Dokalman MK2.5

University of Cincinnati

30th Annual Intelligent Ground Vehicle Competition

Andrew Gerstenslager ✦ Michael Mcadams ✦ Colin Campbell ✦ Prateek Kharangate ✦ J Roth ✦ Andrew Milos ✦ Petar Acimovic ✦ Long Pham ✦ Dhruv Bangad ✦ Luke Sferrella ✦ Alex Suer ✦ Sam Moeller

Contact information for team members provided in Team Organization section

Submitted 5/15/23

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 05/15/2023

Doctor Janet Dong, Advisor

Introduction 3
Team Organization 3
Design Process/Assumptions 3
Innovative Concepts and Technologies 4
Mechanical Design 5
  Overview 5
  Frame Structure, Housing, and Structure Design 5
  Suspension 5
  Weather Proofing 6
Electronic and Power Design 6
  Overview 6
  Power Distribution System 6
  Electronics Layout 7
  Safety Devices 7
Software Strategies and Mapping Techniques 8
  Overview 8
  Obstacle Detection and Avoidance 8
  Software Strategy 8
  Path Planning 8
  Map Generation 8
  Goal Selection 8
  Additional Creative Concepts 8
  Path Generation 8
Failure Handling 9
  Vehicle Failure Modes 9
  Vehicle Failure Points 10
  Vehicle Failure Prevention Strategy 10
  Testing 10
  Vehicle safety design concepts 10
Simulation 10
  Simulations in virtual environment 10
Performance Testing 11
  Component Testing 11
  Subsystem Tests 12
  Full System Tests 12
Performance Assessment 12
Appendix A: Bill of Materials 12
Acknowledgements 13

Introduction

The University of Cincinnati Robotics Team's proud tradition of competing in the Intelligent Ground Vehicle Competition will be restarted after a 4 year break due to covid complications and a complete rewrite of the software package with the goal of upgrading from ROS1 to ROS2 as ROS1 is no longer receiving supported updates from the developers. Since 2021, our team has had to reconstruct a lot of the networking, software, hardware, and navigation strategies from scratch as many veteran members graduated and took their knowledge without a lot of proper documentation to refer to. This can be seen especially with the coprocessor board that was in the process of a rework in 2019. Due to complications of that project leading into the Covid-19 lockdown, the board was never finished with work to still do with no guidance leaving the team to scrap and redo that functionality. Much of our work in the time since the Covid-19 lockdown was to take the existing robot hardware and reconnect all the different sensor/actuation systems into ROS2 nodes and rebuild out the robot's functionality.

Team Organization

Role	**Name**	**Major**	**Email***
President	Andrew Gerstenslager	Computer Engineering	gerstead@mail.uc.edu
Vice President	Colin Campbell	Computer Science	campb3c7@mail.uc.edu
Treasurer	Michael McAdams	Electrical Engineering	mcadammt@mail.uc.edu
Secretary	Dhruv Bangad	Computer Engineering	bangaddj@mail.uc.edu
Member	Prateek Kharangate	Computer Science	kharanpv@mail.uc.edu
Member	J Roth	Electrical Engineering	rothsj@mail.uc.edu
Member	Long Pham	Computer Engineering	pham2to@mail.uc.edu
Member	Andrew Milos	Computer Science	milosam@mail.uc.edu
Member	Petar Acimovic	Electrical Engineering	acimovpr@mail.uc.edu
Member	Sam Moeller	Electrical Engineering	moellesu@mail.uc.edu
Member	Luke Sferrella	Mechanical Engineering	sferrelj@mail.uc.edu

Design Process/Assumptions

The Dokalman MK2.5 design process was focused on rebuilding the systems so we understood how they worked and had better control over them. Our team has made improvements in these various areas of the robot.

Hardware: With the hardware, we replaced the raspberry pi along with the coprocessor board with an Arduino Mega. The arduino controls the state of the robot (autonomous, manual drive, or stopped) by listening to a RC Receiver governed by a controller. The Arduino receives serial communication from the PI onboard and then drives the motors appropriately for autonomous navigation mode and sends back updates as needed.

Mechanical: We connected a new Velodyne lidar to Dokalman. We designed a new mount for the lidar so it is fixed where the old lidar was located.

Electrical: Since the remote hardware E-Stop was located on a coprocessor for the PI previously, we purchased a remote relay to interpose between the battery and the motors to allow for the robot to be stopped easily.

Vision: We built reusable pipelines for our camera systems. Once we run the proper calibration script, we can then reuse the ROS2 nodes for any number of cameras in the future.

Networking: We had to redefine the networking so all of the devices could communicate properly within the network as well as reconnecting and updating our base station so we can perform remote access and development.

Software: A large portion of our work has been re-integrating systems back into ROS2 from ROS1. A lot of the packages and code could not be reused. We have built nodes for the large portion of the hardware. This includes finding the Velodyne Lidar Package for the LaserScan message, building the vision pipeline, and reconnecting the GPS and Arduino to new ROS2 nodes.

The only assumption that we made was that the robot frame/construction was more than sufficient for our use. It was designed by a team of mechanical engineers who rigorously tested the structural integrity with software and after construction. Because of that, we did not make any major modifications to the design of the frame, panels, gearboxes, or drive system.

Innovative Concepts and Technologies

Dokalman MK2.5 features ROS2 as the core system driving all of its components instead of ROS1. With ROS2, we are leveraging the new Nav2 package that contains useful tools such as waypoint navigation, slam mapping, and obstacle avoidance all in one ecosystem with the benefit of using newer hardware such as our Velodyne LiDAR that supports ROS2. Additionally ROS2 removes the need for a central computer continuing the roscore process which is a central process to one computer. ROS2 provides a way to communicate without requiring master nodes.

Our camera system also allows us to calibrate and connect as many cameras as we can process by having reusable ROS2 node pipelines. In the future, we will have the ability to quickly/efficiently add more cameras to detect the ground in up to a full 360 degree view. Doing so will improve our line detection accuracy as we can see more of the ground at a time from the robot and we will have less chance of missing any important data..

The Dokalman MK2.5, maintaining the reengineered frame and custom gearboxes of its predecessor, optimizes space for integral components while meeting competition size limits. Offering robust power and speed for competitive performance, it also boasts an IP53 rating, ensuring reliable operations in potentially hazardous conditions.

The base station is an auxiliary robot utilized for monitoring and developing on Dokalman MK2.5. This additional robot contains its own router and a wifi access point. Both Dokalman and the base station have Bullet M2 devices, which allow network traffic to pass through freely over radio transmission which allows us at up to 500+ feet away to monitor ROS2 messages, start/stop ROS2 processes, or program directly on each device on Dokalman utilizing both SSH and remote servers. The base station also has the ability to add WAN access anywhere utilizing a 5G router which provides internet access to remote PCs as well as any device inside Dokalman. This allows package installation on the fly or commits/pulls from the central code repository from any location.

Mechanical Design

Overview

The hardware configuration of Dokalman MK2 remains unchanged from 2019, maintaining the same components and design elements as in the previous version.

Dokalman's elements consist of four main sections: the drive platform; frame; hinged top panel; and the electronic panels. The drive platform consists of the bottom tray, two gearboxes, and the front caster wheel assembly. The top panel acts as a weatherproofing shield around the drive platform and electronics panels while also allowing quick access to the robot's interior through an electronic latch. 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 lidar assembly of the robot.

Frame Structure, Housing, and Structure Design

The drive platform consists of two panels which act as a lid/rim on top and a pan on the bottom. Both of the panels were laser cut and bent to shape. The framing of the robot is created from laser cut aluminum struts attached by rivets to the pan and rim. The top panel was 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 are secured with screws to the frame.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 account for heat in a closed system, two fans located at the back of the robot act as an air intake and exhaust to help circulate fresh air through the system.

Suspension

With the rugged design of the gearboxes and robot assembly, we do not have a suspension system on the robot. Because of this we do remove a failure point on the robot. As the competition is held on pavement this year as well the terrain does not warrant the need for increased suspension. Though adding suspension could help with the accuracy of the camera line detection systems and LiDAR readings, we think any negative effect is marginal.

Weather Proofing

The Dokalman MK2.5 is capable of functioning in almost all types of adverse weather scenarios, barring the most extreme. Its projected IP53 rating suggests that it should perform seamlessly, even when subjected to harsh conditions expected at the upcoming competition, which may range from dust and pollen contamination to heavy rainfall.

The top circumference of the robot features a sealed top panel, achieved by compressing a rubber weatherstrip, creating a secure seal when the panel is shut. Instead of a conventional hinge that may allow water seepage through crevices, the panel's hinge is constructed from solid plastic for enhanced protection.

The top and bottom panels are bent, and welded seamlessly to prevent the penetration of water or dust. Any apertures present in these panels are equipped with a grommet or sealed in other ways.

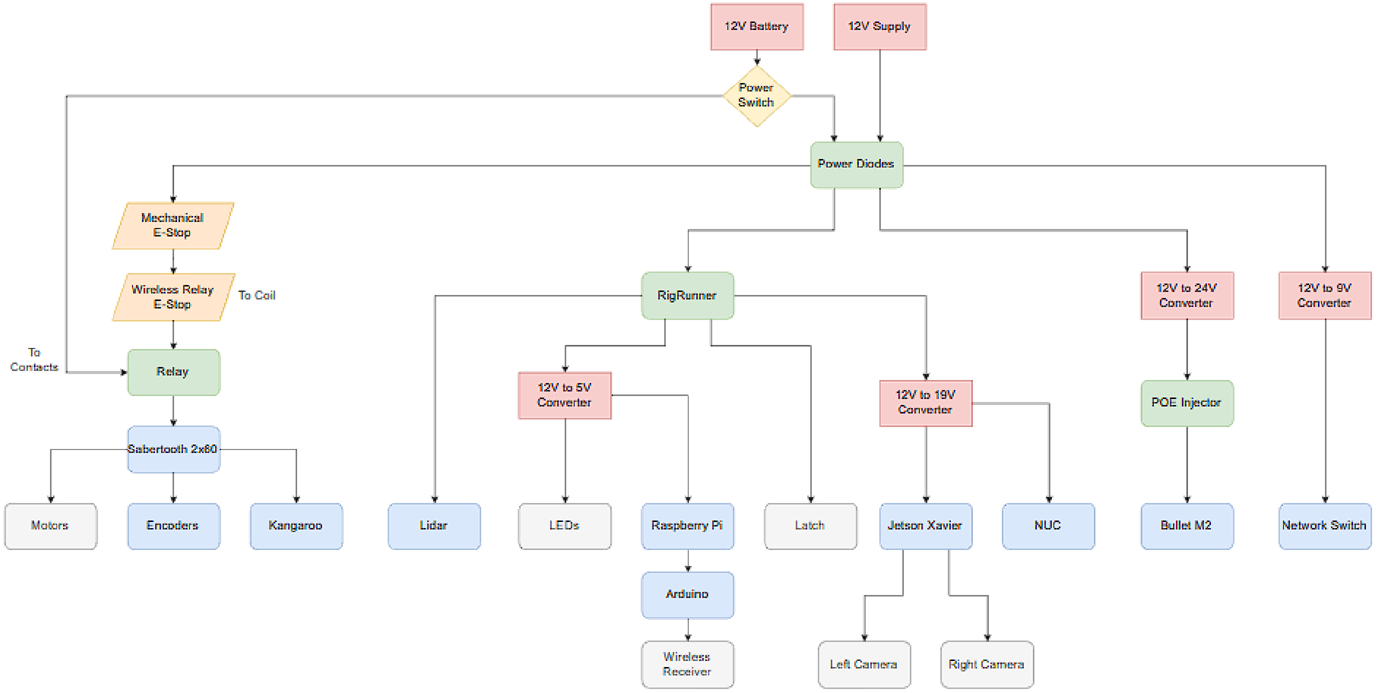
For added security, the robot's interior electrical panels are elevated from the base panel. This design ensures that in the event of any water intrusion, the liquid would accumulate at the bottom, thus minimizing the potential damage to the electronic components.

Electronic and Power Design

Overview

Dokalman MK2.5’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 circuitry consists of a 120A relay controlled by the large e-stop button on the top of the robot as well as an additional remote relay. 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. Some of the components are not passed through the rigrunner.

A secondary 12VDC power source may be used to power Dokalman’s electronics when it is not being driven to save charge on the battery. This method does not power the motors, keeping Dokalman stationary while the network and sensors can be worked on. Below is the basic power diagram for Dokalman.



Power Distribution System

The complete robot operates on a single 12V 100AH deep-cycle lithium iron phosphate battery. From the battery, a 120A resettable fuse, as part of the power switch, is utilized to distribute power and prevent current overdraw. Power is distributed throughout the robot in several ways:

1. The motors and motor driver receive power through the emergency stop system directly from the battery. The 120A fuze prevents extreme current draw situations.
2. Most of the onboard devices receive power through the network-enabled RIGrunner power distribution unit. The RIGrunner has current protection and can provide power over 5 separate ports. Voltage and current use can be monitored over each channel.
3. The remaining components have their own 12V converters that receive power from the main supply through the diodes.

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.

At the heart of the hardware interface subsystem is an Arduino Mega and a Raspberry Pi. The Arduino is connected to the Raspberry Pi over USB which allows serial communication to interface the Arduino to ROS2 as the Pi relays any important commands to the Arduino.

The Arduino determines state control from a RC receiver that is bound to a controller. The states of the Arduino signify remote control, self drive, and stopped statuses. In the remote control state, the robot can be driven normally with a controller and the robot ignores any serial commands from the PI to the Arduno. The self drive mode is the opposite of that state where the Arduino only accepts commands from the serial port and ignores any remote control input. The stopped state actively sends commands to the motor controller to set it to the stopped state. Toggling a tri-state switch on the RC Controller allows for seamless transitions between these modes. The Arduino also changes LED indicator lights depending on the stage of the setup process and then the current driving state determined by the controller.

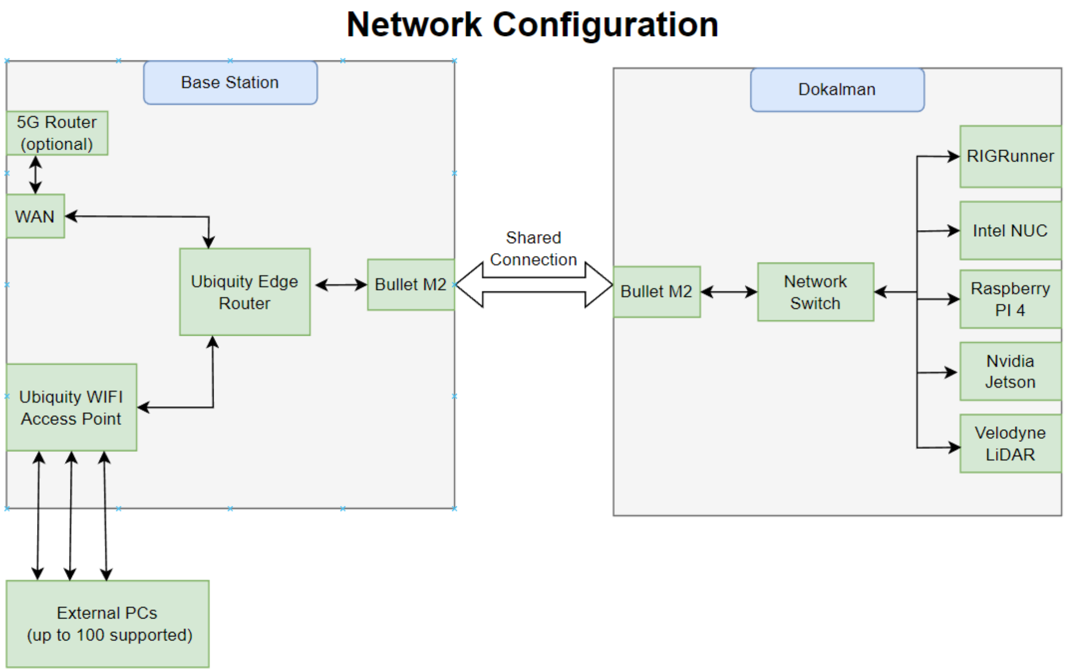
The Raspberry Pi handles all of the serial communication. As stated before, the Pi communicates to the Arduino. Additionally, the Pi communicates to the Novatel GPS over serial to retrieve logging for position and heading values as well as the onboard IMU sensor. The LiDAR is connected to the network where its messages are available through ROS2.

The vision subsystem plays a crucial role in handling and processing the vast amount of visual data, refining it into more actionable information for the control subsystem. This subsystem consists of an NVIDIA Jetson Xavier module accompanied by two wide-angle USB3.0 cameras equipped with wide dynamic range capabilities. The inclusion of wide dynamic range enhances the cameras' ability to capture and process images containing significant variations in lighting conditions within a single frame. The combination of wide-angle lenses and wide dynamic range enables the cameras to capture extensive areas while effectively managing both brightly lit and shadowy sections of the course.To effectively manage the data, the NVIDIA Jetson Xavier module efficiently processes the full bandwidth of both cameras. This processing is achieved using C++ OpenCV image processing techniques, leveraging the power and capabilities of the Jetson Xavier platform.

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. The implementation for navigation systems are still in the early stages, but the NUC will handle all SLAM mapping, path planning, and actuator state updates to follow the optimal path.

Network

All devices under Dokalman share the same 10.0.0.* subnet. Through this subnet, ROS2 messages are communicated between devices. This network is extended through the base station where any computer connecting to its wifi network is automatically assigned an IP address in that same subnet by the router inside the base station. Below is the complete network configuration of Dokalman and the base station



Safety Devices

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 RIGrunner described later is our main power delivery system that has overdraw protection for all devices that are powered by it.

The e-stop system, improved substantially since 2019, incorporates a multi-layered approach to ensure safety and control. The most visible element of this system is a large red button located on top of the robot. When pressed, it activates a 120A relay that immediately cuts off all power to the motor controller.

In addition to this, a remote relay, connected to the first, provides the ability to halt the motors from a distance of over 100 feet. This distance control is further facilitated by the RC controller, which is designed to send cease commands to the motors with each cycle of its primary loop when activated. Moreover, the system is designed to react swiftly when the RC controller/receiver is not detected by the Arduino. In such an event, the robot will instantaneously dispatch stop orders to the motors, reboot the RC receiver, and switch to pairing mode to mitigate any risk of uncontrolled movement.

During self-drive mode, we have the capability to send ROS2 messages to the drive_speed and turn_speed through the network, allowing us to halt the robot. This functionality can be operated remotely from significant distances, thanks to the base station.

Software Strategies and Mapping Techniques

Overview

All processing nodes on the robot are managed and connected 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 Xavier, or the Raspberry Pi. The nuc handles localization and path planning, and the Jetson Xavier handles vision processing.

Something of important note in reference to this section is that we are still actively in development in this area. We are still working on the navigation as we lost a month of time with no access to our lab, meaning we were forced to wait to implement and test many of Dokalman's sensor systems. This put us behind our projected schedule for the IGVC.

Obstacle Detection and Avoidance

The Velodyne LiDAR allows detection of the construction barrels placed on the course. The LiDAR is placed at the optimal height to detect the traffic cones for avoidance.

Additionally the cameras detect lines on the ground using OpenCV image processing techniques to skew the image to a top-down view then to extract the lines from the image. The detected lines are then converted to a point cloud.

Avoidance for these point clouds is under development.

Software Strategy

All of the code is packaged inside of separate ROS2 packages. This allows all code to be very portable as any selection of nodes can be built/moved on any device. All packages are available in the main git repository and updates are consistently pulled on all devices utilizing the ease to connect to devices remotely with the base station.

Once the packages/nodes are necessary for each device, the appropriate launch file is created to launch those specific nodes from each package. Once the launch file is created, a partner BASH file is also produced that when executed automatically builds (if necessary) and sources the specified packages from their appropriate directories. The launch file can then be launched using ROS2s launch feature.

Path Planning

Still under current development.

Map Generation

Still under current development. Work is being done to implement slam mapping with the LiDAR and camera line point clouds.

Goal Selection

Still under current development. Work is being done to build behavior and state trees to model Dokalman's action planning and goal selection.

Additional Creative Concepts

With Dokalman simulated in RViz, the next step is to fully simulate Dokalman in Gazebo to be able to test navigation changes/development.

Path Generation

Still under current development.

Failure Handling

Vehicle Failure Modes

Time Desynchronization

The robot is equipped with multiple computers, each acting as a distinct time authority. A plethora of calculations are performed across these boundaries, predicated on the synchronization of these time authorities. We understand that time desynchronization can cause a large problem so we plan to implement the ROS2 TimeSynchronizer to keep the camera and LiDAR point clouds from diverging with the camera requiring additional processing to generate the point clouds.

Communication Failure

Given more time, addressing the potential problem with devices becoming disconnected from the system would be developed. ROS2 nodes will operate on each device and the NUC will have a central process that ensures that each required device is outputting status updates periodically to the network. If a device does not respond, then the NUC will stop all navigation and driving utilizing ROS2 services.

E-stop Remote or Drive Remote Failure

If one E-stop method fails to stop the robot, there are multiple systems that allow redundancy to prevent an out of control scenario.

Vehicle Failure Points

Wear Damage

The project design minimizes wear damage, particularly on the drivetrain and lid, which are critical to the robot's functionality and weatherproofing. Wear is mitigated via a grease-packed gearbox and an easily replaceable plastic hinge.

Vibrations

To address vibration issues caused by rough terrain, the robot uses rivets for joining frame members and torqued bolts with Nyloc nuts for attachable parts, reducing the chance of loosening due to vibrations.

Vehicle Failure Prevention Strategy

A robust vehicle failure prevention strategy is essential for maintaining the efficiency and reliability of any automated system. This strategy primarily revolves around proactive maintenance, regular system checks, and effective fail-safe measures. Preventive maintenance is key to mitigating wear-and-tear and ensuring the longevity of the system. This includes periodic inspections and replacement of parts that are prone to damage over time. In addition, regular checks on the system's power supply, such as battery health, can prevent unexpected failures.

Testing

All developments to Dokalman are then checked with the existing systems to ensure that all parts of Dokalman work together and there are no conflicts. Regular checks of already developed systems occur on a semi-frequent basis to test that the performance has remained the same..

Vehicle safety design concepts

The rugged nature of the construction of Dokalman MK2.5 and the redundant e-stop ecosystem help reduce any external damage to Dokalman as well as any damage to the environment around Dokalman as well. Dokalman's bright and clear indicator lights also allow all personnel to easily identify the operating state of Dokalman.

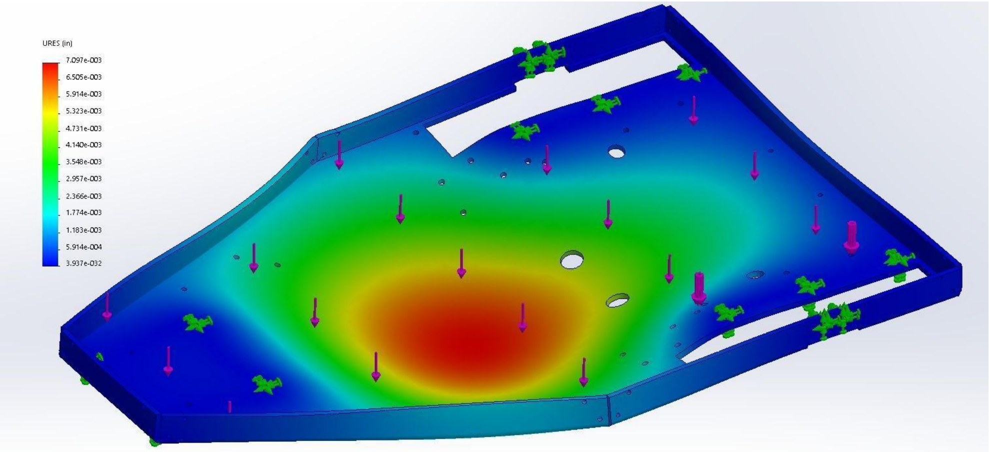
Simulation

Vision Simulation

Using video data from visiting the IGVC in 2022, we were able to capture and save those frames and render them to extract white lines in order to build our image processing pipeline. As stated later, we are still working on implementing virtual cameras in a Gazebo environment using that same pipeline process.

Mechanical Simulation

In 2019 when Dokalman MK2 was constructed, the team aimed for a factor of safety of at least 3, but it was found to be greater than 5. A figure below shows the frame deflection when a large force is exerted to the bottom panel. The results of the testing showed that Dokalman was more than capable of handling large forces and impacts.

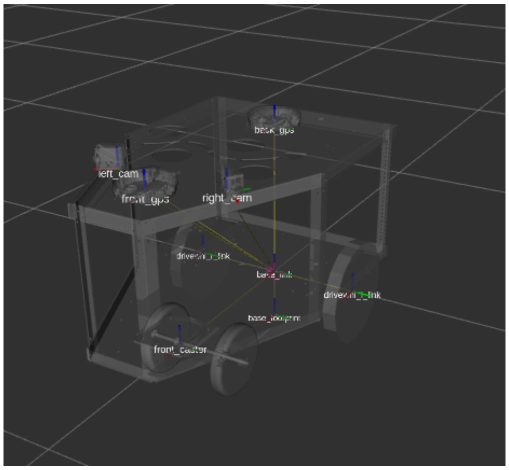


Deflection of base pan with 120lbf load and scale of 507 actual deflection

Theoretical concepts in simulations

URDF - RViz2

Our team is currently working on re-developing the URDF to define the robot's geometry and properties. We are able to load and render our robot in RViz2 and we are still developing the inertial, mass, and collision components so we can convert our URDF into RDF for gazebo to interpret properly. We are working on aligning each sensor's output to this frame to use for navigation and mapping.



Simulated Robot in RVIZ using URDF

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

Each e-stop system was tested thoroughly to ensure that each individual system was able to stop the motors of the robot successfully.

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

Several components of the robot rely on radio frequency (RF) for their functioning, notably the wireless access point, the emergency-stop remote, the remote drive receiver, and the GPS. However, there are also other devices emitting electromagnetic fields (EMF) that could potentially impair the performance of these components.

To address this, we employed a software-defined radio to inspect the robot, focusing specifically on the frequencies of interest. The aim was to ensure minimal interference between devices. In situations where interference was detected, we repositioned the conflicting devices to reduce the interference to an acceptable threshold.

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 has all of its sensor systems operational. The GPS detects location and heading. The LiDAR reports obstacles. The cameras also capture lines on the ground from a top-down perspective. We have started implementing the URDF and all other actions necessary for navigation. Our team lost over a month of time this spring semester due to certain lab complications. Nonetheless, the systems we have implemented in our available time work very well and are robust. After turning in this notebook we plan to keep pursuing the navigation utilizing Nav2 from ROS2’s package ecosystem.

Appendix A: Bill of Materials

Part	Manufacturer	Model No.	Quanity	Unit Price	Total
Battery	Smart Battery	SB100	1	1300	1300

Motor Driver	Dimension Engineering	Sabertooth 2x60	1	190	190
Processing Computers	Intel	NUC8i7BEH	1	634.99	634.99
Motion controller	Dimension Engineering	Kangaroo X2	1	23.99	23.99
GPS	Novatel	FlexPak6D and antenna	1	3600	3600
Lidar	Velodyne	Puck LITE	1	3000	3000
Rigrunner	West Mountain Radio	4005i	1	280	280
Wheel Encoder	CUI Inc	amt102v	2	25	50
Raspberry Pi	Raspberry Pi	4	1	160	160
Relay	ARTGEAR	E193	1	12.25	12.25
Remote Relay	Fusionsea	WT-03	1	20	20
12 to 19v Converter	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
Vision Processing Unit	Nvidia	Jetson Xavier	1	600	600
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
12v to 9v Converter	Chuangruifa	FBA_OT446	1	9.99	9.99
IMU	Spark Fun	MPU-9250	1	34.99	34.99
				Total	12247.68

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),
- The University of Cincinnati