University of Wisconsin-Madison

Wisconsin Robotics 00101010

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.

M Zinn

Professor Michael Zinn
Department of Mechanical Engineering

Table of Contents

1 Introduction

Singularity returns to the 2013 Intelligent Ground Vehicle Competition with significant changes for its third year. Wisconsin Robotics focused this year on improving Singularity's overall robustness while continuing to innovate aspects of the mechanical, electrical, and software systems. As in years past, the team sought to go beyond the challenges of the competition to 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 this year. Mechanical and electrical systems that exhibited performance issues in past years have been redesigned and replaced. Within the software platform, an entirely new vision system has been implemented, in addition to significant optimizations for localization, mapping, and navigation.

2.1 Mechanical Innovations

Several new mechanical components have been installed on Singularity to increase reliability. The first major improvement is the installation of mechanical slip rings to transmit power and encoder data through rotating connections where the drive pods connect to the frame. The previous design was limited in the number of rotations the drive pods could undergo because wires were passed through the drive pod shafts and were susceptible to stretching and tearing. The new slip rings allow for unlimited number of rotations with no chance of the wires tearing, since wires do not pass completely through the drive pod shafts. This increases the reliability of the system and also removes the need for software to occasionally halt the robot to unwind the drive pods. The drive encoders and mounting system have also been redesigned to improve reliability and drivability.

2.2 Electrical Innovations

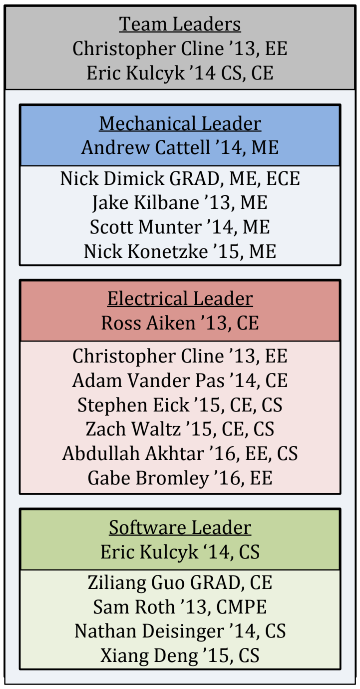
Several of Singularity's core electrical systems have been updated this year. An innovative optical system transmits quadrature encoder signals reliably through the new slip rings. New team-designed motor controllers improve the robustness and modularity from the previous generation. A new custom wireless emergency stop provides reliable vehicle monitoring and control in a user-friendly 3D-printed package.

2.3 Software Innovations

This year innovations are being introduced which improve Singularity's overall software strategy. The most significant of these is the vision system, which was redesigned to be able to detect lines accurately in a variety of lighting conditions. Improvements were also made to Singularity's driving algorithms to allow for more smoothly navigating a chosen route. This year an inertial measurement unit (IMU) was incorporated into the localization process, complementing data from the encoders. The JAUS implementation used on Singularity has also been rewritten to be simpler and more extensible. The reaction time of Singularity has been significantly reduced. Finally, this year Singularity is able to autonomously take full advantage of the unique drive modes that its omnidirectional design allows.

3 Design Process

The team's design process was revised this year to facilitate improved inter-team communication and progress tracking. The mechanical and embedded teams focused mostly on incremental updates addressing specific issues observed with past systems. The software team focused on testing systems regularly throughout the course of the year; this was enabled by maintaining Singularity's basic driving functions throughout various mechanical and electrical revisions.



Team Leaders
Christopher Cline '13, EE
Eric Kulcyk '14 CS, CE

Mechanical Leader
Andrew Cattell '14, ME
Nick Dimick GRAD, ME, ECE
Jake Kilbane '13, ME
Scott Munter '14, ME
Nick Konetzke '15, ME

Electrical Leader
Ross Aiken '13, CE
Christopher Cline '13, EE
Adam Vander Pas '14, CE
Stephen Eick '15, CE, CS
Zach Waltz '15, CE, CS
Abdullah Akhtar '16, EE, CS
Gabe Bromley '16, EE

Software Leader
Eric Kulcyk '14, CS
Ziliang Guo GRAD, CE
Sam Roth '13, CMPE
Nathan Deisinger '14, CS
Xiang Deng '15, CS

Figure 3.1 Wisconsin Robotics team roster

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.

3.2 Project Planning

This year the team took a more formal approach to project management, using the open source Redmine software, running on a team server. This system facilitated the creation of discrete sub-projects and tasks, and delegation to various team members. Each team member could view their own tasks as well as team-wide issues. Redmine helped to improve overall transparency and communication throughout the team, increasing accountability and accelerating the design process.

3.3 Development

The Mechanical, Embedded, and Software teams each used separate development methodologies in order to meet their design deadlines. The mechanical team, whose work consists of mostly hardware design, used a 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 tools like EAGLE and Altium, and prototyping whenever possible. After receiving the blank printed circuit boards from a PCB fabrication house, the team members then assemble, test, and perform any programming and debugging that may be necessary.

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 features that make it an efficient, robust, and unique platform. The entire design has been built to be fully omnidirectional. The drivetrain system consists of four independently rotated and driven drive pods that provide Singularity with the ability to make a variety of space-efficient maneuvers. Singularity also has the ability to see in all directions with the use of four laser range finders (LRFs) and a unique camera setup that utilizes a single camera and cone-shaped mirror. Other mechanical components such as the chassis, electrical component boxes, and sensors housing have all been designed to be serviceable and modular for easy upgrades.

4.1 Drivetrain

Singularity's drivetrain system consists of four drive pods that can be independently rotated and driven. The independent turning is accomplished with the use of four 24 V motors which have been geared down to provide a maximum angular velocity of 35 RPM. The independent driving is accomplished with four 450W drive motors geared down to provide a maximum driving speed of 5.5MPH with 10" diameter wheels.

The ability to control the drive pods independently provides Singularity with numerous methods of driving. Singularity can translate in any direction (turning all drive pods in the same direction at once) and perform zero-radius turns (turn in place), as well as perform simultaneous rotational and translational movement. This allows Singularity to be as space-efficient as possible when navigating obstacles.

4.2 Chassis

Singularity has a welded tubular steel chassis that is designed to provide a durable, easily accessible platform on which to build and mount electrical and mechanical components. The chassis is comprised of MIG welded 0.75” square steel bars, as well as a removable center column of aluminum. An aluminum payload tray is mounted in the center of the robot to support the central computer and removable payloads without blocking any exterior sensors. The chassis measures 36” by 32”, which allows it to fit through doorways and maneuver around obstacles in any orientation. The maximum height of Singularity is 56” but can be reduced to 34” for easier transportation by removing the center column.

Two component boxes are mounted above the main frame to provide an accessible and protected layout for the motor controller boards, embedded logic boards, and power distribution circuitry. They also contain the slip ring and encoder assemblies for the drive pods. The boxes are constructed of ABS, acrylic, and aluminum angle stock to provide durability and aesthetics, and to facilitate easy visual inspection of the various components. The component boxes are cooled using 80 mm case fans installed at each end of each box. Numerical simulations were conducted to ensure the electronics in the boxes do not overheat. Using 25 one-dimensional duct-like divisions, the heat added to each duct was calculated from estimates of the maximum heat dissipated by the electrical boards and from incident radiation on a hot summer day. Internal duct correlations were used to determine heat transfer coefficients and the mass flow was determined from the data sheet for the two 80mm computer fans installed at opposite ends of the box. The model was analyzed using ambient air temperatures ranging from 80 to 105° F and resulted in a maximum temperature in the box of 130° F. Since the original model assumed complete absorption of the sun’s incident radiation on a worst-case estimate for all heat sources, this maximum value is well within acceptable bounds.

4.3 Serviceability

Singularity was designed to be easy to service. Access to most of the electrical components can be achieved by lifting the lid on the clear electrical boxes mounted to the robot’s frame. In cases where visual inspection is sufficient, no parts need to be opened. To allow quick access to the onboard computer and other peripheral ports a drop-down paneling section has been installed. Miscellaneous sensors such as the camera, LRFs, IMU, and GPS can be accessed by removing a single snap-on/press fit paneling component. The batteries are accessible by sliding out the battery tray located behind one of two laser range finders on the bottom of the robot. The operator panel is conveniently mounted on the back of the robot underneath the emergency stop and above the payload tray. 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 subsystems, in the component boxes.

4.4 Sensor Placement

In order for Singularity to detect lanes and flags, an omnidirectional optical system was implemented. A camera is mounted directly upwards and is pointed at the tip of a convex, cone-shaped mirror, as shown later in Figure 6.1. The mirror is a cone machined out of an aluminum block, coated with a sheet of metallic Dura-Lar for reflectivity.

Since the camera is mounted on the center of the robot, the height and angle of the mirror were carefully chosen to maximize the view of the ground plane near the robot.

In addition to the omnidirectional camera, a total of four LRFs are mounted on Singularity, one on each side. Each LRF has a 180° field of view, so the data from all four can be combined to eliminate blind spots and increase obstacle detection redundancy. The IMU and GPS are placed close to the center of the robot in order to achieve the most accurate readings as well as to reduce distortion from the steel frame and power electronics.

4.5 Slip Rings

Singularity has recently been equipped with electromechanical devices called slip rings to replace the old method of threading power and encoder cables down through the shafts of the drive pods. Slips rings allow for the passage of current through a rotating connection without the need for a single continuous wire. This is accomplished by connecting the stationary end of the wire to spring-loaded carbon brushes that make contact with a continuous brass ring around a rotating shaft. The brass ring is connected to the end of the wire that rotates with the drive pod. Because the carbon brushes are constantly in contact with the brass ring as the shaft rotates, current is conducted continuously without interruption and without the need for a continuous length of wire. This has the advantage of allowing the drive pods to make an infinite number of rotations since there are no wires to stretch and tear. This is also advantageous because the software does not need to monitor the number of rotations the drive pod has made and there is no need for periodic unwindings to be performed. On the embedded side, the slip rings remove the need for a potentiometer since the drive pod can rotate an unlimited number of times without damage. This decreases the complexity of the embedded and software systems. Figure 4.1 shows a picture of the slip rings installed on Singularity.

The slips rings on Singularity are 1” diameter through-bore rings with four contacts rated at 20 amps each supplied from United Equipment and Accessories. Two of the brass rings are used for providing power to the drive motor while the other two contacts are used to power the drive shaft encoder. To receive the drive shaft encoder data, an optical transmitter and receiver has been designed and installed inside the shaft of the drive pod, as described in Section 5.4.1. To record the absolute position of the drive turn motor, a plastic gear is mounted to the section of the drive pod shaft that extends above the slip rings and spins an idler gear connected to the turn motor encoder.

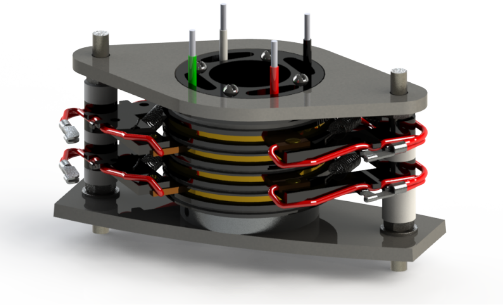


Figure 4.1 Slip Rings Assembly

Overall, the slip rings improve Singularity’s robustness and reliability by allowing an infinite number of turns to be made by the turn motor. This reduces complexity for both the embedded and software systems and also eliminates the possibility of wires twisting and tearing.

5 Electronics Design

Singularity's power and computer systems are designed to provide a reliable yet simple interface between the high-level software system, sensors, and output devices to produce an efficient, functional, and safe platform. The electrical system provides power to all systems onboard the robot as well as handling sensor and output data transfers between subsystems of Singularity.

5.1 Sensors

Singularity uses a collection of sensors to provide spatial data to its systems. A Sentech TC33 camera provides 640x480 pixel images at 60 fps used for lane detection over a USB2.0 connection. Four SICK PLS201 laser range finders provide a complete 360° plane of view. Other sensors include an Ocean Server Technology OS4000-T digital compass with built-in accelerometer, a Garmin GPS 18X-5Hz GPS, and quadrature and absolute encoders.

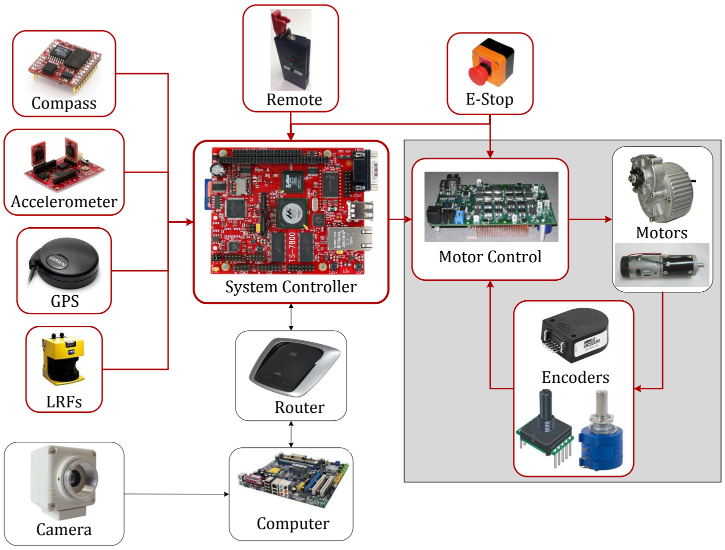


Figure 5.1 Embedded system diagram

5.2 Embedded Control System

The central hub for all of the sensors and low-level control systems on Singularity is a TS-7800 ARM-based single-board computer (SBC). Running Debian Linux, this system acts as an interface between the main software system and the rest of the electrical system. It communicates with other devices, such as the main computer, over a Gigabit network. All of Singularity's sensors, except for the camera, communicate with the embedded system over TTL or RS232 serial. To interface with all of these devices the embedded control system offers 16 serial ports, 9 of which are provided by the SBC itself; 7 additional ports are provided by a dedicated Atmel XMEGA microcontroller acting as a serial hub.

5.3 Main Computer

To provide ample processing power for its advanced sensing and control algorithms, Singularity has a custom-built high-performance computer. This computer consists of an Intel Core i5 2500 CPU, 8GB of DDR3 RAM, and a 60GB SATA-III solid-state drive (SSD), all attached to a microATX motherboard inside a custom-made case. The CPU was chosen due to its high clock speed (3.3GHz, with Turbo Boost, which allows it to intelligently overclock itself based on usage and thermal headroom), as well as the presence of four cores. The SSD provides much faster access times than a traditional mechanical hard drive, and due to the lack of moving parts, is better suited to the abusive environment that a mobile robot provides. Using an SSD, the main computer can boot in under 30 seconds. The M4-ATX Power Supply Unit provides the necessary ATX voltages from the battery supply, and automatically sends a shutdown command to the computer when power is cut to the rest of the robot.

5.4 Motion Control

5.4.1 Slip Rings

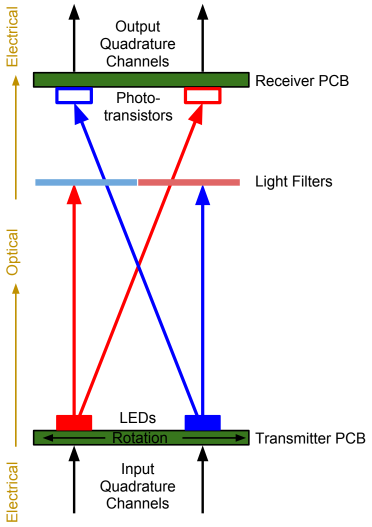


Figure 5.2 Simplified Diagram of Optical Quadrature Encoder

In order to avoid the noise produced by a mechanical slip ring connection, encoder data from the drive motors is fed back through a frequency-multiplexed optical connection, as shown in Figure 5.2. Power is transmitted to the motors and quadrature encoders via four slip ring conductors. Inside the drive pods along the axis of rotation, a small team-designed PCB converts the two data channels from the quadrature encoder into optical signals at two different wavelengths, using LEDs of two different colors. The light that encodes the two quadrature channels then passes through one of two wavelength-specific optical filters, and then reaches one of a pair of phototransistors on the receiving PCB which converts the optical signal back to an electrical signal. In this way, the electrical input to the transmitter PCB on the rotating drive pod is mirrored by the electrical output from the receiver PCB on the non-rotating chassis, which then gets sent to the motor controllers. The LEDs and phototransistors are arranged to maintain optical connections at any given drive pod angle, thus enabling transmission of quadrature encoder signals through an infinite turning range.

5.4.2 Motor Controllers

One of the projects undertaken this year was the redesign of the motor controllers. The motor controller performs closed loop control for an individual pod, with the central embedded SBC handling the higher-level coordination of the motor controllers. The primary motivations for the motor controller redesign were improving modularity and robustness. The team-designed motor controller board used in previous years has been split into three smaller boards. Two of the boards, dubbed the ‘power’ boards, can each drive one 500 W motor. The third board, dubbed the ‘logic’ board, includes an ATxmega128A1 microcontroller, which communicates with the embedded control system and runs motion control loops via feedback from up to two quadrature encoders, one absolute encoder, and one potentiometer. Improvements to the actual circuitry include the use of various forms of overvoltage protection, reverse-polarity protection, and opto-isolation to create a significantly more robust system. Each pair of ‘power’ boards shares a team-designed buck-boost converter that allows the motor controllers to run off of a wide range of input voltages (anywhere from 10-30 V). This was created with the same modular ideology as the rest of the system, so it is extremely easy to replace this board in the event of a critical component failure.

5.4.3 Drive Modes

Singularity’s unique design provides many degrees of freedom, allowing it to implement many different driving modes. The main mode used by Singularity is arcing mode, where the system calculates individual direction and speed values for each drive pod in order to maximize driving efficiency. This mode allows for simultaneous rotational and translational movement in any direction. Other drive modes include “Crab” drive, where each pod turns to the same heading and drives at the same speed, “Ackermann” steering, where the ‘back’ two wheels have a fixed heading and the ‘front’ two turn in unison, and “Double-Ackermann”, where the ‘back’ two wheels turn in unison and the ‘front’ two wheels turn in unison. Since the arcing mode can simulate each of the other modes, given the proper inputs, it is the mode that the team has elected to use both for autonomous and manual control.

5.5 Power System

To provide a robust, simple, and modular system, Singularity has a fully distributed power system. The unregulated 24 V battery voltage is distributed throughout the robot via a central switching system. Each switch controls power for a functional group of components, and each group is individually fused, so that a short circuit in one group does not take down the entire robot. In addition to the fuses on each line, there is a high-current auto-reset breaker, to protect against wiring faults or otherwise extraordinary power usage. Each electrical board can then regulate the 24 V using switching power supplies to generate the voltages required (5 V, 3.3 V, and 1.8 V). This conversion is done through the use of commercial off-the-shelf switching regulators, which are both readily available and easily replaceable should one fail. Singularity’s entire electrical power system is controlled by two switches: an internal toggle switch and an external key switch. Both switches need to be enabled for power to be supplied to the robot, but the external key switch can be taken out of the loop during testing.

5.6 Electrical Safety Features

Singularity’s electrical systems were designed with safety as a top priority. The emergency stop system directly disables the H-bridge drivers on the motor controllers and subsequently the motors themselves, regardless of their state. The emergency stop can be triggered both with the on-board e-stop switch, as well as the wireless e-stop switch, which communicates via 2.4 GHz XBee modules that provide robust performance with automatic channel-hopping and 128-bit AES encryption. If the wireless module loses power or goes out of range, a timeout is triggered to halt the robot.

The warning light system is implemented with 24 independently controlled high-brightness LEDs. These LEDs are used to distinguish between a possibly moving (flashing) and a stationary (solid) robot through their blinking pattern. Because Singularity’s omnidirectional mechanical platform may make discerning its direction of movement difficult for observers, the warning lights can also be used to indicate the direction of movement of the robot. Their brightness can be manually adjusted to maintain visibility in a wide range of environments.

5.7 User Interaction

Singularity can receive input from and provide feedback to its operators through a variety of mechanisms. A wireless Xbox 360 controller is the primary means of manual control, and provides access to a variety of intuitive drive modes. This year we have implemented several assisted-driving modes that filter user input through autonomous driving routines in order to prevent inexperienced human operators from driving the robot into obstacles or people.

Singularity also features a completely redesigned wireless emergency stop system. The new e-stop uses a team-designed PCB housed inside an ergonomic 3D-printed case, and is powered by a rechargeable Xbox 360 wireless controller battery, which allows the team to source a common battery for both the e-stop and manual controller.

6 Software Design

In addition to building on the existing Robotics Simulation and Control Lab (RSCL) software platform, a brand new vision program called Wisconsin Vision Systems (WVS) was designed and implemented for Singularity.

For the 2013 IGVC, several new innovations are being introduced in our overall strategy. The most significant innovation is the vision system, which was designed to be able to detect lines accurately in a variety of lighting conditions. While last year the robot was able to navigate to GPS coordinates while avoiding obstacles, navigating at high speeds was an issue. Improvements were made to Singularity's driving code to allow for smoothly navigating a chosen route. Accelerometer data was also incorporated into the localization process, complementing data from the encoders. Reaction time of Singularity was also improved by bypassing mapping operations when new data did not indicate a need to change the existing map. Finally, this year Singularity is able to autonomously take full advantage of the unique drive modes that its omnidirectional design allows.

6.1 Localization

Accurate localization is an important part of being able to drive at high speeds without crashing. Four turn encoders, four drive encoders, and an inertial measurement unit (IMU) are used to maintain a local position. A compass and GPS are used to maintain the positions of global waypoints relative to robot's position in the local map.

New this year is the inclusion of the IMU to the localization operation. Challenges involved with this were calibrating the accelerometer, filtering its output, obtaining an updated state using the output, and integrating that state with the encoder data.

The IMU outputs raw digital ticks along six axes: three accelerometer axes and three gyroscope axes. During a fixed calibration period when Singularity boots up and is stationary, each axis is averaged to provide an initial offset. During normal operation, all axes go through individual low pass filters. These filters consist of a windowed average across multiple data points, which help to filter out jolting robot movements. They also include a minimum threshold for the change in orientation and change in velocity. This helps with random noise in the gyroscope and accelerometer that can cause drifting. The filtered gyroscope values are integrated with the current orientation to produce a new orientation. The filtered accelerometer values are then rotated using the derived orientation and integrated to determine the current position.

The position obtained from the IMU is integrated with the position obtained by the encoders before it is used for navigation. It was determined that the random error of the IMU is less than the random error in the encoders while the systemic error of the encoders is less than the IMU. For this reason, the IMU is trusted for very short-term movement and errors built up over time are corrected using the encoders.

6.2 Vision

To accomplish our goal of making Singularity omni-directional, the robot supports a 360° camera system. By pointing our camera upwards at a conical mirror, Singularity obtains a warped image that captures its surroundings. A drawing of this setup is shown in Figure 6.1.

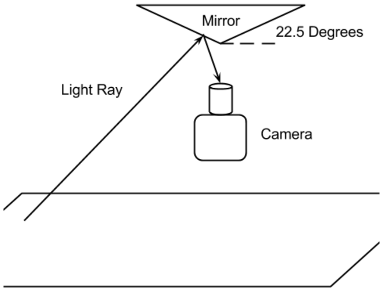


Figure 6.1 Omnidirectional camera setup.

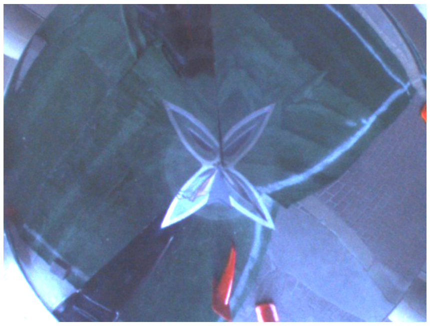


Figure 6.2 Example warped image taken while driving.

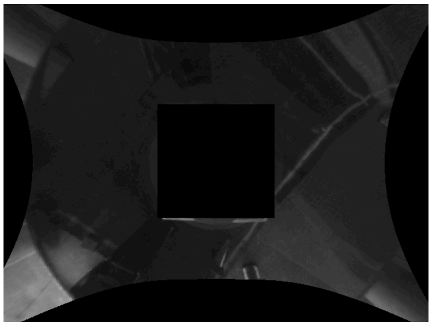


Figure 6.3 Unwarped and grayscaled version of Figure 6.2.

Before images gathered from the camera can be used to detect lines, they must first be unwarped. This year a new technique was used to unwarp images from the camera. As part of the initialization of the vision system, a lookup table is generated which maps the warped space to the unwarped space using the following equation.

$$R_{unwarped} = \left(\operatorname{atan}\left(R_{warped}\frac{Diameter_{mirror}}{Height_{mirror}}\right) - 45^{\circ}\right)/Zoom$$

This new equation produces a more accurate approximation than the previous method and the lookup table can be implemented efficiently using OpenCV. An example of the unwarping is shown in Figure 6.3.

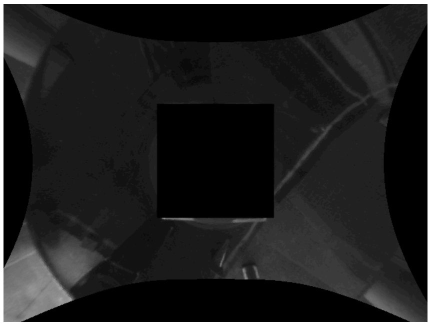


Figure 6.4 Sample Unwarped Image

Figure 6.5 Adaptive threshold applied to Figure 6.5

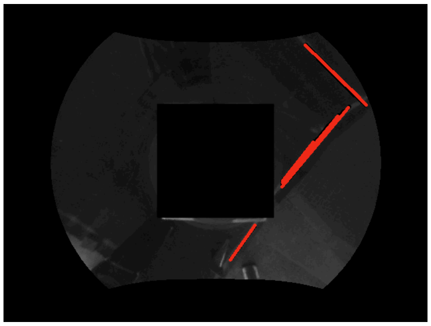


Figure 6.6 Lines detected by Hough Transform overlaid in red

Over the past year, several methods were tested to detect lines in an image. The primary motivation was the ability to detect lines robustly in varying lighting conditions, a major shortcoming in previous systems. Methods tried included: color segmentation, canny edge detection, morphological transformations, and parameter optimization. The current system utilizes an adaptive threshold on a grayscale image. This produces a black and white binary image by comparing the brightness of each pixel to those in the local neighborhood around that pixel. Based on various tests, the adaptive threshold performed just as well or better than any color-based technique and requires only a single parameter to be tuned. This parameter determines how faint of lines it will detect and is not dependent on the ambient light. An example of the adaptive threshold is shown in Figure 6.5.

To detect lines from the binary image produced by the adaptive threshold, the Probabilistic Hough Transform is used. Because the adaptive threshold produces an image invariant to ambient light, the parameters to the Hough Transform can stay constant across a variety of lighting conditions. This decreases manual tuning and increases the reliability of the algorithm. The results of the Hough transform can be seen in Figure 6.6.

6.3 Mapping

A map is generated using data from the four LRFs and the vision system to indicate passable areas on the course. First, obstacle feature points are expanded in polygons using a Minkowski sum[1]. The area represented by each polygon describes locations where Singularity cannot go without risking collision. A triangular navigation mesh is then constructed using the edges of the polygons as edges of triangles in the mesh. The navigation mesh provides an accurate and efficient way to traverse passable areas while avoiding non-passable regions.

To avoid unnecessary computation, this year additional logic was added to avoid the need to constantly recreate the navigation mesh. If an existing path does not cause a collision with any newly detected obstacles, a new path does not need to be generated. This collision check is done by comparing the distance between each point on the path with each newly discovered obstacle. If every point is far enough away from the obstacles, a new map is not created. Since a new map and path is now only created approximately every two seconds, the average time required to complete to one scan has decreased from 100 ms to 5 ms.

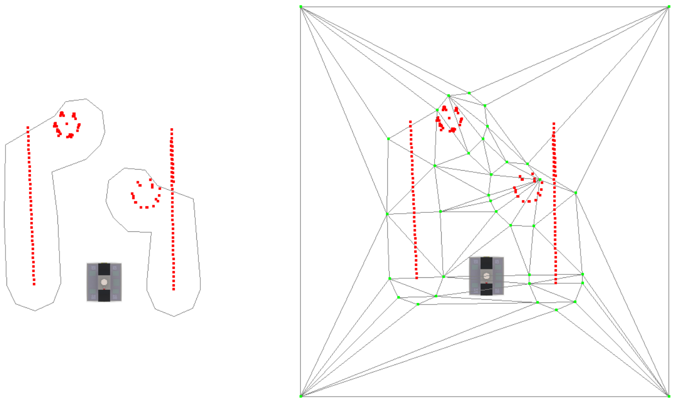


Figure 6.7 Birdseye view of an example course containing lines and barrels. Each red point is a data point seen by our sensors. On the left, non-passable regions are shown as polygons. On the right, the navigation mesh is shown.

6.4 Path Planning

Singularity tries to find the best path to its destination using the navigation map created. Due to turn tolerances, the best path is often not necessarily the shortest path. The "best" path is defined by several factors: length of path, distance from the path to obstacles, and smoothness of the path. Our algorithm defines the smoothness of the path as the angle between any three consecutive points on the path. The closer they are to a straight line, the smoother the path is and the easier it is for the robot to drive. Preference for paths that go in the same direction as the robot is currently traveling is also factored in to avoid sharp turns and going in circles.

---

[1] 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

Finding the best path to the goal is done in two parts, an A* search[2] and a path optimization. Each edge of the triangular mesh that is passable is divided into locations to search through. The distance between each of these points determines how long the A* algorithm takes to run and the resolution of the initial path. Based upon test results, a distance of one meter between A* nodes was chosen. The heuristic used for A* is designed to find the shortest path while avoiding going backwards. This helps to deal with the limited range of the laser range finder and camera by ensuring the robot will always be making progress through the course.

Since the path found by the A* algorithm is fairly coarse, a custom optimization algorithm is used to refine the path. As the optimizer only has to search in a neighborhood of the initial path (unlike A*, which searches the entire graph), a better path can be found in a reasonable amount of time. Last year the optimizer operated under two constraints: maximizing the distances from the nearest obstacles, and maximizing the smoothness of the path. This works fine for very enclosed areas where there is only a tight gap. However, for larger areas it produces a path much longer than necessary. To fix this issue, a heuristic was added this year which describes whether the chosen path is heading toward the goal or not, which has the effect of seeking to minimize the length of the path. Much better results were obtained in open areas with the new heuristic.

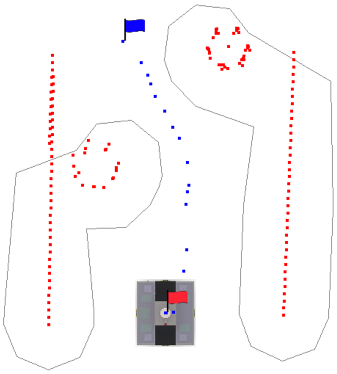


Figure 6.8 The blue points indicate the chosen path to reach the destination, which is shown as the blue flag. The red flag indicates the current goal.

Since the running time of the optimization algorithm depends of the number of points in the path, the calculations can become very computationally expensive. To limit this, the optimization is split into multiple passes. Each time a new scan is done, a new part of the path is optimized. By limiting the number of points optimized in a single pass to a fixed number, the computation time becomes fixed.

6.5 Driving

Getting Singularity to its destination requires smooth driving that closely follows the calculated path. This includes deciding which direction to move, the speed the robot should move, and how fast it should accelerate or decelerate at each point in time. These values are recomputed every time a new scan is done. This year, the method for computing these values was changed. To reach a desired velocity, separate acceleration and deceleration parameters are used. The deceleration is much higher than the acceleration to allow for fast stopping. The desired velocity is computed using the smoothness of the path being driven. If the path requires sharp turns the desired velocity is decreased. Otherwise, it is increased towards a maximum value. When Singularity encounters an obstacle to avoid, the path will naturally curve and result in a slower desired velocity.

---

[2] 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

This year another of Singularity's unique driving modes has been enabled for autonomous use, arcing mode. This causes the robot to tend to face 'forward' while it is traversing a path. Since the width of the robot is shorter than the length, always facing 'forward' means that small gaps are easier to go through.

6.6 JAUS

The JAUS library utilized in previous years was originally imported from the OpenJAUS project and updated to conform to newer versions of the JAUS specification. That library carried with it considerable complexity due to its original design goals, goals far beyond what was necessary for use with Singularity. Borrowing from lessons learned in previous years, the team pared down its JAUS implementation to a set of core functionality and began rebuilding the infrastructure needed to handle JAUS messages.

The new JAUS library, BadgerJAUS, provides a handful of utility functions to help with packing and unpacking JAUS data types for transmission over networks. It is both simplistic enough to allow for intuitive use and flexible enough to handle some of the more complex, dynamically-sized JAUS messages. Heavy emphasis was placed on code reuse and minimizing the number of exposed functions that needed to be reimplemented for each message. This significantly reduced the size and complexity of the JAUS library, making the development of additional messages and services simpler once the baseline library was completed.

To ensure that the new BadgerJAUS library correctly implemented the specification, a unit testing framework using the old, proven OpenJAUS library was implemented. This framework stepped through various exchanges such as discovery of services and requesting and relinquishing of control to sending drive commands. By using a known, working JAUS implementation, the new BadgerJAUS implementation could be verified to be functionally correct and suitable as a foundation for further extension in the future.

6.7 Simulation

A robot simulator, first built last year, aids in designing and testing software features for RSCL. Currently, the simulation packages allow simulation of a virtual robot's movement, LRF and line data, and localization sensors including the compass, GPS, encoders, and IMU. It can also add noise to each of these signals to evaluate algorithm robustness. Finally, it supports saving and replaying of real data sessions to enable more realistic and accessible test environments.

This year the user interface was cleaned up and a distributable package was created for users to be able to play with a virtual robot without having to download a much larger code base. This package can now be used for outreach.

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.

Table 7.1 Performance comparison

Metric	Design Goal	Recorded Value
Battery Life	3 hours	4 hours
GPS Arrival Accuracy	1 m	~1.5 m
Maximum Speed	10 mph	5.5 mph
Obstacle Detection Distance	10 m	10 m
Ramp Climbing Ability	15% gradient	15% gradient
Reaction Time (includes I/O)	150 ms	~50.0 ms

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, emergency stop, 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
	Slip Rings	4	$540	$540
Computer	Motherboard, Memory, SSD	1	$367	$367
	Wireless Router – Linksys WRT320N	1	$60	$60
Vehicle Control	TS-7800 SBC and Peripherals	1	$420	$420
	Motor Controllers – Team Designed	4	$600	$300
	Wireless Emergency Stop – Team Designed	1	$200	$200
	Warning Lights – Team Designed	1	$80	$80
	Wire and Interface Hardware	-	$120	$120
Sensors	SICK PLS101 Laser Range Finder	4	$12,000	$300
	Sentech TC33 USB Camera	1	$700	$700
	Garmin GPS 18x-5Hz	1	$160	$160
	OS4000-T Compass	1	$250	$250
	Encoders and Potentiometers	-	$300	$300
Power	ATX Power Supply – M4-ATX 250W	1	$100	$100
	Batteries – 75Ah 12V Deep Cycle Lead-Acid	2	$120	$120
Total			**$18,195**	**$6,195**

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 2013 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.