Sooner Competitive Robotics

The University of Oklahoma
"Weeb Wagon"

Submitted May 15th, 2023

Braden White	white@ou.edu	Team Captain
Noah Zemlin	noah.zemlin@ou.edu	Advisor
Tyler Julian	tyler.james.julian@ou.edu	Mentor

I certify that the design, development, and work put towards this project by the Sooner Competitive Robotics students is significant and equivalent to a senior design course.

X N Zemlin

1 Introduction

1.1 Overview

In anticipation of the 30th Annual Intelligent Ground Vehicle Competition, the Auto-Nav team from the University of Oklahoma set out to improve upon the design from last year's competition. For 2023's Weeb Wagon, we created a chassis with a durable design and larger wheels for better mobility around the course. The team prioritized creating a more seamless and well integrated software stack that allowed for easy debugging, configuration, performance monitoring and logging.

1.2 Team Organization

The AutoNav team is a team under Sooner Competitive Robotics at the University of Oklahoma. It is comprised of eleven undergraduate students with no returning members from last year's team. A large portion of the team is composed of first and second year students without prior experience in robotics.

The team is organized into electrical, software, and mechanical sub-teams. Each sub-team was led by an experienced member or upperclassman that guided members toward completion of the required tasks. Sub-team leads periodically checked in with their members to ensure that the tasks were being completed sufficiently. If additional help was needed, it was the sub-team leads' job to provide assistance. Also, sub-team leads report to the team captain to ensure that each member is on the same page regarding the design and progress of the robot.

Name	Grad Year	Major	Role	Hours
Braden White	2024	Computer Engineering	Team Captain/Electrical Lead	350 hrs
Jorge Exinia	2023	Computer Engineering	Electrical Member	150 hrs
Brandon Kahn	2024	Computer Engineering	Electrical Member	250 hrs
Min Kang	2026	Electrical Engineering	Electrical Member	150 hrs
Michael Falcone	2026	Electrical Engineering	Electrical Member	200 hrs
Matthew Gonzales	2026	Electrical Engineering	Electrical Member	150 hrs
Dylan Zemlin	2025	Computer Science	Software Lead	400 hrs
Antonio Chappell	2025	Mechanical Engineering	Software Member	250 hrs
Vivian Easley	2024	Mechanical Engineering	Mechanical Lead	150 hrs
Logan Williams	2026	Mechanical Engineering	Mechanical Member	100 hrs
Yuhao Zhang	2024	Mechanical Engineering	Mechanical Member	200 hrs

2 Design and Strategy Overview

2.1 Assumptions and Priorities

The overarching goal in the creation of the Weeb Wagon was to create a robot that improved upon our past iteration by integrating electrical and software more efficiently to avoid obstacles. Also, to utilize our systems in conjunction to complete the course in a quicker time than its predecessor.

2.2 Cost

For the Weeb Wagon, it was a point of emphasis to maximize our resources and keep our cost down. In total, the overall cost to build the Weeb Wagon was around 2,550 dollars, including our purchase of the Intel NUC.

We used previously purchased electrical equipment, such as our camera, and low-cost extruded aluminum to build the chassis. We also created our own printed circuit boards (PCBs), rather than purchasing manufactured electrical products. This was done to keep costs down and to provide valuable experience in the design and fabrication of electrical components. The team is confident that despite its low cost, the product will not be hindered in its performance.

2.3 Safety

While designing the Weeb Wagon, safety was an essential element of the design process. Most notably, this is evident in the care we took in creating our E-STOP electrical subsystem and the additional safeguards put into place regarding software.

Multiple different systems can disable the motors when the emergency stop state is declared. In order to leave that emergency-stop state, the E-STOP requires physical contact with the robot to disengage.

The E-STOP receiver (see section 5.4.1) has become the central hub of the E-STOP line. It and only it can control the line, which prevents instances of the line being driven differently by two different electrical systems. In addition to this, a timer is constantly checked between the E-STOP remote and its receiver to make sure that the robot is only able to move when the remote is turned on and nearby. If the receiver is unable to connect to the remote for a specified amount of time, connection is considered lost and the robot E-STOPs.

A new electrical subsystem, object detection, has been put into place this year. The ultrasonic sensors attached to the robot send distance data directly to our motor controller. If the robot crosses a certain threshold distance away from an object, it sends a message to the motor control subsystem to prohibit movement in the forward direction.

2.4 Reliability

Each of our subsystems were tested thoroughly throughout the course of the design process to make sure that everything functions properly, regardless of the circumstances. In addition to this, modularity has been key to our design as well. Each electrical and mechanical part of our system is easily replaceable in the event that it fails. Our robot is not reliant on just the main computer or a single system either. Each defined subsystem contributes equally. Because of this, parts can be easily replaced with minimal difficulty.

2.5 Durability

The Weeb Wagon uses extruded aluminum that is bolted together and supported extensively, leading to an extremely durable chassis. Its octagonal shape functions as a brace for the main portion of the chassis containing the electronics and motors. Important electronics pertaining to motor control are secured in between each wheel of the frame. Polycarbonate sheets are placed in between the wheels to protect those important devices. Furthermore, the electronics on top of the platform are inside of an/a enclosure, protecting the components from the elements that it may interact with while testing or navigating the course.

3 Innovations

3.1 Tangent Based Path Planning

One of the largest limiting factors of last year's design was the computation time of its path planner. This year, the team developed and implemented a novel path planning approach called tangent based path planning. Unlike other path planning algorithms, tangent based path planning avoids the long computation time associated with searching a grid. Minimizing computation time allows the Weeb Wagon to complete the course more quickly, as objects that are detected can still be avoided at high speeds.

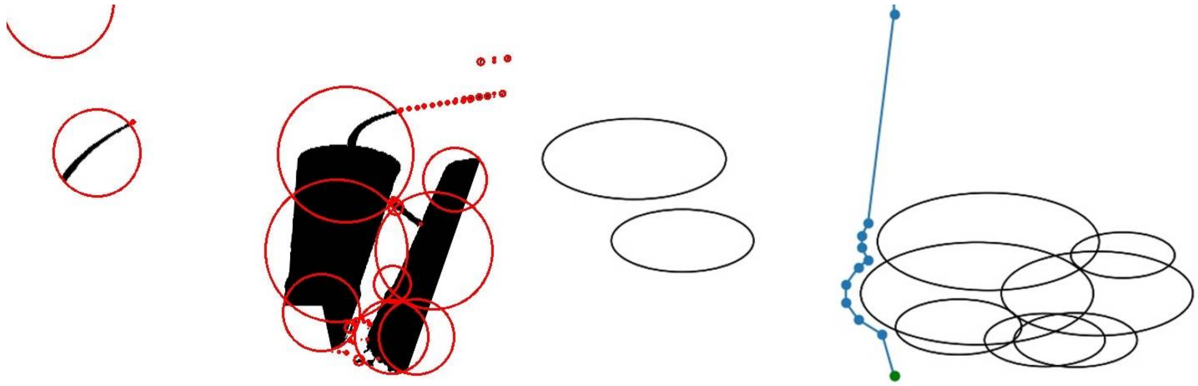


FIG. 1: Tangent Based Path Planning

3.2 Improved E-STOP System

While last year's E-STOP system was sufficient, we decided to further safeguard its important use case on the Weeb Wagon. The E-STOP receiver that receives messages from the E-STOP remote now acts as a hub for the devices that interact with the E-STOP line. This leads to a decrease in probability of the line being unintentionally driven high or low.

3.3 Object Detection on the Firmware Level

The computer vision software is still the main driver of detecting and avoiding obstacles. But, in favor of redundancy, the team added an object detection system on the firmware level. The object detection hardware sends the distance and location of the nearest object over CAN-bus to the system. In the event an object is less than a certain threshold distance, the system prohibits movement in the forward direction, only allowing the main computer to back up.

3.4 Mechanical Design

The mechanical design for the Weeb Wagon is an improvement over last year's iteration. The casters are more robust than last year's and the front caster has suspension to allow the robot to go over the ramp smoothly. The Weeb Wagon features wider wheels and an even more durable design that improves navigation. An improved design for the sensor pole has improved the stability of the camera as well.

4 Mechanical Design

4.1 Overview

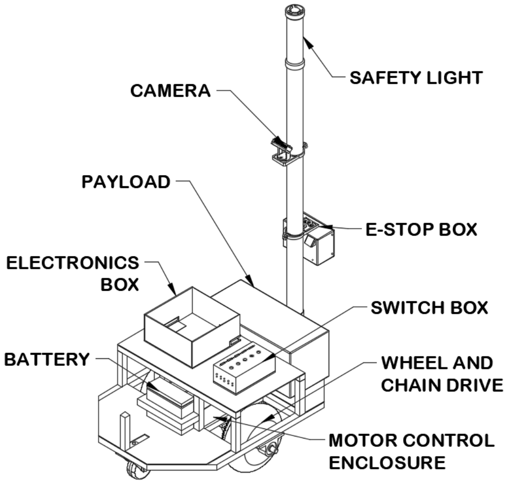


FIG. 2: Render of the Weeb Wagon

4.1.1 Chassis

The Weeb Wagon's design was built from the ground up to create an overall improvement in the maneuverability of the course. The chassis was made with extruded t-slot aluminum due to its cost, weight, and modularity. An octagonal shape improved stability compared to a rectangular design. Furthermore, the nature of the new design increases durability due to the main supports having more points of contact. The addition of wider wheels allows for better traction, resulting in improved maneuverability on obstacles such as the ramp. The chassis sits lower to the ground, which enables better maneuverability by reducing the impact of abrupt movements. Polycarbonate is used as the surface area on top of the robot and in-between the motors, protecting its internals from weather and damage.

4.1.2 Suspension

The inclusion of two separate casters on opposite ends of the robot enabled us to create a suspension system. The rear caster, the more robust of the two, is the primary load-bearing caster. It stabilizes the forward motion from the acceleration generated by the motors. The front caster was decided to include a spring to improve stabilization when descending the ramp.

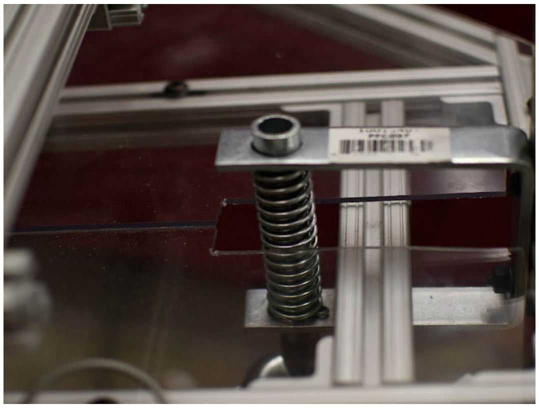


FIG. 3: Suspension of the Weeb Wagon

4.2 Sensor Pole

The usage of a sensor pole allows our camera and safety lights to be as high as possible, our sensors to be easily accessible, and as a mounting location for the E-STOP button.

FIG. 4: Sensor Pole of the Weeb Wagon

4.2.1 Camera and GPS

The camera is mounted on the PVC pipe with a 3D-printed mount that allows the camera to be moved to an optimal height and angle. While the GPS is mounted on the sensor pole just below the camera, its antenna is attached to the top of the pole. This placement decreases the probability of electromagnetic interference.

4.2.2 Safety Lights

The safety lights are mounted on top of the pole for maximum visibility. They are bright strips of LEDs on a contrasting background making them apparent in direct sunlight. They are wrapped tightly onto a separate PVC piece that is then placed on top of the main pole for easy removal and replacement. This piece also has a protective layer around the lights, ensuring that the lights will not get damaged by outside elements.

4.2.3 E-STOP Box

The E-STOP box houses the mechanical E-STOP button, E-STOP receiver PCB, and safety lights PCB. Each PCB is mounted on standoffs vertically in the box. Each is visible due to the use of clear polycarbonate, making it easy to see the LED indicator lights on each. The button is directly wired into the E-STOP receiver PCB and the safety light's wires are routed through the back and into the PVC pipe.

4.3 Weatherproofing

Every item that is susceptible to damage from outside elements is in an enclosure of its own. This includes our power delivery system, our motor controllers, safety system, and the main computer. Canopies or "vent hoods" are utilized on openings of each box or enclosure to prevent the electronics inside from getting damaged. In addition, wires are routed in enclosed spaces to shield them from possible damage.

4.3.1 Electronics Enclosure

Our electronics enclosure is divided into two boxes: the power delivery box and the electronics box. The power delivery box has easy-to-use switches on top of it to power the different subsystems of the robot. The electronics box includes the DB9 hub, object detection hardware, NUC computer, and the associated cables with each device. The power delivery box is a custom 3D-printed enclosure, while the electronics enclosure is a modified, weatherproof junction box.

5 Electronic and Power Design

5.1 Overview

For the electrical and power design, the team focused on improving systems for increased redundancy regarding safety and usability. The goal was to create PCBs and devices that were comparable to what you would find in a consumer product. As a result of this mindset, the team has created a robust system that functions as needed and is usable to those that didn't have a hand in making it.

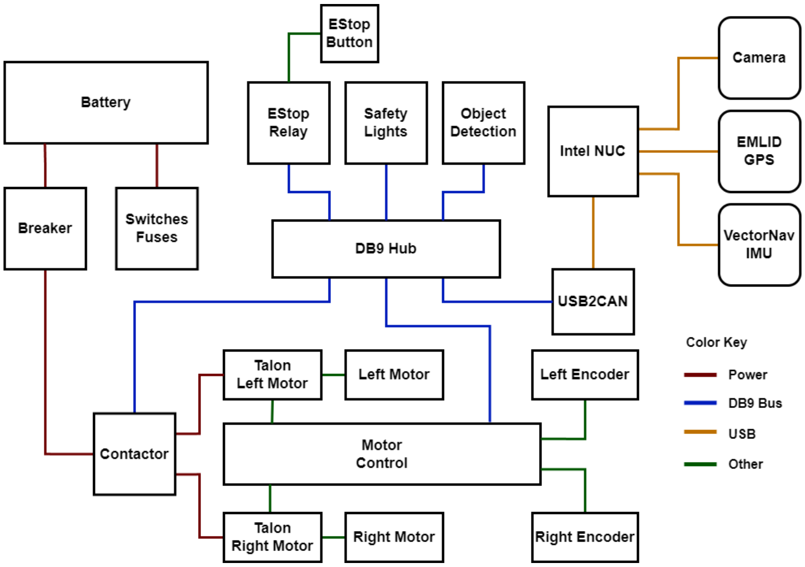


FIG. 5: Electrical Block Diagram

5.2 Power Distribution

For the Weeb Wagon, we went with a 12V battery to power the robot. Power is delivered to each subsystem by the switch box, which has multiple switches paired with fuses to ensure safe power delivery. By having individual switches for each subsystem, we can conserve power while testing different parts of the system.

5.3 Electronics Suite

5.3.1 DB9 Bus

The Weeb Wagon uses the DB9 bus to distribute both power and data using the CAN-bus protocol throughout the robot. CAN-bus is an industry standard and is incredibly robust. This protocol is used to send messages to and from the NUC computer and the various microcontrollers.

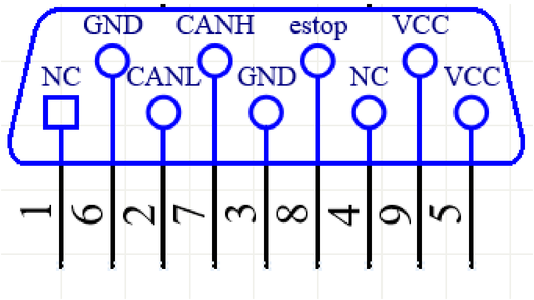


FIG. 6: DB9 Pinout

5.3.2 Object Detection

A new addition to our robot this year, the object detection PCB accomplishes object avoidance on the firmware level. While computer vision is still the main driver of our autonomous system, this PCB is an extra layer of redundancy. The Weeb Wagon utilizes three ultrasonic sensors positioned on the front of the robot: one on the left, right, and center. The microcontroller driving the PCB sends data over CAN-bus continuously to notify the system how close the Weeb Wagon is to an object. In the event one of the ultrasonic sensors sends a message that is under a certain threshold distance, the motor controller makes it so that the Weeb Wagon is unable to move in that direction. This allows the software to make the decision to back up and try again. With this implemented, the Weeb Wagon should never come in contact with an obstacle.

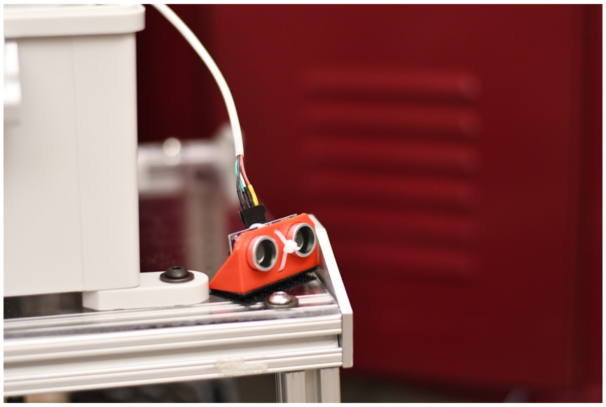


FIG. 7: Object Detection of Weeb Wagon

5.3.3 Motor Control

The motor control subsystem provides the main computer feedback about velocity, position, and orientation of the Weeb Wagon. The motors are driven using a microcontroller sending PWM signals. Quadrature encoders are attached to each motor to monitor velocity and position. A PID controller is used to maintain a constant set velocity received from commands over CAN-bus. The software can precisely control the movement of the Wagon without any hiccups.

5.4 Safety Devices

5.4.1 Remote E-STOP System

The remote and receiver utilize a set of LoRa radio modules to continuously communicate the state of the robot. The E-STOP remote sends state-requests for three different states: mobility-stop, mobility-start, and E-STOP. The mobility-stop state simply commands the robot to stop moving, mobility-start allows the robot to move, and an E-STOP signal completely locks the robot from moving until it is power cycled. When this occurs, the E-STOP line located within the DB9 bus gets driven low and the contactor cuts power to the motors.

The receiver is the only device that can drive the E-STOP line directly; it acts as a hub for the E-STOP. This makes it so that the E-STOP line can never be unintentionally driven high or low across the DB9 bus. The mechanical E-STOP button is plugged into the receiver via screw terminals, physically acting as an override on the E-STOP line when pressed.

A watchdog timer on the receiver will assume the remote has disconnected and initiate an emergency-stop state if it does not receive the current status from the remote for 2 seconds.

5.4.2 Safety Lights

The safety lights consist of a white LED strip and an RGB LED strip for more specific indication about the state of the robot. Whenever the robot is in a powered-on state or in manual mode, the white LEDs are shown to be solid. If the robot is in autonomous mode, the white LEDs will pulse at a frequency of 1 Hz. The RGB LED strip can also indicate different states, such as mobility-start, mobility-stop, and E-STOP. With the addition of the RGB lights, it makes it much more apparent what state the robot is in while operating and can help diagnose problems.

6 Software Design

6.1 Overview

The software is designed around being modular and resistant to failure. By using ROS (Robot Operating System), we are able to successfully run the robot even if parts of the system fail. This is because the entire system runs on a set of nodes rather than one process.

The Weeb Wagon codebase is open-source and is available at https://github.com/SoonerRobotics/autonav_software_2023.

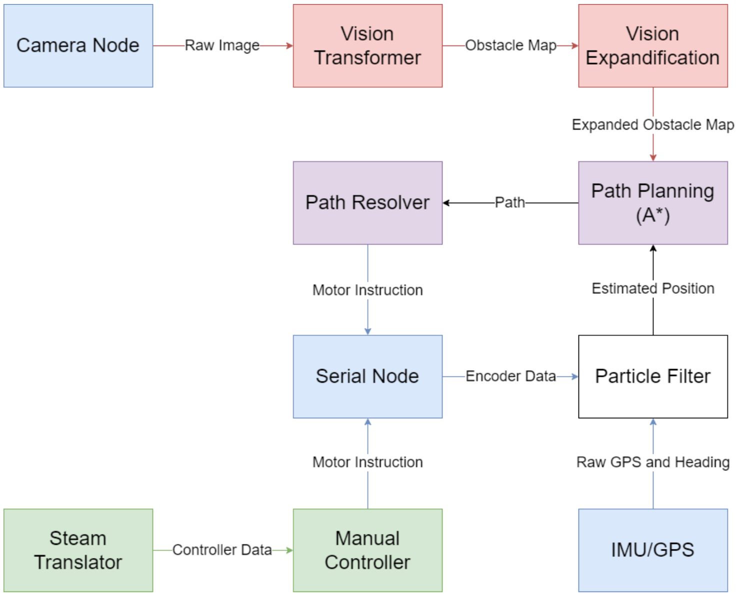


FIG. 8: Our ROS Network Diagram

6.2 Vision-based Obstacle Detection and Mapping

The robots primary source of obstacle detection and mapping is vision. A webcam is mounted to the top of the sensor pole allowing a wide field of view. For detecting obstacles, an image is taken from the robot's camera and fed through several computer vision techniques which transform the raw image into an obstacle map. We blur the image to reduce noise, filter using an HSV range similar to the road, and invert the image to properly treat the obstacles as obstacles. This allows us to detect all obstacles that are not similarly colored to the road.

After an obstacle map is produced, we perform an ”expandification” process which allows us to treat obstacles as direct units of measurements rather than pixels. Once this has been performed, we can use traditional path finding algorithms such as A^* to path plan onto a goal point which we find using in house algorithms described in the next section. Finally, we use pure pursuit which generates ”lookahead” points that the robot tracks onto by adjusting its angular velocity, and relating its forward velocity to the error between the robot's heading and the current lookahead point. If no lookahead point can be generated, the robot drives in reverse until it finds a valid point.

6.3 Goal Selection

We have been optimizing our goal selection algorithm labeled the ”Smelly Algorithm.” Its purpose is to take the map generated by our vision and mapping algorithms, and determine a goal point that the robot will go to. Our algorithm uses A^* that uses a variety of biases, namely: bias towards the center of the map, bias towards the next available waypoint, bias against obstacles, and bias towards lanes.

6.4 Speed, Ramps, and Accuracy

Our optimized vision algorithms and upgraded computer allow for an increased max speed of the robot. This year, we plan to reach a speed of 2 to 3 miles per hour, as a direct result of that increased performance. Furthermore, as a result of our vision algorithm and the methods we use to detect obstacles, the ramp presents no additional issues.

6.5 Computer Vision Techniques

The team created an algorithm using OpenCV and standard computer vision techniques that allow for lane and obstacle detection. This method takes the webcam images and feeds them through several steps to produce a final binary classified image that describes occupancy of lane or obstacle. The technique uses blurring, HSV thresholding, dilation, and camera perspective correction. In Figure 7, the top and bottom images represent the before and after of our computer vision techniques. The green triangle on the bottom image is used to cut out the robot, allowing the HSV thresholding to be exclusively for object detection.

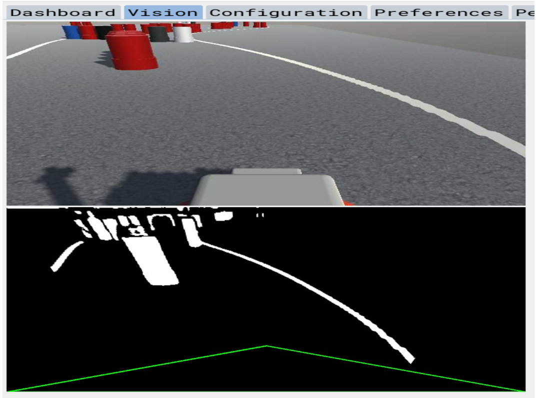
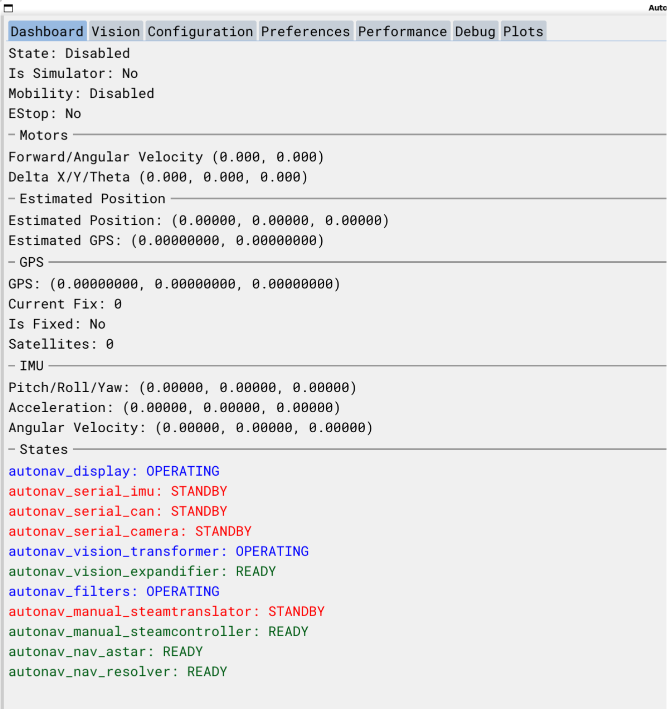


FIG. 9: The vision nodes view of the simulated course

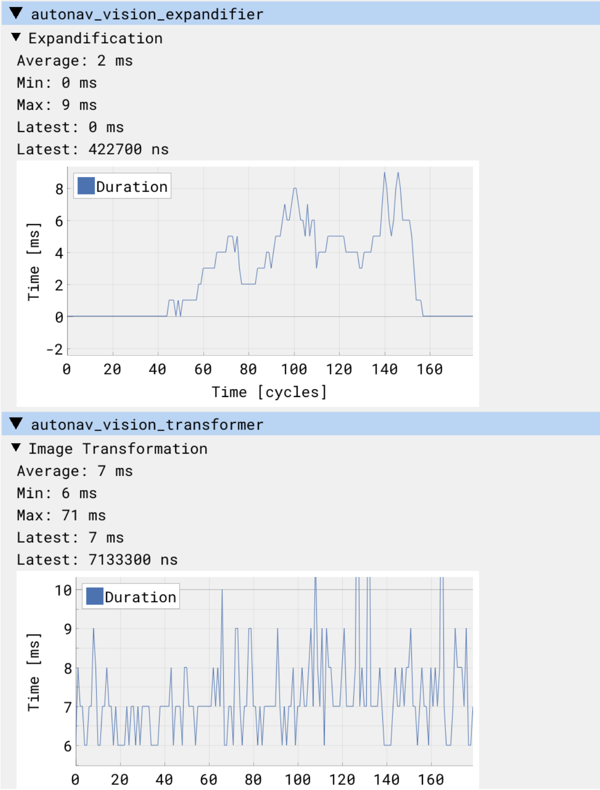
6.6 State Estimation

The autonomous system uses a Particle Filter (PF) to estimate the robot's pose and perform navigation. The pose is composed of global location and heading. The PF takes sensor data along with a kinematic model and produces a filtered estimate of the robot's state. By continuously taking GPS data and comparing it against the expected movement of the robot's encoders, applying a bit of randomness, we can reduce the drift that is produced by a pure dead reckoning system. Typically, the particle filter is capable of getting a precise location and heading of the robot within the first few seconds. This again allows us not only to keep a precise location of the robot that is mostly resilient to encoder drift, but estimate our heading which we have no original estimate on due to the lack of a tested IMU.



```
State: Disabled
Is Simulator: No
Mobility: Disabled
EStop: No
- Motors
Forward/Angular Velocity (0.000, 0.000)
Delta X/Y/Theta (0.000, 0.000, 0.000)
- Estimated Position
Estimated Position: (0.00000, 0.00000, 0.00000)
Estimated GPS: (0.00000000, 0.00000000)
- GPS
GPS: (0.00000000, 0.00000000, 0.00000000)
Current Fix: 0
Is Fixed: No
Satellites: 0
- IMU
Pitch/Roll/Yaw: (0.00000, 0.00000, 0.00000)
Acceleration: (0.00000, 0.00000, 0.00000)
Angular Velocity: (0.00000, 0.00000, 0.00000)
- States
autonav_display: OPERATING
autonav_serial_imu: STANDBY
autonav_serial_can: STANDBY
autonav_serial_camera: STANDBY
autonav_vision_transformer: OPERATING
autonav_vision_expandifier: READY
autonav_filters: OPERATING
autonav_manual_steamtranslator: STANDBY
autonav_manual_steamcontroller: READY
autonav_nav_astar: READY
autonav_nav_resolver: READY
```

FIG. 10: The primary dashboard of the Weeb Wagon



```
▼ autonav_vision_expandifier
▼ Expandification
Average: 2 ms
Min: 0 ms
Max: 9 ms
Latest: 0 ms
Latest: 422700 ns
```

```
▼ autonav_vision_transformer
▼ Image Transformation
Average: 7 ms
Min: 6 ms
Max: 71 ms
Latest: 7 ms
Latest: 7133300 ns
```

FIG. 11: The performance monitoring dashboard during a live run

6.7 State Machine

This year, our team implemented a new state machine that allows us to monitor the individual actions and states of every node on the network. Specifically, each node in the system is allowed to be in one of four states: offline, standby, ready, and operating. The entire system also has its own states: disabled, manual, autonomous, and shutdown. The primary goal when creating this state system was to ensure that individual nodes control their own states, while the system's states are controlled by the collection of individual states. For example, the autonomous system state is only *legal* when the vision nodes, serial nodes, and path planning nodes are all in a ready or operating state. If a required node falls out of its ready or operating state, then the entire system reverts back to the disabled state. So, this means that the robot is only capable of driving either manually or autonomously when all required nodes are in ready or operating, preventing the robot from losing control. Lastly, the shutdown state is designed purely to tell every node in the network that it must shutdown, allowing the entire network to be restarted easily.

6.8 Performance Monitoring

A new performance monitoring system was implemented this year that allows us to analyze time complex operations such as object expandification and obstacle detection. With this system, we were able to drastically increase the performance of our object expandification, leading to an average 10 millisecond run time compared to last year's 150 millisecond average.

6.9 Tangent Based Path Planning

In response to performance improvements, we also set out to create and implement a new approach to path planning. Tangent based path planning is a path planning strategy that models obstacles as circles or groups of circles. A straight line path between waypoints is then intersected with these circles. If intersections are detected, the path is updated so that it lays tangent to each obstacle at the point of intersection. The time complexity of the algorithm grows in proportion to both the number of obstacles and the number of waypoints, but notably not the size of the area being searched. This translates to more rapid path planning, which means faster maximum speeds for the Weeb Wagon.

default	Preset

Path Resolver

0.300	Forward Speed
-0.500	Reverse Speed
1.200	Radius Multiplier
4.000	Radius Offset
0.700	Radius Max

Vision Transformations

0	Hue Lower
0	Saturation Lower
35	Value Lower
255	Hue Upper
200	Saturation Upper
180	Value Upper
5	Blur Amount
2	Blur Iterations
0	Trapezoid TL
0	Trapezoid TR

FIG. 12: The configuration dashboard

6.10 Configuration over CAN-bus (CON-bus)

Additionally, in response to long compilation times when needing to change configurations, the team created a new configuration system that allows us to read and write values to any part of the network using the CAN-bus protocol. Furthermore, it also allows the user to save and load presets to swap between options like vision thresholding in different weather. This configuration system runs not only on the robot's main computer, but also on each microcontroller. For example, the motor control system allows the PID controller to be tuned live, which enables the ability to save and load the optimal configuration without the need to re-flash the firmware.

7 Failure Modes and Resolutions

Failure Mode	Resolution
Software vision inconsistency	Added object detection on the firmware level for extra redundancy.
Electrical part failure	Robot would see subsystem drop from CAN-bus and fail gracefully; independent subsystems.
Non-critical software system failure	Dropped from the network to prevent interference.
Critical software system failure	Stopping robot while trying to redeploy to prevent further failures.
Mechanical chassis damage	Chassis is modular and its parts are easily replaceable
Software parameters not tuned correctly for environment	Using CON-bus, values can be modified accordingly

8 Simulation

To validate and test our software, we developed an in-house simulator that accurately simulates the 2023 Weeb Wagon and the Auto-Nav course. This allows the software team to test and design software from home or when the robot is under maintenance. The simulator allows us to adjust the noise profiles of the simulated sensors, which enables us to evaluate the performance of our particle filter under various conditions. We have found that the software's performance in the simulator is very similar to the robot's performance in real-world testing. The team plans to further improve the simulator by allowing for more vehicles, sensors, maps, and conditions so that we can use it for Self-Drive or other robotics competitions.

The simulator is open-source and the latest version can be downloaded at https://github.com/SoonerRobotics/scr_simulator.

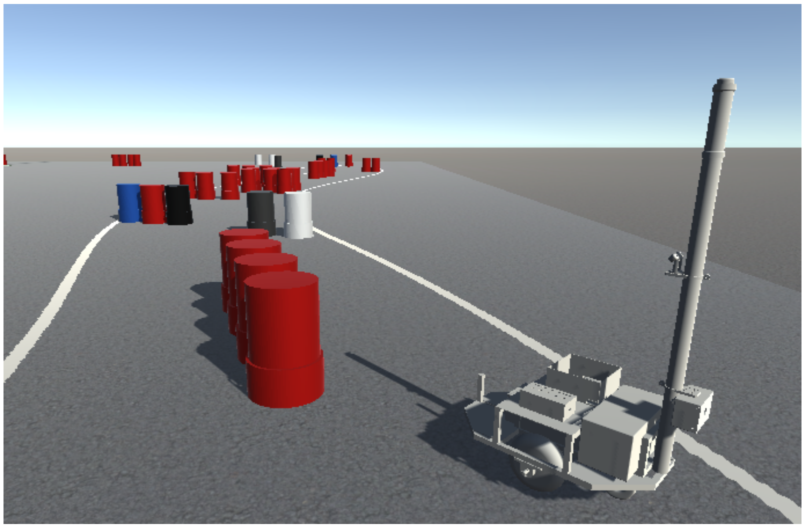


FIG. 13: Weeb Wagon in the simulator. The map emulates a high difficulty AutoNav course.

9 Performance Testing

The Weeb Wagon has been extensively tested in simulations and real-world replicas of the AutoNav course.

- The Weeb Wagon has achieved a max measured speed of 2.2 m/s or 4.9 MPH.
- The Weeb Wagon has an estimated run-time of 140 minutes with a measured run-time of 120 minutes.
- The E-STOP remote has a measured range of at least two-hundred feet in an urban setting.
- The Weeb Wagon easily ascends and descends a ramp with a 15% incline.
- The robot is able to navigate around obstacles on mock courses that were recreated on concrete and carpet.
- The Weeb Wagon consistently completes the course in simulation.

10 Assessment

The team's current assessment of the Weeb Wagon is that it is performing better at all levels than our previous year's robot, the Rat Van. Primarily, this increase in performance is seen through the use of optimized algorithms, better testing and debugging tools, a sturdier chassis design, and new components such as the Intel NUC. The mechanical design itself has seen a redesign in concerns of strength and stability, particularly targeting the design constraints and limitations due to the ramp. Furthermore, as mentioned, the electrical system has seen improvements in object detection at the firmware level and a reliable E-STOP system. Lastly, the software team made it a critical goal to rework and rewrite the entire system, enabling substantial performance gains. Our assessment concludes that our robot will perform at a higher level this year in all matters.