



Wisconsin
Robotics

Singularity

Faculty Advisor Statement

I certify that the engineering design of Singularity, the robotic vehicle described in this report, has been significant and equivalent to what might be awarded credit in a senior design course.

Professor Michael Zinn
Department of Mechanical Engineering

Table of Contents

1	Introduction	2
2	Innovations.....	2
2.1	Mechanical Innovations	2
2.2	Electrical Innovations	2
2.3	Software Innovations	3
3	Design Process	3
3.1	Team Structure	3
3.2	Project Planning	3
3.3	Development.....	4
4	Mechanical Design.....	4
4.1	Drivetrain	5
4.2	Chassis	5
4.3	Serviceability	6
4.4	Sensor Placement.....	6
5	Electronics Design	7
5.1	Sensors	7
5.2	Embedded Control System	8
5.3	Main Computer	8
5.4	Motion Control.....	8
5.5	Power System.....	9
5.6	Electrical Safety Features.....	9
5.7	User Interaction.....	9
6	Software Design.....	10
6.1	Localization	10
6.2	Feature Extraction.....	11
6.3	Mapping.....	12
6.4	Path Planning.....	12
6.5	Driving	13
6.6	JAUS.....	13
6.7	Simulation	14
7	Performance.....	14
8	Cost Summary	15
9	Conclusion.....	15

1 Introduction

Wisconsin Robotics is pleased to reintroduce Singularity to the 20th annual Intelligent Ground Vehicle Competition. After the 2011 IGVC, the team set out to make Singularity smarter, more reliable, and easier to operate, and is proud to submit a robot that is both innovative and practical. As in years past, the team sought to go beyond the challenges of the competition and design a versatile and adaptable platform that is useful for a wide range of applications.

Wisconsin Robotics is composed primarily of undergraduate students along with a few graduate students at the University of Wisconsin–Madison studying engineering and computer science. Each sub-team meets several times every week outside of class to work on their projects. All of our members are volunteers who participate without receiving compensation or course credit.

2 Innovations

Singularity has been significantly upgraded since last year, and features a more robust and resilient mechanical design, several new electronic components, and a more intelligent and reliable software platform.

2.1 Mechanical Innovations

Many changes have been made to the mechanical systems to improve the simplicity and reliability of the robot. The one-piece outer shell has been removed in favor of two component boxes and more permanent paneling, providing improved weather resistance and serviceability. A new operator panel has been added to consolidate frequently accessed controls and simplify the operation of the robot. The turn motors have been repositioned for easier access and adjustment, as well as more efficient use of space. A single bolt provides tensioning for two turn motors. A new gear train allows both the potentiometer and absolute encoder to feed directly off of the drive pod, providing more accurate and reliable wheel orientation data. This also permits the sensors to be placed directly in the new component boxes with their circuit boards, simplifying the wiring.

2.2 Electrical Innovations

The modularity and usability of Singularity's embedded system has been markedly improved this year. The bulky and overly-complex power supplies of past years have been replaced with a distributed system, which allows for simple control of power to individual subsystems and dramatically reduces the volume of wires routed between boards. A new team-designed serial hub expands the number of available serial ports on the embedded computer from nine to sixteen, enabling communication with additional embedded subsystems and improving overall system integration. More advanced drive modes are implemented to take advantage of Singularity's unique drivetrain. Users can interact with the robot through a variety of mechanisms. An on-robot touchscreen UI provides an extremely intuitive and simplified set of controls, allowing users unfamiliar with the underlying design to comfortably interact with the system. For more complete access, a remote provides manual control via Singularity's many drive modes, and can be flexibly reconfigured for a variety of use cases.

2.3 Software Innovations

Taking advantage of Singularity's increased mobility, autonomous driving now features simultaneous strafing and arcing which allows the robot to traverse tight areas quickly, smoothly, and safely. A complete 360 degree sensor view is used to build a complete global map of the robot's surroundings over which a path to the robot's destination can be drawn and updated in real time. Advanced simulation functionality allows Singularity's software platform to be thoroughly tested in a variety of realistic scenarios. The JAUS protocol now supports additional messages, and is used to communicate with Singularity's interfaces, including the touchscreen, a web-based user interface, and the wireless remote control, enabling software-assisted control of the robot to compensate for operator error.

3 Design Process

The design process for Singularity was significantly improved over the past year to better track progress and allow for more extensive testing. Since Singularity was designed and built between 2009 and 2011, this year was spent upgrading and improving the robot based on its performance in the 2011 IGVC. As a result of the design process improvements, Singularity was operational earlier in the school year than any team robot since 2006.

3.1 Team Structure

Wisconsin Robotics is a student organization comprised entirely of volunteers. The team consists of graduate and undergraduate students from various engineering disciplines and computer science.

The team is broken up into three sub-teams: mechanical, electrical, and software as shown in Figure 3.1. Each sub-team leader is selected based on past involvement and level of experience. Weekly work sessions are scheduled such that all sub-teams have overlapping meetings once a week to facilitate communication. In addition, formal all-team meetings are held on a regular basis to discuss progress and major design decisions.

Team Leader Wes Miller '12, CS, French	
Mechanical Leader Andrew Cattell '14, ME	Electrical Leader Christopher Cline '13, EE
Grant Allen '13, ME, CS Nick Dimick GRAD, ME, ECE Noah Fang '14, ME Jake Kilbane '13, ME, CS Scott Munter '14, ME James Vanderheiden '13, ME	Ross Aiken '13, CE Nick Ambur '13, CE Theodore Dahlen '15, CE Ziliang Guo GRAD, CE Steve Hanson '12, EE Katelyn Kufahl '13 EE, Phys Andy Lee '15, EE Josh Petit '13, EE Ahmed Saif '15, EE Joe Schultz '12, EE Adam Vander Pas '14, CE Bryan Williamson '14, EE
Software Leaders Eric Kulcyk '14, CS Sam Roth '13, CMPE	
Nathan Deisinger '14, CS Alex Durgin '14, CS Alex Miller '13, CS Ben Moench '14, CS	

Figure 3.1 Wisconsin Robotics team roster

3.2 Project Planning

The planning process began immediately after the 2011 IGVC and continued through September. Proposed improvements were prioritized such that the robot would be competition-ready before the February registration deadline and major upgrades were scheduled to coincide with breaks in the academic year, when fewer members would be affected by the downtime.

Projects were broken down into three 10-week phases throughout the year, with progress update meetings occurring five weeks into each cycle, and demonstration meetings held at the end of each cycle to demonstrate the robot's capabilities on a practice course. These meetings not only helped track progress of projects more accurately than in the past, but also encouraged communication between members of different sub-teams and helped maintain member interest throughout the year.

3.3 Development

Although constrained to the same development cycles, the mechanical, electrical, and software sub-teams each used their own development processes suited to their specific task requirements. The mechanical team, whose work consisted of mostly hardware design, used a stricter phase-based development process consisting of computer-aided design, prototyping, production, and testing phases. Features were first modeled on a computer using SolidWorks, then prototyped. Only once the prototype proved effective in testing was a final version manufactured in-house. The electrical team followed a similar development process for their hardware design, using computer-aided design and prototyping whenever possible. Custom boards were designed using EAGLE and Altium Designer, computer-aided design printed circuit board layout tools.

Conversely, the software team used agile methodologies to allow for easier adaptation to the changing scope of their projects, including weekly driving tests which helped identify not only software bugs, but also mechanical and embedded issues that could occur during competition. Members were encouraged to routinely commit functional code revisions to the Mercurial repository, so that the code could be tested against other components. Due to the modular nature of the software platform, however, any outstanding issues detected would not interfere with the development and testing of other components.

New members – many of whom had little to no experience in engineering or software development – explored their talents and interests through hands-on training and guided group projects. In the spring, many of our new members took charge of their own projects, including the motor control system and simulation software. At times, experts on campus were contacted about how to best solve a specific design issue in the most efficient and effective way.

4 Mechanical Design

Singularity incorporates many innovative mechanical concepts that make the robot space and power efficient, easily upgradeable, and robust. The goal of Singularity's design was to provide an easily testable and serviceable platform for embedded and software systems, one which has the capacity to incorporate new and challenging control concepts with its fully omnidirectional drive capabilities in addition to being easy to service. Singularity's main mechanical features are: the omnidirectional drivetrain, chassis, and omnidirectional vision and sensor system. The omnidirectional drivetrain consists of four independent 'drive pods' which can each be turned and driven independently of one another. Singularity's chassis incorporates the bearing system for the drive pods, mounting for the powertrain and power systems, and support for the circuit boards and computer.

4.1 Drivetrain

The omnidirectional drivetrain has the capability to drive Singularity with more degrees of freedom than most conventional vehicles. Each pod is turned by a 24 V motor which has been geared down twice (externally with a gearing ratio of 35:11 and internally with a planetary gearbox to a ratio of 53:1) for a maximum estimated angular velocity of 35 RPM. Additionally, each pod is driven by a 24 V, 450 W motor that has been geared down internally to a maximum of 550 RPM. This was further reduced by a factor of 3:1 from the motor to the wheel by chain driven sprockets. The large gear reduction from the high power drive motor enables a high torque capacity for each wheel. With 10" wheels, the maximum speed of the robot is approximately 5.5 mph. Because the speed and direction of each pod can be independently controlled to minimize drag regardless of the robot's motion, Singularity is easily able to navigate rough and uneven terrain.

This year, the mounting system for the turn motors was upgraded to a significantly simpler and more reliable design. The four turn motors are mounted on two flat steel plates, which are attached upside down near the center of the frame. The simple mounting system consists of a single tensioning bolt which tensions both chains at once, and two slot-mounted bolts to secure the entire system. The slot in the mounting plate allows the motors to be removed without having to disassemble the chain. Mounting the motors this way frees space for sensors and electronics.

4.2 Chassis

Singularity's chassis was designed to emphasize the robot's omnidirectional capabilities, be structurally robust, and provide easy access to electrical and mechanical components. The chassis measures 36" long by 32" wide, which allows the robot to navigate between obstacles without altering its orientation and fit through standard doorways. The robot measures 56" tall, but in order to fit into a minivan for transportation, the top of the robot is removable. Without the top of the robot in place, the height is 34".

Most of the chassis is made of hollow, square steel bars MIG welded together in a strong and open design. Aluminum was chosen for the removable center column to save on weight and to reduce the amount of rust protection required. An aluminum payload tray runs through the center of the chassis, allowing the payload to be stored without blocking the view of any sensors.

Singularity's one-piece shell was replaced this year in favor of more permanent paneling around the chassis and with component boxes for easy access to electronic components. The permanent paneling provides better wire protection, since fewer wires are exposed during routine activities such as inserting or removing the payload. The boxes allow one or more boards to be accessed without exposing the entire robot, reducing the probability of damage. The boxes also provide protection from rain and supply adequate air flow to cool the electrical components. Two hinged boxes were built along the 36" sides of the chassis using ABS, acrylic, and aluminum angle stock. These materials were chosen for their light weight and aesthetics. The transparent acrylic allows for easy inspection of electrical components while also protecting the circuitry from the elements. The box lids slant

toward the outside of the robot slightly to prevent water from accumulating on top of the robot. Each of the boxes has two 80mm case fans positioned at opposite ends of the box to provide cooling for the electrical components.

To ensure the electronics in the large boxes do not overheat, a numerical heat transfer model was developed to predict the cooling performance of the fans in the large boxes. The model incorporated the box dimensions, material properties, and assumed a worst case solar radiation flux of 1000 W/m^2 . The model discretized the box into 25 smaller one-dimensional divisions and applied energy conservation equations to predict the temperatures in each section of the box. The internal heat transfer coefficients were calculated using duct flow correlations and each small division had an equal amount of heat applied to it that was a fraction of the total heat generated by the electronic boards and from solar radiation. Since fan curve data was not available to describe the volumetric flow as a function of flow resistance, the nominally rated volumetric flow value for only one of the two 80 mm fans was used to represent a worst case air flow scenario. Results from the model showed a steadily increasing temperature distribution as the air flowed through the box from the entry fan to the exit fan. The model also tested ambient outdoor temperatures up to 105 degrees Fahrenheit, resulting in 130 degrees Fahrenheit as an absolute worst case temperature in the box. Given that the model neglected emissive radiation (which would have cooled the box), assumed a worst case air flow condition, assumed worst case heat generation scenarios, and allowed for temperature ranges in excess of likely operating conditions, the model shows that cooling system operates well within acceptable bounds.

4.3 Serviceability

One of the main considerations in Singularity's design was serviceability. With an open and modular design, upgrades, repairs, and modifications can be done relatively easily and cost effectively. The hinged lids on the component boxes allow for easier access to the various electrical components. Access to the individual components themselves was considerably improved as well thanks to better component layout, specifically by avoiding stacked components. A separate box was manufactured to house the main computer to protect it from damage: since the computer uses off-the-shelf components, physical access to the board is rarely needed. The batteries are placed on a tray at the bottom of the robot between two of the laser range finders (LRFs). This design allows easy access to both the LRFs and the batteries.

The operator panel is mounted on the back of the robot underneath the emergency stop and above the payload tray. The operator panel includes a touchscreen, the main power switch, and two USB ports connected to the computer. The USB ports allow an operator to conveniently connect a remote control, flash drive, or wireless card. This layout allows the operator full access to the robot from a centralized location, while keeping more advanced controls, such as the ability to power off individual components, in the component boxes.

4.4 Sensor Placement

Singularity implements an omnidirectional optical system for lane and flag detection. A single camera is mounted facing directly upwards and pointed at the tip of a 6" diameter convex axicon (cone-shaped) mirror with an angle of depression of 18° . The shape, size, and position of the mirror were chosen to increase the proportion of pixels

that map to the nearest areas of the robot as compared to a spherical or parabolic mirror. The cone shape also allowed the mirror to be easily and inexpensively manufactured: the cone was machined out of aluminum to match the specifications required to achieve the desired field of view, then coated with a sheet of metallic DuraLar for reflectivity. Since the camera mount is centered on the top of the robot, the view of the ground immediately surrounding the robot is obscured. The height and angle of the mirror were chosen to reduce the size of this area. Also, since lines are visible on all sides of the robot, lines that pass through the obscured area can be interpolated from lines detected in the surrounding area.

In addition to the omnidirectional camera, an LRF is mounted on each side of Singularity. Since each LRF has a 180° field of view, the LRFs provide redundancy in many areas while minimizing blind spots. Improved accuracy and reduced interference was achieved by offsetting adjacent LRFs vertically 13 cm. The compass/accelerometer and GPS are located as close to the center of the robot as possible to ensure accurate measurements, and the compass is also placed away from the motors and steel frame to reduce distortion from any generated magnetic fields.



Figure 4.1 Singularity's new turn sensor gear train

The motor feedback system has been redesigned for improved robustness and accuracy. Each drive pod requires one quadrature encoder to measure the wheel speed, one potentiometer to measure the number of rotations of the wheel from the initial state, and an absolute encoder to measure the exact turn angle of the wheel. The quadrature encoders are mounted directly on the drive motor shaft and are clamped onto the motor mount. The clamping system was redesigned to more easily adjust the encoders as well as hold them in the proper position once located. The turn motors were modified to incorporate a gear train for the encoders and potentiometers, creating a 1:1 gear ratio with the drive shaft and eliminating any slippage. The mount went through several iterations, where the final design constrains the drive shaft relative to the gears to prevent any of the gears from slipping. Modifications

to the frame design also allowed mounting of the gear train directly in the component box using an ABS adapter. This keeps the wiring for the sensors contained in the component box and protects the entire assembly from outside elements.

5 Electronics Design

Singularity's electrical system is designed to provide a simple, robust interface between the high-level software system, sensors, and effectors in order to produce an efficient, functional, and safe platform. In addition to handling all of the low-level sensor interfacing, the electrical system provides power to all system components.

5.1 Sensors

Singularity uses an AVT Guppy F-080C digital camera for lane detection. This camera provides 1024x768 pixel, color images at 30 FPS via a IEEE 1394a connection, which also transmits power. Singularity has four SICK

PLS201 laser range finders, providing a complete 360° plane of view around the robot. Other sensors include an Ocean Server Technology OS4000-T digital compass with built-in accelerometer, a Garmin GPS 18X-5Hz GPS receiver with one meter accuracy, and quadrature and absolute drive pod encoders.

5.2 Embedded Control System

At the center of the embedded control system is a TS-7800 ARM-based single-board computer (SBC). This runs a customized Gentoo Linux distribution and provides a simple interface between the main computer and the rest of the electrical system. It communicates with the main computer via a Gigabit Ethernet connection, enabling simple, reliable high-speed data transfer. To communicate with Singularity's multitude of sensors the system offers a total of 16 serial ports. The ARM SBC provides 9 of these, and the rest are provided by a dedicated Atmel XMEGA microcontroller acting as a serial hub.

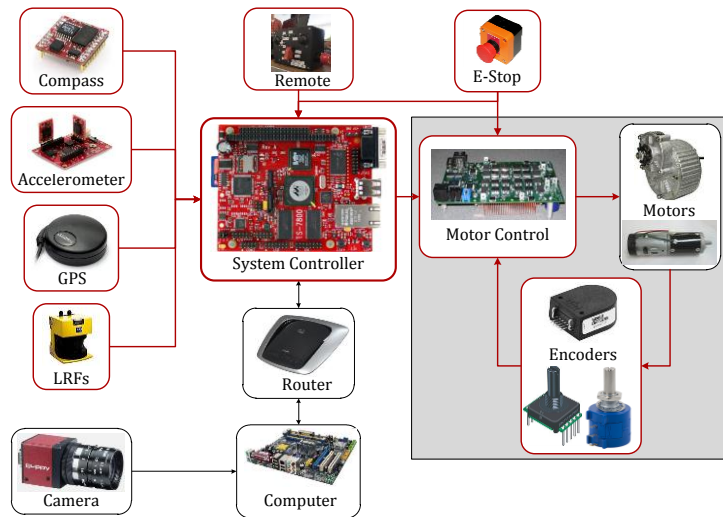


Figure 5.1 Embedded system diagram

5.3 Main Computer

Singularity has a high-performance computer featuring an Intel Core i5 2500 CPU, 8GB DDR3 RAM, and a 60GB SATA III Solid State Drive, all attached to a microATX motherboard and enclosed in a custom-made case. The quad-core i5 processor was chosen for its high clock speed of 3.3GHz, Turbo Boost (which intelligently overlocks the CPU automatically when necessary), and relatively low power consumption. The SSD provides faster read access than a traditional hard drive and is more resistant to shocks and vibrations experienced during robot operation. An M4-ATX power supply allows the computer to shut down gracefully after the power is turned off, and can be used to monitor the state of the batteries.

5.4 Motion Control

Continuing Singularity's emphasis on modularity, a team-designed motor control board independently controls each drive pod. Each of the four motor controllers is designed to drive two 500W motors and receive feedback from up to two quadrature encoders, one absolute encoder, and one potentiometer. All functions related to closed loop control are performed on the motor controllers, and the central embedded SBC is only responsible for the coordination between motor controllers. This reduces the computations required on the central SBC, and prevents a failure of the Linux system from affecting the drive pods.

To take advantage of the freedom of movement that Singularity offers, various driving modes have been implemented: 'Crab' drive, where each pod turns and drives in unison, and arcing mode, where a point is

specified relative to Singularity and each drive pod aligns and drives itself such that the robot can turn around that point with minimal lateral drag on each wheel. This mode also allows for rotating in place by specifying the point of rotation to be Singularity's center. Other driving modes, like Ackermann steering, where the 'back' two pods are locked pointing 'forwards', and the front two pods turn, are possible with this configuration but have not been implemented, since arcing mode provides a greater range of motion.

5.5 Power System

Singularity uses a brand-new distributed methodology to power all of its components. Rather than powering all of the boards on the robot with a single centralized power supply, unregulated battery power is routed through a series of switches to each individual board, each of which has an inexpensive switching DC-DC regulator on-board. The routing system allows for fine-grained control over which boards are turned on, improving battery life during long testing sessions when only particular subsystems are needed. A single common supply voltage (24 V) reduces the number of wires needed to distribute power throughout the system, and allows each board to convert down to a voltage tailored to its particular usage (5 V, 3.3 V, 1.8 V, etc). The use of inexpensive, readily-available power supplies ensures that the robot will not be incapacitated for large periods of time if a single power supply fails. A main power switch accessible from the outside of the robot allows for the entire robot to be turned on or off without opening the component boxes, which is desirable when operating in inclement weather.

Each board in the system is fused to minimize the possibility of local electrical damage, and to isolate the failed component from the rest of the system. Overvoltage protection is present on boards with more sensitive electronics. A high current breaker cuts off power to the entire robot in the case of extraordinary power usage.

5.6 Electrical Safety Features

Safety was an important consideration when designing Singularity. The emergency stop line directly controls the disable pin on the H-Bridge drivers on each of the motor controllers, completely disabling all motors regardless of the state of other control logic. This line can be triggered by the on-robot e-stop switch, the remote e-stop switch, which communicates via digital I/O on a pair of Zigbee modules, or by software on the embedded computer. This allows for humans, or even the robot itself, to activate the e-stop whenever it is deemed necessary.

The warning light system consists of 24 independently-controlled high-brightness LEDs, which can provide additional information such as the robot's current direction by varying the pattern of illuminated LEDs. Additionally, the warning light brightness can be adjusted for indoor or outdoor use, ensuring that the lights are clearly visible in any environment without becoming distracting while servicing the robot.

5.7 User Interaction

Singularity can receive input from and provide feedback to its operators through a variety of mechanisms, ranging from an Xbox 360 controller to a web interface. An Xbox 360 controller can be plugged directly into the main computer, or any computer on Singularity's wireless network, to manually control the robot using any of the implemented driving modes.

A wireless controller was also created by customizing an original Xbox controller. The internal electronics were removed and a wireless Zigbee module was installed to extend the communication range, along with a custom control board to properly translate the various buttons and joysticks into drive commands and an e-stop signal for Singularity.

A touchscreen, powered by a team-designed control board, provides on-robot access to a custom-made browser-based user interface, providing real-time information about the robot's current state as well as simple configuration options. A more advanced version of the web interface designed to be viewed on a computer screen is also available over Singularity's wireless network.

Due to the difficulties of manually driving a robot or any other motorized vehicle, the team devised a software-assisted driving mode. In this mode, a human operator provides a desired direction and speed to the software system, which can adjust the commands forwarded to the motors to account for detected obstacles in the desired path. This helps reduce the chances of human error, such as driving the robot into obstacles, and makes operation by untrained users considerably safer.

6 Software Design

Singularity's software platform is a heavily modified version of the Robotics Simulation and Control Lab (RSCL) package originally developed by the team in 2005. Java was selected as the platform of choice because of its widespread adoption in the University of Wisconsin-Madison's computer science department. This was intended to make it easier for new members without extensive programming experience to navigate RSCL's codebase and begin contributing in meaningful ways.

In contrast with 2011's reactionary approach to course navigation, the software team this year returned to the global map architecture first conceptualized two years ago. A global navigation mesh is built using the data from the four laser range finders and the camera. A* search is run over this mesh to find a path to each target while avoiding obstacles. The construction of the navigation mesh and the generation of a path take several steps: localization, feature extraction, mapping, path planning, and driving.

6.1 Localization

Accurate localization is critical in maintaining a consistent global map. The digital compass, four turn encoders, and four drive encoders are used to provide a dead reckoning estimate as to where Singularity is. To refine this position, the iterative closest point¹ scan matching algorithm is used to align two different LRF scans. Iterative closest point matches two sets of points together by rotating and translating one set onto the other. By matching consecutive LRF scans, a very accurate position is obtained. Because iterative closest point finds a local optimum solution, an initial dead reckoning position increases the likelihood of finding the correct location of the robot.

¹ Paul J. Besl and Neil D. McKay. 1992. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* 14, 2 (February 1992), 239-256. DOI=10.1109/34.121791

6.2 Feature Extraction

Singularity uses data from its omnidirectional camera and its four LRFs to detect a wide variety of obstacles that may appear on the IGVC course. These can be generalized into three main types of obstacles: lines, flags, and physical objects sitting on the ground. Physical objects are detected using the LRFs. Where the LRF data overlaps, the closest obstacle is chosen. Lines and flags are identified using Singularity's modular vision system, written in C++ for improved performance, and broken down into stages for easy adaptation to new types of obstacles.

The first stage of the vision algorithm is a rectification operation called unwarping. This operation reverses the distortion caused by the shape of the mirror, resulting in a simulated birds-eye view of the robot's vicinity. A depth map is used to compensate for complex distortions caused by imperfections in the catadioptric system. The points from the depth map are loaded into a k-d tree² and then the closest point in the output image to each point in the depth map is located using the k-d tree. Then, for each input image, each point on the corresponding output image is mapped to the corresponding point on the input image.

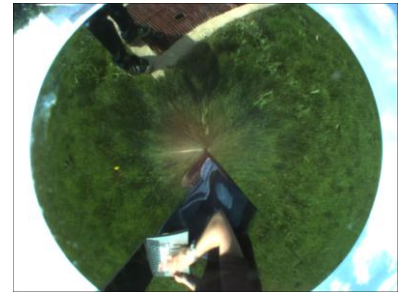


Figure 6.1 Raw camera image



Figure 6.2 Unwarped image

Line detection, based on the Hough Transform³ and the Canny Edge Detector⁴, is performed on the unwarped image. This is to ensure that lines appear linear in the image, making them easier to accurately identify. Candidate lines are checked to ensure they meet a minimum length before they are considered. If similar lines are detected in subsequent images, the confidence rating of the line increases.

Flag detection is also performed on the unwarped image using a Haar Cascade classifier. As a preprocessing step, the image is filtered to permit only pixels with a similar hue to the flags to be detected. The classifier is run on the filtered image. The points identified by the classifier are marked as flags of the color in question. This process is repeated for each color of flags to be detected. Once a flag is identified, its position is calculated based on the expected height of the flag off the ground. A line is then drawn from the flag to the side of the course, preventing the robot from driving on the wrong side of the flag.

Features are extracted from the combined LRF and line data before the map is created so that obstacles can be characterized using minimal data. The goal of the feature extraction algorithm is to reduce the search space used

² Jon Louis Bentley. 1975. Multidimensional binary search trees used for associative searching. *Commun. ACM* 18, 9 (September 1975), 509-517. DOI=10.1145/361002.361007

³ Richard O. Duda and Peter E. Hart. 1972. Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM* 15, 1 (January 1972), 11-15. DOI=10.1145/361237.361242

⁴ Canny, John; , "A Computational Approach to Edge Detection," *Pattern Analysis and Machine Intelligence*, IEEE Transactions on , vol.PAMI-8, no.6, pp.679-698, Nov. 1986 doi: 10.1109/TPAMI.1986.4767851

for pathfinding while retaining important information about the world. The feature extraction system has a single absolute constraint along with a single relaxation of that constraint:

Constraint: Any two points which are closer together than the width of the robot cannot be traversed.

Relaxation: Any objects directly behind non-traversable zones can be ignored.

From those guidelines a natural greedy algorithm arises which connects data points together that are located closer together than a threshold value into lines. Any data points which lie 'behind' these lines are simply ignored.

6.3 Mapping

Each of the obstacle features extracted from sensor data is expanded into a polygon using the Minkowski sum⁵. The resulting polygons represent areas in which the robot cannot enter without risking collision. To enable simple pathfinding which avoids collisions with these polygons, a navigation mesh composed of triangles is created such that each edge of a polygon is the edge of a triangle in the mesh. The triangulation algorithm employed is a randomized incremental Delaunay construction⁶. Delaunay triangulation involves creating new, more uniform triangles from quadrilaterals created by two existing adjacent triangles. It also ensures that the smallest angle of any triangle in the triangulation is maximized. In practice, a Delaunay triangulation results in a more uniform distribution of candidate path finding nodes than a generic triangulation. The triangulation is only semi-Delaunay because of an added requirement that all of the edges of the obstacle polygons are edges of triangles in the mesh. These special triangle edges are deemed "invalid" because crossing them would result in the robot getting too close to an obstacle.

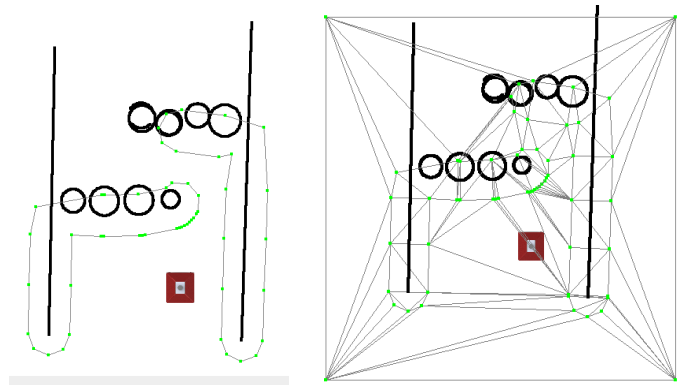


Figure 6.3 A bird's eye view showing Singularity as a red square with simulated walls and barrels as seen by the LRFs. Green denotes feature-extracted data points extruded into non-traversable polygons (left). Obstacle polygons are triangulated into a navigation mesh (right). Each valid triangle edge is used for pathfinding.

6.4 Path Planning

Since non-navigable regions are bordered by invalid triangle edges, each triangle that can be reached without crossing an invalid edge is navigable. The final representation of the drivable space that the A* search⁷ is run over

⁵ Behar, Evan; Lien, Jyh-Ming; , "Fast and robust 2D Minkowski sum using reduced convolution," Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on , vol., no., pp.1573-1578, 25-30 Sept. 2011 doi: 10.1109/IROS.2011.6094482

⁶ Guibas, L.; D. Knuth; M. Sharir (1992). "Randomized incremental construction of Delaunay and Voronoi diagrams". *Algorithmica* 7: 381–413. doi:10.1007/BF01758770

⁷ Hart, P. E.; Nilsson, N. J.; Raphael, B. (1968). "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". *IEEE Transactions on Systems Science and Cybernetics* SSC4 4 (2): 100–107. doi:10.1109/TSSC.1968.300136

is generated by splitting the triangle edges into discrete path finding nodes. A* search was chosen because of its efficiency and simplicity. A custom A* heuristic is employed that negatively weights paths which proceed in the opposite direction of the robot's current vector. The combination of the global map and the A* heuristic help the robot avoid looping back to areas it has already traversed.

An optimization made in the pathfinding process was to choose coarse candidate nodes for A* (approximately four meters apart), and refine the result to find the 'best' path: the shortest path to the target which stays a reasonable distance (50 cm) away from obstacles whenever possible and avoids sharp turns. More specifically, for every three consecutive points in a candidate path, a nonlinear relation that represents the sharpness of the angle the three points compose and the distance of each point from the nearest obstacle is minimized. The solution to this minimization is approximated by considering all paths using a discrete set of points along each edge centered on the A* node which was chosen. The total cost of finding the optimal path is $O(n^3 * m)$ where n is the number of candidate points for each line and m is the number of points in the A* path. An approximation of the full optimization is used that breaks the computation into multiple steps. At each step a fixed number of candidate points are optimized, leading to a fixed computational cost at each step. Since the path does not need to be recomputed every time new data is received, multiple cycles can be afforded to optimize the path if the result is less latency for an individual cycle.

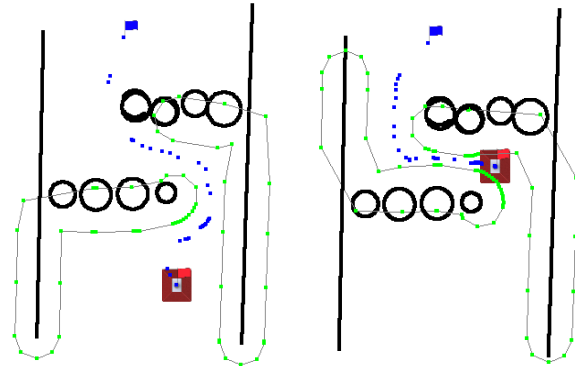


Figure 6.4 The final optimized path from the red flag to the blue one is shown as a set of blue points. Singularity's omnidirectional mobility allows it to drive the path quickly and accurately.

6.5 Driving

The aforementioned optimizations help to ensure that the paths chosen by Singularity are easily traversable. Due to the omnidirectional drivetrain of Singularity, the direction the path takes is not a concern; however, the abruptness of changes to the robot's vector is. As the turning speed of each drive pod is limited, if the path changes direction too quickly, deviation from the intended path can occur and collision could result. To compensate, we use a time-to-contact driving approach which maintains a speed such that the nearest obstacle in the robot's path is 4 seconds away. Furthermore, if the robot ever enters a non-navigable area (as outlined by the obstacle polygons), it stops immediately as a safety measure before reevaluating its path.

6.6 JAUS

RSCL has used a derivative of the OpenJAUS Java library, dubbed libjaus, for several years now, with each subsequent year seeing further work to increase the number of messages the library supports in addition to updates to ensure the library matches the latest published specification. This year, the team decided to employ

J AUS more extensively between Singularity's constituent subsystems, as well as for communication with additional control interfaces.

One new application of J AUS was in the remote control receiver module developed by the electrical team. In conjunction with the software team, the electrical team created a minimalistic J AUS library for use on an Arduino platform, which is used to receive and translate messages from the remote control into their J AUS equivalents before being relayed to RSCL for software-assisted driving. The creation of a new J AUS library also provided a second benchmark to test the compliance of libjaus and helped in training the new library maintainers as they worked to understand the specification.

The software team also worked to convert existing GUI control programs to use J AUS, replacing the custom messages used in previous years before upgrades and extensions to libjaus were completed. In addition to the existing programs, the team decided to construct a new, more user friendly interface for controlling Singularity. The frontend is an HTML5 website using Server-Sent Events to receive data pushed from the Java backend. The backend uses the same J AUS library used by RSCL to communicate with the robot over the network. Two variations of the frontend were created: one for use in a regular browser, and one optimized for the touchscreen mounted on the robot.

6.7 Simulation

An extensive simulator was built from scratch to aid in testing RSCL. The simulator includes the ability to simulate motor movement, LRF and line obstacle detection, compass orientation, GPS location, and the physical dimensions of the robot. Both systemic and random error is simulated for most sensors to determine how robust each algorithm is to bad input and how well the localization algorithm performs. Maps can also be created from data recorded during testing and replayed at a later time.

7 Performance

Over the course of the year, Singularity has been rigorously tested to ensure it meets or exceeds the requirements of the IGVC and the design goals for safe and reliable operation set by the team.

Although the maximum speed was reduced to increase torque, it is still fast

enough to easily complete the course in the allotted time. The GPS accuracy is highly dependent on the number of visible satellites. The expected performance on the competition field should exceed these results due to reduced interference from surrounding buildings.

Metric	Design Goal	Recorded Value
Battery Life	3 hours	3 hours
GPS Arrival Accuracy	1m	~1.5m
Maximum Speed	10 mph	5.5 mph
Obstacle Detection Distance	10m	10m
Ramp Climbing Ability	15% gradient	15% gradient
Reaction Time (includes I/O)	150ms	~92.5ms

Table 7.1 Performance test results

8 Cost Summary

Ideally, the team would design and manufacture all components on the robot for the experience it would provide. However, several components are too expensive to make in small quantities, require access to specialized equipment, or are simply beyond the level of undergraduate expertise. These components, such as motherboards, motors, the GPS, and others, were purchased, saving both time and money. The team designed and manufactured a vast majority of the components on Singularity including the frame, operator control unit, and motor controllers. Most of the software is written entirely by team members. In many cases, code was imported from various open source projects and is continually updated and improved upon.

System	Item	Qty	Cost	Our Cost
Mechanical	Raw Materials	-	\$985	\$985
	Drive Components (Sprockets, Chain, etc.)	-	\$490	\$490
	Motors	8	\$703	\$703
Computer	Motherboard, Memory, SSD	1	\$447	\$367
	Wireless Router – Linksys WRT320N	1	\$60	\$60
Vehicle Control	TS-7800 SBC and Peripherals	1	\$470	\$420
	Motor Controllers – Team Designed	4	\$600	\$300
	Operator Control Unit – Team Designed	1	\$300	\$300
	Warning Lights – Team Designed	1	\$180	\$80
	Wire and Interface Hardware	-	\$120	\$120
Sensors	SICK PLS201 Laser Range Finder	4	\$12,000	\$300
	AVT Guppy F-080	1	\$1,025	\$1,025
	Garmin GPS 18x-5Hz	1	\$160	\$160
	OS4000-T Compass	1	\$250	\$250
	Encoders and Potentiometers	-	\$300	\$100
Power	ATX Power Supply – M4-ATX 250W	1	\$100	\$0
	Batteries – 75Ah 12V Deep Cycle Lead-Acid	2	\$120	\$120
Total			\$18,310	\$5,780

Table 8.1 Component cost breakdown

9 Conclusion

Singularity was designed with military and commercial applications in mind, and with the hope of advancing the field of unmanned ground vehicles. It was designed to meet and exceed the challenges presented by the 2012 Intelligent Ground Vehicle Competition, and to highlight the strengths of the Wisconsin Robotics team. Singularity’s modularity, versatility and efficiency should prove to be an ideal platform for autonomous vehicle research and development.