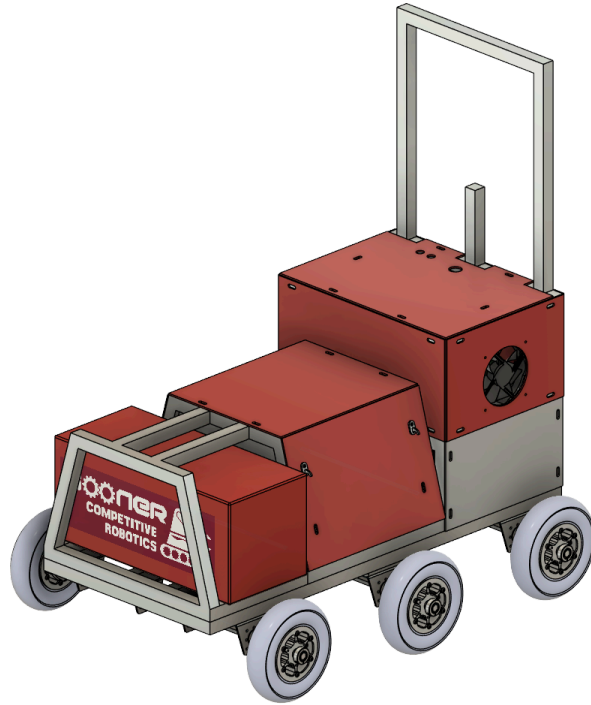


SOONER COMPETITIVE ROBOTICS

The University of Oklahoma
“Danger Zone”

Submitted May 15th, 2024



I hereby certify that the development of the vehicle, The Danger Zone, as described in this report, is equivalent to the work involved in a senior design course.

N Zemlin

Software	Mechanical	Electrical
Dylan Zemlin	Vivian Easley	Michael Falcone
Daniel Brown	Yuhao Zhang	Matthew Gonzales
Antonio Chappell		Brandon Kahn
Abby Roland		Min Kang

Design Process, Team Identification and Organization	1
Introduction	1
Organization	1
Design Process	1
Bill of Materials	2
Effective Innovations	2
Software	2
Electrical	2
Mechanical	3
Description of Mechanical Design	3
Overview	3
Drive Train Analysis	3
Cooling System	3
Improved Accessibility	4
Weatherproofing	4
Description of Electronic and Power Design	4
Overview	4
Power Distribution	5
Motor Control	6
Safety Lights	6
E-Stop and Safety	6
Description of Software Systems	7
Overview	7
Image Transformation	7
U-Net for Computer Vision	8
Particle Filter Localization	9
Goal Selection and Path Planning	9
Motion Generation and Monitoring	10
Configuration and GUI	10
Logging and State Machine	11
Cyber Security Analysis	12
NIST RMF Process	12
Threats	12
Cyber Controls and Implementation	12
Vehicle Analysis	13
Lessons Learned	13
Failures Points	13
Bug Tracking and Version Control	13
Simulation	14
Performance Assessment	14

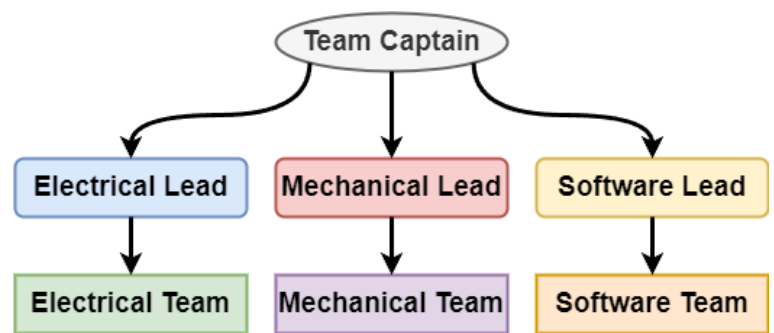
Design Process, Team Identification and Organization

Introduction

Sooner Competitive Robotics, representing the University of Oklahoma, is proud to present the Danger Zone, the team's entry for the 2024 IGVC AutoNav competition. The university faculty advisors are Noah Zemlin, a radar research engineer, and Dr. Golnaz Habibi, an assistant professor with a research focus on robotics and autonomous driving. The Danger Zone brings many innovations and upgrades compared to the team's previous entries into the competition.

Organization

The team is organized into electrical, software, and mechanical sub-teams, each led by an experienced member or upperclassman who guides the team members in completing their tasks. Sub-team leads regularly checked in with their members to ensure tasks were being completed sufficiently and to provide assistance when needed. Sub-team leads report to the team captain to ensure alignment and progress in the robot's design and development. The team has dedicated approximately 1500 hours as of the submission of this design report into vehicle development and research.



Design Process



The team's design process is structured around constant refinement, with continuous iteration and testing of solutions. As shown in the figure on the left, the team begins by defining the high-level problems and then breaking them up into smaller research tasks. After each sub-team conducts their research, the team comes together and designs solutions collaboratively. Once designs, models, and documentation are ready, the team

moves to the build stage, where these ideas are created. Building often spans multiple months, and involves ongoing refinement as solutions are tested. Finally, the solutions are implemented and final testing is performed. If the implementation fails to meet the requirements, the team returns to the problem definition stage. This process enables the team to create robust, innovative robots that can be continuously adjusted based on environmental factors and team conditions.

Bill of Materials

The Danger Zone's total production cost is approximately \$2,500. The bill of materials table below outlines the major components costs of the Danger Zone.

Item	Unit Cost (USD)	Quantity	Team Cost
Intel NUC	\$550.00	1	\$550.00
VN-200 Rugged	\$3,050.00	1	\$0.00
Logitech C920x	\$69.99	2	\$139.98
Drivetrain Gearbox	\$80.00	2	\$160.00
Acrylic Sheets	\$21.99	4	\$87.96
Custom PCBs	\$20.00	5	\$100.00
Pneumatic Wheels	\$70.00	6	\$420.00
12V Batteries	\$116.00	1	\$116.00
SPARK MAX MC	\$90.00	2	\$180.00
NEO Brushless Motor	\$48.00	2	\$96.00
Misc Electrical	\$363.15	1	\$363.15
Misc Mechanical	\$283.58	1	\$283.58
Total Cost	\$5546.67	N/A	\$2496.67

Effective Innovations

Software

A goal for the Danger Zone was to have a wider field-of-view (FOV) without introducing the distortions of wide FOV cameras. To achieve this, a second camera was added to the robot which allows its FOV to be theoretically doubled. Each camera is placed in portrait orientation to create a large and distortion-free view of the course.

Electrical

A new electrical hub was designed to protect individual electrical systems on the robot. Each connection on the hub utilizes a resettable fuse and TVS diode to protect against reverse current and overcurrent conditions. Additionally, each connection features an LED to indicate the status of the

individual subsystems. A buffer IC was added to allow any subsystem to read the E-stop line without any risk of accidentally triggering an emergency stop. The hub along with all the subsystems switched from using a DB9 serial connector to a more compact and space-efficient 6-pin micro-fit Molex connector. The introduction of a solid-state relay and brushless motors has improved the vehicle's overall power efficiency by at least 30% compared to the team's previous robots.

Mechanical

The standout feature of the Danger Zone is the introduction of the six-wheel drop center drive train. All six wheels are driven, with each side of the Danger Zone driven independently. This provides superior performance on asphalt and smooth traversal of the ramp.

Description of Mechanical Design

Overview

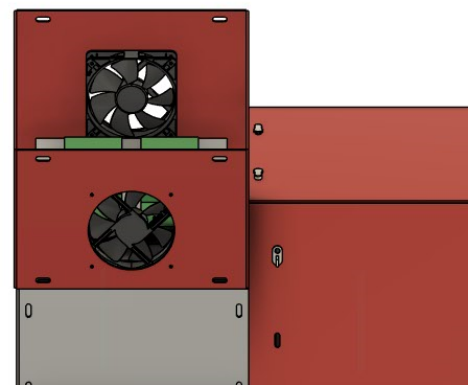
The overall design of the Danger Zone aimed to create something unique and different from the previous years' Rat Van and Weeb Wagon. The largest change in this year's design is the switch from two-wheel center drive with casters into a six-wheel drop center drive. The overall profile of the robot is also slimmer, featuring a trapezoidal profile that leads into the payload at the front. The design also features a fan-cooled electronics compartment with panel doors.

Drive Train Analysis

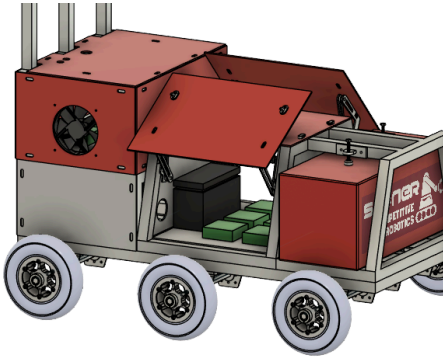
The drive train consists of a total of six wheels driven by two motors. Each side has three wheels connected in series and are driven by a NEO brushless motor. The motor is run through a REV-21-2190 gearbox for a 10.7:1 gear reduction. The gearing was chosen for max efficiency at 4 mph which is the speed achieved during the team's best run in 2023.

Cooling System

A cooling system was implemented to manage the high temperatures faced by the Intel NUC within its weather-proof enclosure. Using two standard 120mm fans, the cooling systems provide constant circulation of fresh air to the primary electronics section. This helps prevent damage to the electronics as well as improves overall computational performance. Not shown in the CAD model on the right are two water-resistant dust covers attached to the external fans, which prevent water, pollen, and dust from entering the enclosure.



Improved Accessibility



Last year, the electronic undercarriage of the robot, which included the motor controller and battery, was difficult to physically access. This year, the design philosophy emphasized ease of accessibility. The Danger Zone features large, easily maneuvered doors that provide access to the electronic undercarriage without limiting the ability to weather-proof the robot.

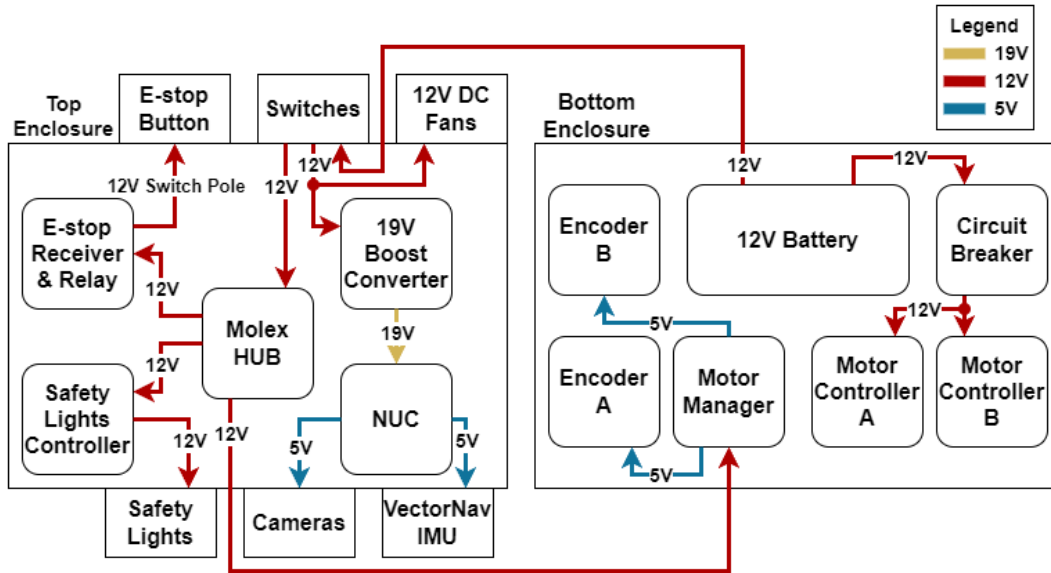
Weatherproofing

Weatherproofing of the robot was improved substantially from last year's entry. Electronic enclosures are now sealed with 1/8 inch acrylic panels and rubber gaskets at the seams for effective weatherproofing. Acrylic panels slotted into the 80/20 chassis are also lined with rubber gaskets to create a seal. Weatherstripping foam tape has been applied to panel doors to prevent water from entering while the panels are closed.

Description of Electronic and Power Design

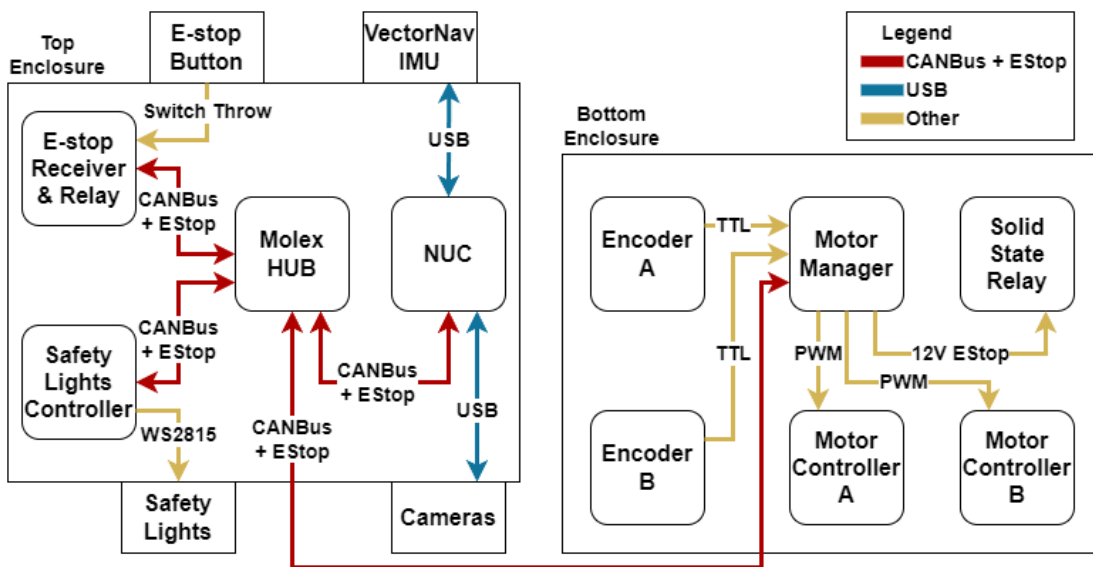
Overview

The Danger Zone has a newly designed electrical system featuring custom PCBs. The custom subsystems include the Molex hub, motor manager, safety lights controller, and E-Stop receiver & relay. The primary processor of the robot is an Intel NUC. The NUC communicates with the electrical subsystems using the CAN bus protocol through custom message definitions. The electrical system has a discrete E-Stop signal that connects all the subsystems through a buffer. The electrical system operates on a 12-volt power source.



Power Distribution

The Danger Zone is powered by a 12-volt lead acid battery chosen for capacity, safety, cost, and availability. The battery has three primary connections: the Molex hub, a 19-volt boost converter, and SPARK MAX motor controllers. The Molex hub powers the three other subsystems. The 19-volt boost converter powers the NUC. The SPARK MAX motor controllers power the brushless motors. The motor controllers are in series with a 120 amp circuit breaker to prevent motor damage in case of extended motor stall. Each subsystem also comes equipped with its own linear & low-dropout voltage regulator to regulate the 12 volts supplied by the hub down to 5 volts for the RP2040 microcontrollers and other 5-volt components.



Motor Control

The motor manager subsystem on the Danger Zone controls the motors of the robot using SPARK MAX motor controllers. The motor manager communicates with the motor controllers using the PWM protocol. The motor manager uses a PID controller to adjust the PWM signal provided to the motor controllers. The feedback input into the PID controller comes from encoders placed on the output shafts of each motor. The tuning coefficients of the PID controller can be adjusted in real-time over CAN bus from the main computer. The motor manager operates in terms of linear and angular velocity to provide an easier control interface for the main computer. The motor manager will also report odometry, PID, and motor statistics back to the main computer.

Safety Lights

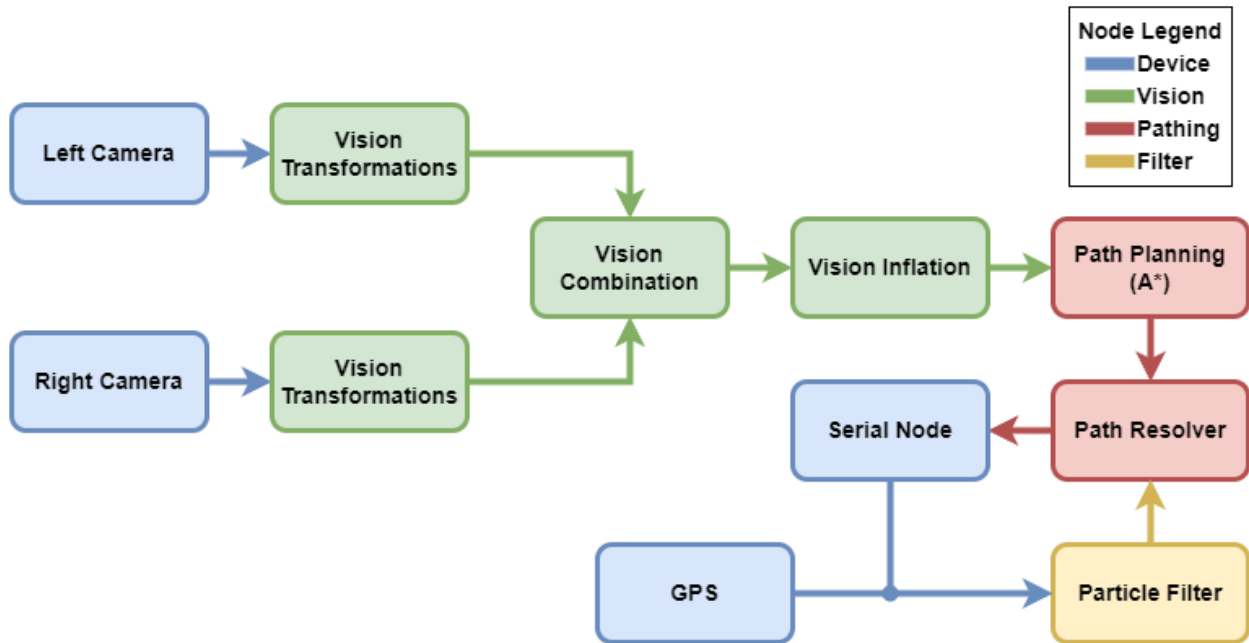
The Safety Lights subsystem on the Danger Zone controls white LED strips and an RGB LED strip. The white LED strips exist to satisfy the competition rules. The white LED strips are flashing in autonomous mode and solid in manual mode. The white LED strips are located on the front and back of the cross beam that connects the two camera poles which makes the safety light visible from all directions. The primary use of the RGB LED strip is to provide color codes to help provide information for debugging. They can also be repurposed to replace the white LED strips in case of a failure with the white lights. The white LED strips are controlled using a MOSFET switch controlled by the microcontroller. The RGB LED strip is controlled using the WS2815 pixel protocol and is given high-level commands from the main computer.

E-Stop and Safety

The E-Stop receiver & relay subsystem on the Danger Zone satisfies the emergency stop requirements. The E-Stop subsystem operates as a relay and activates E-Stop mode by pulling the E-Stop line signal low which will stop the robot. The two ways to initiate this E-Stop sequence are with the wireless E-Stop remote or the emergency stop button. The wireless E-Stop remote uses the LoRa protocol to communicate with the E-Stop subsystem. LoRa provides high reliability over long distances so the remote can stop the robot from any distance within the LoRa max communication range. The emergency stop button is an SPST switch located on the back of the robot and is within the height range specified in the competition rules. The E-Stop line goes to every subsystem on the robot, but is protected from being overwritten by other subsystems using a buffer on the Molex hub. The E-Stop line stops the robot by opening a solid-state relay that is in series with the motor controllers. This solid-state relay implementation allows the motors to be instantly disconnected from power whenever the E-Stop mode is activated. The E-Stop line provided to the solid-state relay is level-shifted to 12 volts on the motor manager subsystem to be able to close the relay and complete the connection for the motor controllers when the E-Stop line is high.

Description of Software Systems

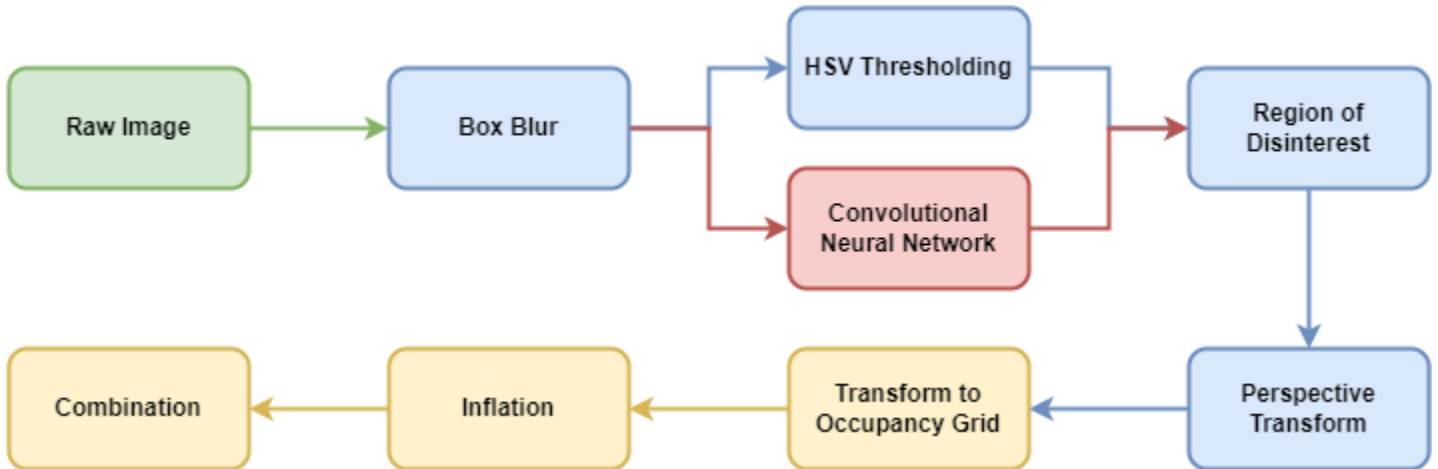
Overview



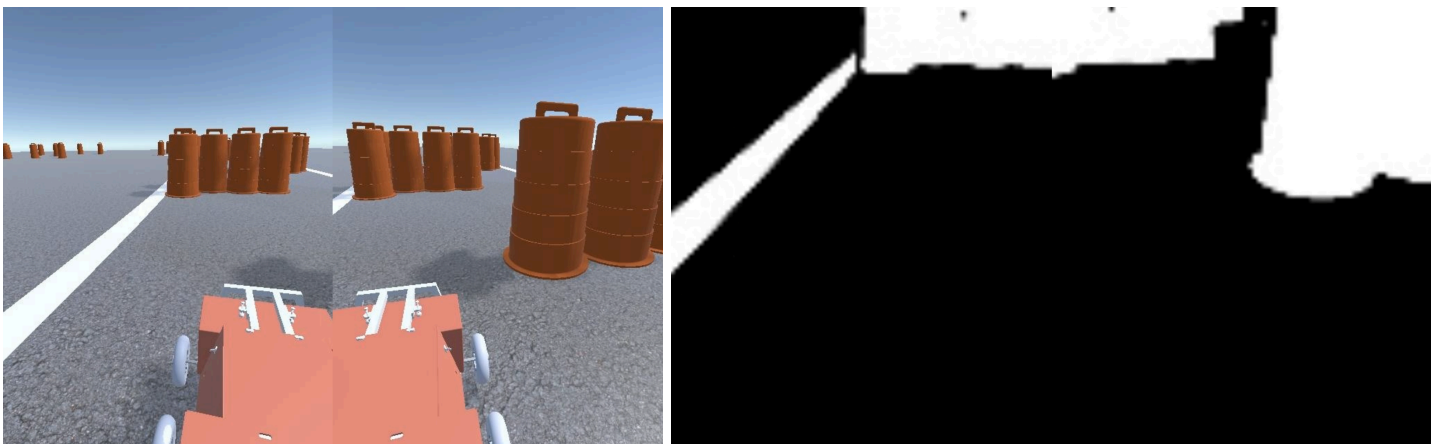
The Danger Zone’s software stack has been designed to be modular. Using the Robot Operating System (ROS), an industry standard tool for complex software systems, the software executables are divided into a set of nodes which each function independently of each other. This independent architecture enables the Danger Zone to be resistant to total system crashes as a result of individual node failure and also allows the team to work on individual nodes in parallel.

Image Transformation

Two Logitech C920x webcams capturing at 8 frames per second are placed at the top of the Danger Zone offering a wide field of view of any obstacles in front of the robot. The cameras are adjustable so that they can be orientated based on performance. Raw images from the camera captured at a 480 x 640 resolution are transformed following a number of steps: a box blur is applied to reduce noise, the CNN or HSV thresholding algorithm is applied to detect obstacles, and a region of disinterest is applied to remove a parallelogram of the image where the robot is so that it is not detected as an obstacle. This produces an image that is equivalent to the black and white figure on the next page. The next step the image takes is to get transformed into an 80 x 80 occupancy grid that is “inflated” up to the size of the robot by taking every pixel and inflating them into circles that overlap each other. Finally, when both cameras produce the final occupancy grid they are combined into a 80 x 80 grid by taking every 2nd column in both grids and inserting them into the new occupancy grid on the left or right respectively. The path a single image takes independent of its location can be seen in the following figure.

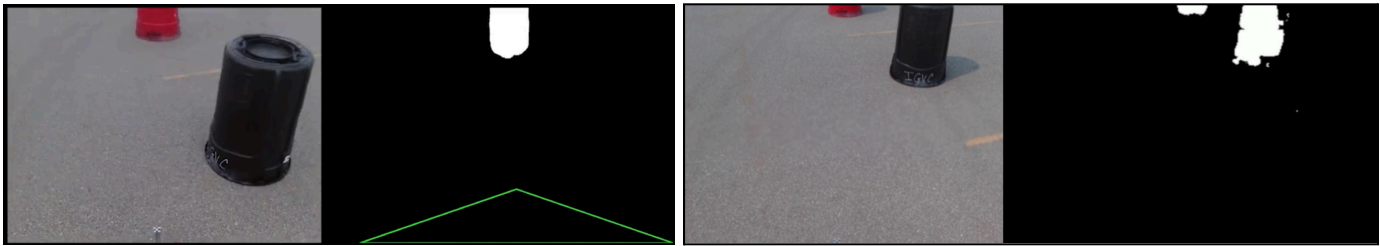


The following two images demonstrate the first five steps of the process from the raw image to the perspective transform. The goal selection and path planning section will demonstrate what the occupancy grid looks like at the combination step.

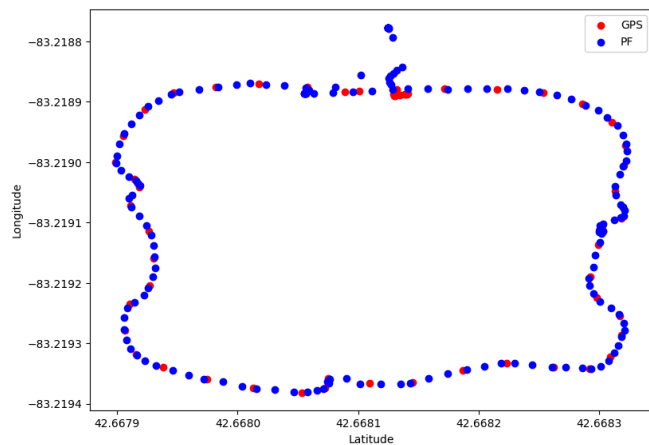


U-Net for Computer Vision

Accurate lane and obstacle detection is critical to the robot’s performance. While HSV thresholding works well in many cases, it can often fail to capture grayscale objects. The white, gray, and black barrels on the AutoNav course therefore present a difficult obstacle. To solve this, a machine learning model was created to identify lanes and obstacles. The model chosen is a U-Net model, a specific convolutional neural network (CNN) architecture that excels at the task of identifying classes of objects. The model was trained on previous competitions as well using images from the simulator. The result is a robust and effective way to identify obstacles that the Danger Zone should path around. In the figure below, the left image shows the HSV method failing to identify the black colored barrel while the right image shows the CNN successfully identifying the barrel.



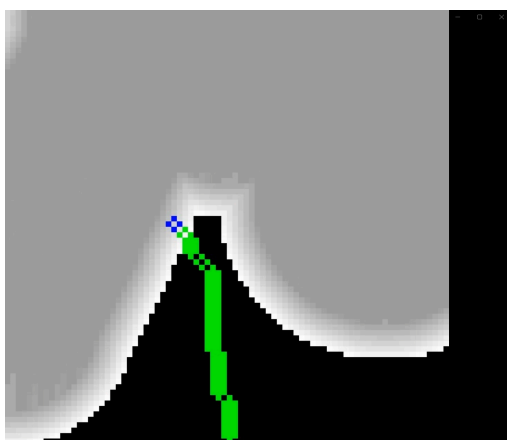
Particle Filter Localization



The Danger Zone localizes itself with a Particle Filter (PF) that continuously estimates the robots position. The position includes a global location, local location, and heading. The PF takes sensor data from the motor encoders and GPS and produces a filtered estimate of the robot's state. Using the GPS data, the robot can compare it against the encoders combined with some randomness to reduce the drift that is seen by a traditional dead reckoning system. In testing, the PF is capable of getting a precise lock on the Danger Zone's location and heading within a few seconds. This allows the robot to maintain a precise location that is resilient to

encoder drift, while also providing an estimate for the heading. The image shown is an example of the particle filter running, with red dots indicating the robot's GPS position and the blue dots indicating the particle filter's estimated position.

Goal Selection and Path Planning



The goal selection algorithm is a combination of multiple functions that produces an optimal position for the robot to head towards. This position is based on a number of factors, but most importantly; the next gps waypoint and the cost map. In order to keep the Danger Zone moving forward and making positive progress on the course, the goal selection algorithm ensures that the next goal is headed in the direction of the next waypoint and is weighted based on the distance from the waypoint and the angle between the waypoint and robot. After selecting a goal, the A* path planning algorithm is used to find an optimal path to the goal. The figure on the left shows the Danger Zone planning through the obstacles shown in the image transformation section.

Motion Generation and Monitoring

As mentioned in the previous section, the path planning algorithm generates a path that the robot will follow. Using pure pursuit, a path tracking algorithm, a lookahead point is generated by continuously expanding the lookahead radius up to a max radius. If no lookahead point is found, the robot enters a backup routine where it will back up until a lookahead point is found or the routine times out after a configurable amount of time. If a lookahead point is found, an angle is calculated between that point and the robots heading which results in an angular error term. The error is then multiplied by a configurable constant that defines the robot's turning speed, allowing operators to adjust how fast or slow the robot turns based on the distance to the lookahead point. Similarly, the forward speed of the robot is determined by that same error term which causes the robot to slow down as it's turning and speed up on straight aways. This process runs approximately 20 times a second which allows for quick response times if obstacles are suddenly detected.

Configuration and GUI

To enable rapid prototyping and debugging, the Danger Zone comes with a custom configuration system and user interface. The configuration system allows the team to configure both system level and firmware level parameters. On the firmware level this enables configuration of the motor manager and safety lights. On the system level, this lets the operators tune vision, the particle filter, camera frame rate, etc. With this system, regardless of environment conditions as they evolve throughout the day, the team is able to adjust parameters to be running the robot at all times. The vehicle's custom user interface enables the team to see exactly what the robot is thinking and use the configuration system. For example, being able to view what the Danger Zone believes to be obstacles proves vital in tuning the vision parameters when the environment changes due to factors such as the time of day. Shown below is the dashboard of the User Interface, which shows information like GPS and IMU data, path planning information, and the latest motor commands and feedback.

The screenshot shows a dark-themed dashboard for the 'Danger Zone' robot. At the top, there is a navigation bar with the following items: Dashboard (selected), Vision, Logging, Configuration, Conbus, Preferences, and Debug. The main content area is divided into two columns. The left column contains several sections: 'General' with State: Autonomous, Mode: Simulation, and Mobility: Enabled; 'Motors' with Forward/Angular Velocity: (0.097, -0.759) and Delta X/Y/Theta: (0.0000, 0.0000, 0.0000°); 'Position' with Local: (6.0678, -0.5530, 35.538°) and Global: (35.19493480, -97.43861390); and 'GPS' with Latitude/Longitude: (35.1949348, -97.4386139), Current Fix: 3, Is Fixed: Locked, and Satellites: 7. The right column is titled 'Device States' and lists several components, all of which are in an 'Operating' state: autonav_display_broadcast, autonav_playback, scr_logging, autonav_filters, autonav_vision_transformer, autonav_vision_combiner, autonav_vision_expandifier, autonav_nav_resolver, and autonav_nav_astar.

Logging and State Machine

Real time data visualization is invaluable during testing and at competition. However, some of the most important data that is used is what is generated whilst the robot is running behind the scenes. After every run, a zip is created that contains logs from many subsystems and includes data such as GPS, estimated position, the camera feed at all stages of the transformation process, etc. Using this data, it is possible to visualize the position of the robot to ensure it is correctly following waypoints, view the course as seen through its transformed vision to adjust any parameters, etc. This has continued to prove absolutely necessary as micro adjustments are able to be made to the Danger Zone pushing it beyond the original expectations.

The state machine allows the robot to switch between four states: autonomous, manual, disabled, and shutdown. It also allows the robot to switch between three modes; competition, practice, and simulation. Each of these states and modes in combination with each other tune different parameters for different environments. For example, when the Danger Zone is doing a run of the simulation it is set to autonomous and simulation respectively, which tunes the parameters to use the simulators GPS system and tuned vision. For ease of use, the robot comes with a manual mode where it is able to be driven via steam controller.

Additionally, the state machine also provides a way for individual nodes to control their own states and view the states of others. This is critical in cases where a node is dependent on another. For example, the vision transformation node is dependent on the camera input node, so if neither of the two cameras are functioning then the vision node will be disabled. All of this information regarding individual device states is visible in the UI as seen in the previous section which lets operators quickly debug issues rather than digging through the logs.

Cyber Security Analysis

NIST RMF Process

The NIST Risk Management Framework (RMF) is a set of guidelines designed to help organizations manage and mitigate risks to their information systems and data. There are seven steps that should be followed when following through the NIST RMF process: prepare, categorize, select, implement, assess, authorize, and monitor.

Threats

Threat Description	Confidentiality	Integrity	Availability	Control Code(s)
--------------------	-----------------	-----------	--------------	-----------------

Physical damage or modification	Low	Medium	Medium	PE-18
Access to runtime software	Medium	High	Medium	AC-3
Communication interruption	Medium	Medium	Medium	AC-4
Unauthorized access	Medium	High	Medium	AC-17
Misuse of Privileges	High	High	Medium	AU-2

Cyber Controls and Implementation

Control	Implementation
PE-18: Location Of Information System Components	Vehicle code and configuration is stored securely and remotely on Github. Any damage to the robot's computer systems could be physically replaced and perform identically.
AC-3: Access Enforcement	Software is uploaded to the robot via an encrypted SSH connection over a secure, physical ethernet connection to the NUC. The WiFi and Bluetooth networks of the NUC are disabled on startup.
AC-4: Information Flow Enforcement	All data is stored locally on the vehicle. The external HMI must authenticate with the vehicle before making data requests.
AC-17 Remote Access	As explained in cyber control AC-3, remote access to the NUC unless connected physically is disabled.
AU-2 Audit Events	All events on the NUC are monitored and placed into a log file for analysis, any critical events like failed SSH attempts are priority notifications and admins will see them upon login.

Vehicle Analysis

Lessons Learned

Throughout the development and design process of the Danger Zone, every member of the team learned many valuable lessons that will help advance their knowledge in Sooner Competitive Robotics and their careers. The first lesson almost every member learned this year was that communication is absolutely critical to building a successful robot. Communication issues were common between subteams and would often lead to issues in the designs for fitting components.

Failures Points

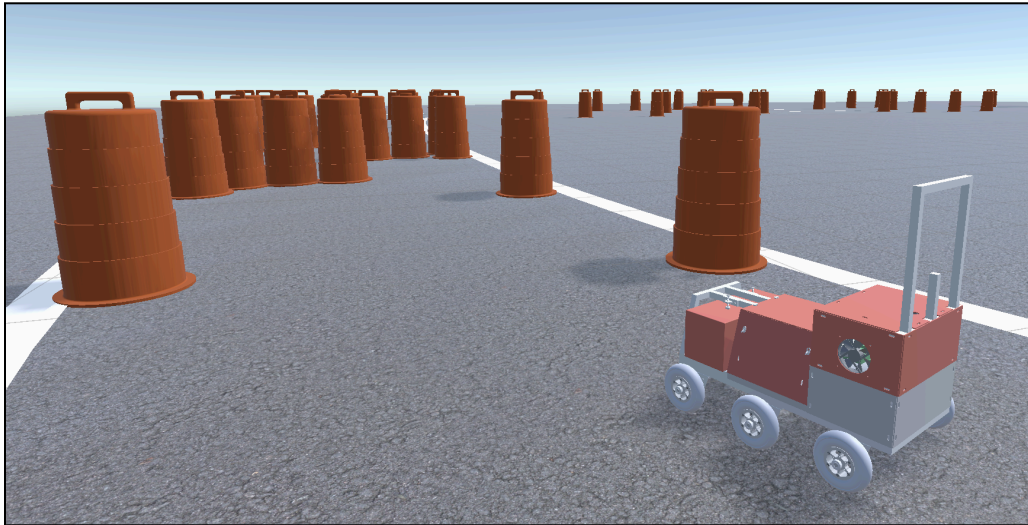
Failure Point	Resolution
Electrical Subsystem Failure	Robot would see the subsystem drop and fail gracefully while other subsystems remain at their original state.
Critical Software Failure	Robot is immediately stopped due to timeout.
Mechanical Chassis Damage	Chassis is modular and all parts are easily replaceable.
Software Parameter Failure	All parameters for the software system, including firmware, are all customizable via an easy to use interface.
Non-critical Software Failure	The node that failed is dropped from the ROS network and restarted if possible.

Bug Tracking and Version Control

The bug tracking and version control is handled through Github, a developer platform that uses the Git version control software and provides an easy to use UI. This enables the software team to create issues, pull requests, and branches that allow for concurrent development among multiple developers. Additionally, Github is used as a deployment platform which will automatically automate and deploy code as needed onto the Danger Zone when it is turned on. The main branch, where the robot pulls from, is protected and can only be pushed to via a pull request that is reviewed and approved by at least two other software members including the software lead and team captain which ensures all code is safe and regression-free.

Simulation

Due to the nature of concurrent design, it is impossible to have the robot operationally available for software testing at all times. For this reason, an in-house simulator was created, Scrabby, which can produce realistic conditions for the robots to virtually test on. The Danger Zone has been tested on this simulator extensively and has consistently been able to complete the course in 3 minutes at 1.5 meters per second. Additionally, the simulator can be used to constantly tune and adjust many conditions the Danger Zone will have to face at competition such as sensor noise, steepness of the ramp, barrel color, lane width, and more. The team plans to further refine the simulator by adding additional vehicles, sensors, maps, and conditions such as rain or clouds.



Performance Assessment

As of submitting this report, the Danger Zone is performing as expected on all fronts. Through all of the extensive testing such as the simulator, physical ramp testing, and more, the vehicle proves to be resilient to environmental change, lighting changes, and other conditions which may be present at competition.

Parameter	Required Value	Measured Value
Max Speed	5.0 MPH	4.9 MPH
Average Speed	1.0 MPH	4.1 MPH
E-stop Range	100ft	200ft
Ramp Ascension	15% incline	15% incline
Terrain Navigation	Concrete/Asphalt	Grass, Concrete, and Asphalt
Estimated Run Time	N/A	120 minutes per 12V battery
Simulated Completion Time	N/A	Average completion in 3 minutes