize the robot to the world. The GPS acts as an absolute location source for latitude, longitude, and heading information. The IMU and wheel encoder feedback are utilized for relative position estimation. All of these sensors inputs are fused together using an extended Kalman filter which provides a high confidence estimate of the robot's absolute position from the mixed confidence inputs. We have found great success in using the EKF in the past, and current tests continue to show incredibly consistent results.

The robot also utilizes the information gathered from the LiDAR for use with localizing the robot with respect to its surroundings, utilizing the SLAM algorithm, specifically implemented by OpenSlam's Gmapping library.

Path Generation

The robot creates two maps of the area around the robot: one "local planner" map and one "global planner" map. These maps consist of a grid of costs which encourage or discourage the dynamic window approach (DWA) algorithm which determines the plan of the robot going forward in time. The two maps only vary in the amount of detail tracked. The global map tracks costs at a resolution of 20 centimeters square, while the local map tracks resolution at 5 centimeters square. DWA computes both a global and local plan. The local plan attempts to stay as close as possible to the global plan, which is able to track farther into the future due to the decreased resolution. This provides us with a high performance algorithm that is still able to plan far into the future at a lower resolution.

Barrel Avoidance

The robot utilizes a Hokoyu LiDAR sensor on the top of the robot for creating a two dimensional planar radial plot modeled in ROS as a LaserScan message with an origin at the LiDAR of obstacles at the height of the LiDAR. These obstacles are represented as intensely high cost points within the cost maps with decreasing cost moving out from the edge of the obstacle. This encourages the path generation algorithm to avoid these obstacles.

Lane Detection and White Line Avoidance

The robot has two cameras located in the front two sides of the robot. These cameras feed into the Jetson TX2 which applies a variety of filters to the image data. The image is thresholded and filtered using a set of convolution matrices for use in edge detection. The vision pipeline does not look for straight lines or relationship between the lines at this stage, rather stores all the edges inside of a point cloud to pass to other nodes. Next, points are stripped away depending on their apparent visibility to the LiDAR; points that appear to be behind obstacles are removed as they are likely erroneous. Finally, the points are fed into a lane detection algorithm. The algorithm is implemented as a costmap layer within the local and global costmaps. The layer uses cost analysis to stay within the same lane it started in, raising the cost as the cells get closer to the detected lane lines and raises the cost incredibly outside of the detected lanes. Remembering where the previous lane is located, the algorithm is able to better predict the location of the lane in future sensor readings.

Failure Handling

Vehicle Failure Modes

Time Desynchronization

Since the robot has multiple computers, the robot thus has multiple different time authorities, one for each computer. Many calculations occur across these time authority boundaries and assume the time authorities are in sync. The robot does run a time daemon, however there have been instances in the past where the time has been out of sync, either by drifting out or not being updated in the first place. Time desynchronization will cause calculations by the robot to either be incorrect (due to being in the wrong time domain) or simply fail (since lookups / other information may not be available to perform the operation). Usually both cases are easily identified by vastly incorrect navigation behavior, and are fixed by forcing a manual time sync between the computers.

Communication Failure

The robot’s main communication bus is the IP network over Ethernet. It is not out of the question (since we use common, off-the-shelf components) that network adapters or other components within one of the computers fails that prevents the computer from communicating with other computers. There are a few links that can be evaluated: between the Raspberry Pi and Intel NUC, between the Intel NUC and the LiDAR, and between the Intel NUC and Jetson TX2. In the case of the Raspberry Pi and Intel NUC, the robot will cease driving. All drive commands (from any mode) pass through the NUC for processing which are then forwarded along to the Raspberry Pi to actually act on the command. If this communication is interrupted, the commands will stop and the Raspberry Pi will time out the movement option. Between the Intel NUC and Jetson TX2, vision processing will halt. This may cause the robot to behave erratically, as it will no longer have the input of the vision subsystem when routing in autonomous mode. If the communication between the LiDAR and the Intel NUC cease, the robot will stop seeing physical obstacles, and will route its plan ignoring new obstacles which were not previously seen. This can be alleviated by having a communications watchdog ensure that messages from both the Jetson and LiDAR are received in a timely fashion, and raising an emergency stop condition.

Coprocessor Total Failure

There may be untested condition that causes the real-time coprocessor to fail. Failure conditions may either be a complete processor reset or a communications desynchronization between the coprocessor and Raspberry Pi. In the first case, the coprocessor will reset to the beginning of its execution, causing the emergency stop to activate as it was written fail-open.

This will stop the robot's movement. In the case of a communication desynchronization or other communications failure, the coprocessor will time out after a set period of not receiving messages and intentionally cause a complete reset, again causing the emergency stop to activate.

E-stop Remote or Drive Remote Failure

The emergency-stop remote or the drive remote may fail at any time. Most common cases of this are due to uncharged batteries. In the case of the emergency stop remote, the coprocessor beacon will time out, and the emergency stop will be triggered. In the case of the drive remote, the coprocessor will recognize that the remote has stopped broadcasting, and will instead broadcast that the remote is signaling no movement.

Vehicle Failure Points

Wear Damage

Wear damage is inevitable with a project of this nature, with moving parts and possible harsh conditions. While designing the robot we sought to minimize this in every way possible and have narrowed our main wear damage concerns to the drivetrain and the lid. The drive train is the main concern as enough wear damage will cease movement of the robot. This is minimized by our gearbox which we have packed with grease to limit gear wear and is easily disassembled to replace worn gears. Due to weatherproofing concerns we implemented a plastic hinge to seal the top from water. Unfortunately, this plastic hinge will eventually wear out requiring replacement to ensure the inside stays waterproof.

Vibrations

Vibrations are a natural consequence of navigating a course over imperfect terrain. To limit any issues that may arise from vibrations we utilized rivets to join frame members which will not loosen due to vibration. Any parts that are attached to the frame may need to be removed frequently so rivets are not an option, therefore we properly torque bolts with Nyloc nuts to the limit probability of vibration loosening critical components.

Simulation

Vision Simulation

To test vision algorithms created by team members, the team evaluated the performance of the algorithms by running it against previously captured vision samples from previous trips to Oakland University. This allowed us to test the algorithms against conditions which were seen at previous IGVC competitions. Visualizers such as rviz and rqt (both ROS visualization tools) allowed us to analyze both inputs and outputs of the algorithm without needing to create a replica course or using bad virtual approximations.

Mechanical Simulation

Finite element analysis was used to confirm the strength of our robot frame. The base pan is the part of the frame that takes most of the load. The pan is constructed of quarter-inch 6061 aluminum. The wheel and caster mounting points were assumed to be fixed and a 120lbf load was applied. This will simulate all the parts of the robot as well as the payload. The max deflection was found to be 0.007097in with a max von mises stress of 2616psi. We wanted a factor of safety of at least 3 and it was found to have a factor of safety higher than 5. The below figure shows the deflection of the frame with a scale of 507 times the actual deflection.

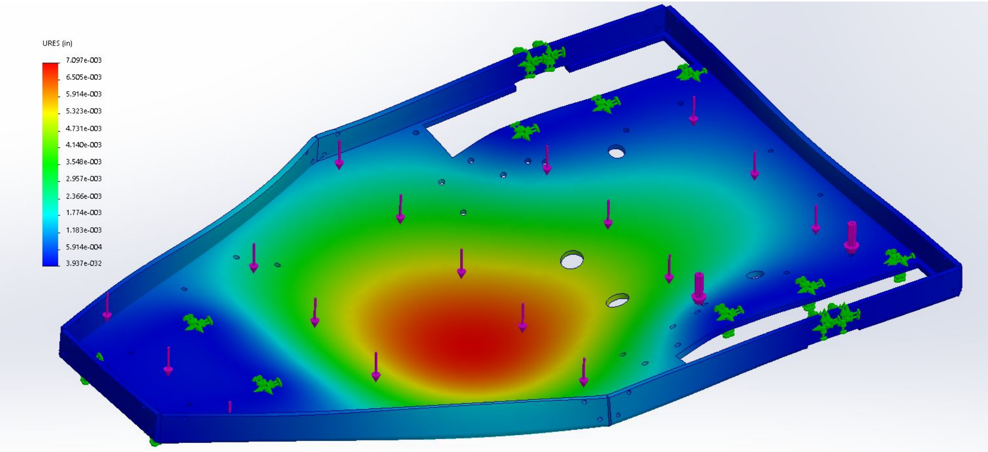

Figure 5 Deflection of base pan with 120lbf load and scale of 507 actual deflection

Performance Testing

All critical modules on the robot were tested independently of each other. In addition, the robot also underwent complete system tests, validating the design of the robot as a whole.

Subsystem Tests

Gearbox and Drive System

The gearbox drive system combination was bench tested at full amperage and voltage drive capacity for the motors which were selected under no load. Both gearboxes exhibited similar performance, thus were appropriate for use on the robot in a differential drive configuration. The attached encoder for each gearbox reported relative revolutions to an acceptable tolerance.

Emergency Stop System

The emergency stop system was tested for the various cases mentioned within the safety section. In each case, the coprocessor board opened the e-stop signal line, causing the robot to stop within the required one second reaction time. Adding the actual safety relay, we observed that the relay was only closed when the entire e-stop signal line was also closed, thus showing that either the coprocessor or hardware e-stop button will be able to reliably stop the robot within the required one second reaction time.

GPS System

The GPS was validated via outdoor test, validating that in multiple locations the longitude, latitude, and heading reported were well within tolerance (less than 30 cm accuracy).

LiDAR System

The LiDAR system was tested with an assortment of application specific obstacles (construction barrels). The LiDAR was able to properly sense the location and size of the barrels at a distance up to 30 m. The LiDAR was also able to sense non-application obstacles, such as people, walls, as well as other opaque obstacles. This increases the safety of the robot as it will try to avoid striking obstacles in general.

Vision System

The vision system was tested with white lines painted on a grass field. Due to the testing areas available to the team, we were unable to paint lines or lanes of our own. The vision system, with its rudimentary line processing, was able to report line positions with acceptable accuracy (with acceptable noise levels) at a distance up to 5 m.

Full System Tests

RF Interference

Many parts of the robot utilize RF for some part of their operation, specifically the wireless access point, the e-stop remote, the remote drive receiver, and the GPS. Other devices also give off EMF which may disrupt the performance of other devices. The robot was inspected with a software defined radio around frequencies of interest to validate that two devices would minimally interfere with each other. When interference conditions were found, devices were moved in order to bring the interference back down to an acceptable level.

Network and Communication

The entire network was tested for connectivity and bandwidth to ensure separate systems were able to communicate information at the speeds required by the robot while in autonomous mode. Tests showed that the robot was able to sustain expected levels of traffic with acceptable transmission latency.

Performance Assessment

To date, the robot is able to properly route a path towards a goal, avoiding both lines painted on grass as well as barrels and other objects standing in the way of the robot's goal. The robot is able to navigate to GPS based goals within 0.6 meters of the center of the robot, functionally hitting the target waypoints. Due to the lack of availability of a proper testing area, obstacles such as fences or potholes have not been tested. The route planning loop occurs at a 20Hz rate, so the robot's reaction time is limited by the acceleration and deceleration able to be safely provided by the drivetrain system. Autonomous navigation achieves speeds of around 1 m/s when in a relatively straight path. Obstacles can be added to the navigation map as soon as they are detected. The LiDAR is able to detect obstacles 30 meters away with a 270 degree horizontal field of view.

Appendix A: Bill of Materials

Part	Manu	Model No.	Quanity	Unit Price	Total
Battery	Smart Battery	SB100	1	1300	1300
Motor Driver	Dimension Engineering	TE-091-260	1	190	190
Processing Computers	Intel	NUC8i7BEH	1	634.99	634.99
Motion controller	Dimension Engineering	X2	1	23.99	23.99
GPS	Novatel	FlexPak6D and antenna	1	3600	3600
Lidar	Hokuyo	UTM-30LX-EW	1	5200	5200
Rigrunner	West Mountain Radio	4005i	1	280	280
Wheel Encoder	CUI Inc	amt102v	2	25	50
Raspberry Pi	Raspberry Pi	3B	1	35	35
Relay	ARTGEAR	E193	1	12.25	12.25
12 to 19v Conv	Aweking	d20161223xj0120	2	19.99	39.98
Gearbox Parts	Vex Robotics	Various	1	415.96	415.96
Motors	Vex Robotics	217-2000	4	32.99	131.96
Camera	WebCamera_USB	SUSB1080P01	2	64.99	129.98
Camera Lens	Vicdozia	GPLS0012	3	11.95	35.85
Signal Light	BTF-LIGHTING	WS2812b	1	32.88	32.88
Coprocessor Parts	Various	Various	1	53.50	53.50
E-stop Housing	Hasbro	Bop-it!	1	15	15
Vision Processing Unit	Nvidia	Jetson TX2	1	300	300
Network Switch	TP-Link	TL-SG108	1	19.99	19.99
12v to 5v Converter	TOBSUN	EA50-5V	1	9.88	9.88
Body Panels	Flood Heliarc	None	1	1500	1500
12v to 24v Converter	E-KYLIN	INT-12T24-10A	1	14.99	14.99
IMU	Spark Fun	MPU-9250	1	34.99	34.99
				Total	14061.19

Acknowledgements

We would like to thank those who have supported our team and without whom none of this would be possible.

- Dr. Janet Dong (advisor),
- Flood Heliarc (sponsor),
- Procter & Gamble (sponsor),
- Northrop Grumman (sponsor), and, of course
- The University of Cincinnati