Singularity

Presented by the IEEE Robot Team
University of Wisconsin-Madison

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

THE UNIVERSITY of WISCONSIN MADISON

_____

Professor Michael Zinn
Department of Mechanical Engineering

Table of Contents

1 Introduction

The University of Wisconsin-Madison IEEE Robot Team is pleased to introduce Singularity to the 19th annual Intelligent Ground Vehicle Competition. Singularity was first conceived in 2009 shortly after IGVC to address a number of its predecessor's shortcomings, both at the competition and in other scenarios. After two years of work, the team is proud to submit a robot that is both innovative and practical. The team's objective for this project is to go beyond the challenge of the competition and design a versatile and adaptable platform that is useful for other applications as well.

The UW-Madison IEEE Robot Team is composed primarily of undergraduate students studying engineering and computer science. Each sub-team meets several times a week outside of class to work on their projects. All of our 27 members are volunteers who participate without receiving compensation or course credit.

2 Innovations

Singularity is the IEEE Robot Team's latest robot incarnation, featuring entirely new mechanical and embedded platforms. Additionally, the software system has been significantly updated to address shortcomings identified in previous years, and to take advantage of the new platforms' capabilities.

2.1 Mechanical Innovations

The key innovative features of Singularity's mechanical platform are the omnidirectional drivetrain and vision systems. Omnidirectional movement allows for more efficient driving and obstacle avoidance by adding several degrees of freedom that do not exist with conventional differentially driven drivetrain systems. Also, omnidirectional movement makes the robot easier to maneuver from a software standpoint. To support this new maneuverability, the mechanical system provided the robot with omnidirectional vision and laser range detection.

2.2 Electrical Innovations

Singularity's electrical system features many improvements over that of Paradroid, the team's previous robot. Strict safeguards are placed on commands sent to the motors to prevent high-level software from compromising the structural integrity of the robot. The motion control system of the robot is implemented using four team-designed circuit boards which allow for the independent control of Singularity's drive pods. One innovation not yet seen at the IGVC is the use of four laser range finders (LRFs) in tandem with an omnidirectional camera, which give more accurate 360° obstacle and lane detection than global mapping or vision-only approaches.

2.3 Software Innovations

This year the software team developed several software innovations that improve the performance and efficiency of Singularity. In development of the omnidirectional vision system, the team created the software necessary for converting camera images into 360° ground-plane images. Also, Singularity uses a novel, new lane detection algorithm tailored to IGVC competition requirements, providing significant performance improvements over out-of-the-box lane detection algorithms. Another notable software innovation is an improvement in the team's path

planning algorithm, which, using inverse kinematics, increases power efficiency and reduces wear and tear on the mechanical system.

3 Design Process

The development process for Singularity began in September of 2009. The original mechanical and electrical designs were completed in spring of 2010, but it was determined that the software could not be sufficiently tested on the robot before the 2010 IGVC, since the build process itself would take several months. The robot was built starting in the fall of 2010 and continued through the following spring; the electrical and software components were built in parallel. The team spent an estimated 10000 hours over a period of 18 months designing and building Singularity.

3.1 Team Structure

The UW-Madison IEEE Robot Team 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. All-team meetings are held monthly to

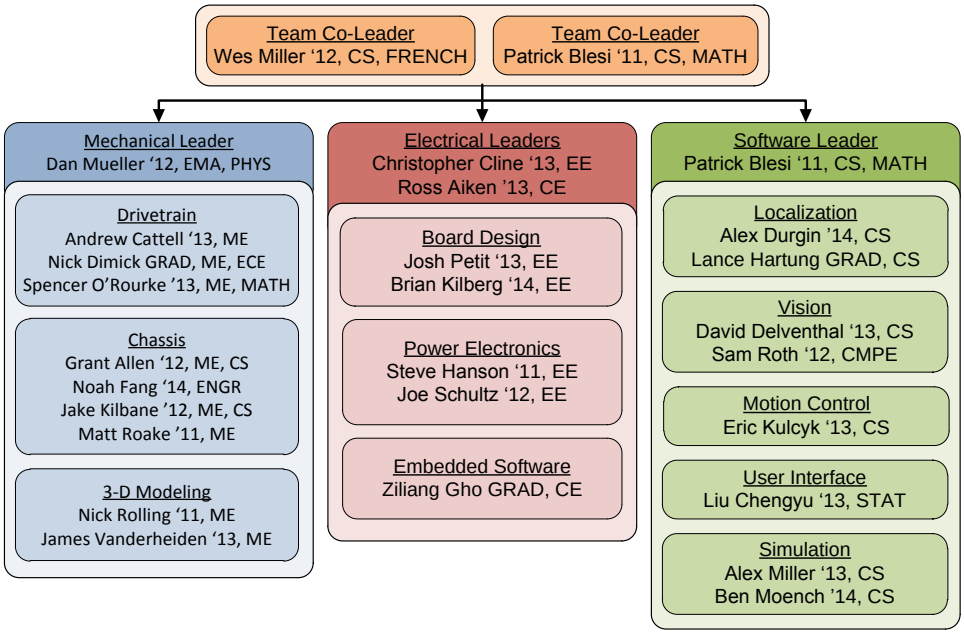

Figure 3.1 Team organization diagram.

facilitate communication between the sub-teams and to showcase the progression of the development process. Major design decisions are made by a consensus of team members.

3.2 Project Planning

The planning process began at the first meeting of the 2009-2010 academic year. The decision to implement an omnidirectional drivetrain was immediately favored for its myriad of benefits, and was followed by proposals for omnidirectional sensors and convenience features such as a simpler method of replacing batteries. Deadlines were shared between team leaders and at all-team meetings. The in-person communication of deadlines made

interdependence of different features easy to understand, and allowed the team to adjust its priorities based on which tasks would add the most value to the robot given the time remaining until competition.

3.3 Development

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. Conversely, the software team used agile methodologies to allow for easier adaptation to the changing scope of their projects. The electrical team, whose projects involved both hardware and software design, used a combination of both development processes. 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.

The mechanical team's development cycle consisted of computer-aided design, prototyping, production, and testing phases. SolidWorks, a computer-aided design program widely used in industry, was used to model each component in the vehicle. By using SolidWorks, many ideas could be visualized quickly without cost, and components could be tested for interference and proper interaction before being built. At times, experts on campus were contacted about how to best solve a specific design issue in the most efficient and effective way. After designs were completed and tested on a computer, prototypes were built for proof-of-concept testing. If the prototypes worked, then the designs were finalized and the parts were 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, computer-aided design printed circuit board layout tools.

The software team carried out much of its development using pair programming techniques. This reduced the amount of debugging needed and resulted in more legible code. Pairs worked on individual components and unit tests for the components. When unit tests passed, each pair moved on to testing their component in conjunction with other components. The software team also focused on producing working revisions of software whenever possible. The use of a modular software framework made this relatively easy, because nonfunctioning components could be kept in the root of the versioning repository without being included in a build.

4 Mechanical Design

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

system for the drive pods, mounting for embedded components, and the shell which protects the inside of the robot.

4.1 Drivetrain

The omnidirectional drivetrain has the capability to drive Singularity with more degrees of freedom than its predecessor. Each pod is turned by a 24 Volt 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 Volt, 450 Watt 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 will enable a high torque capacity for each wheel. The maximum speed of these drive motors far surpasses the 10mph speed limit for the competition; however the additional power will enable Singularity to drive over the rough terrain and uneven surfaces encountered at competition. The maximum speed of the robot is restricted by the motor controller boards.

Figure 4.1 One of four drivetrain pods attached to Singularity's frame.

The pneumatic wheel on each pod is 10" in diameter and is carefully pressurized to be soft enough to absorb impact yet hard enough to reduce the power needed to rotate the pod. Furthermore, the drivetrain allows for more efficient driving because the speed and direction of each wheel can be independently controlled. For every direction in which the robot can drive, there exists a speed and direction for each pod such that no drag is applied to any of the wheels.

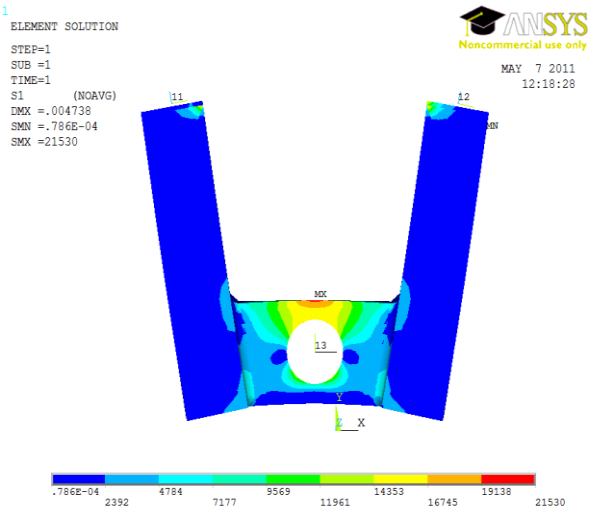

Figure 4.2 ANSYS structural analysis results for the base of the drivetrain pod, showing a maximum tensile stress of 21530 psi.

To reduce the amount of maintenance required throughout the vehicle's lifetime, each pod is supported on a carefully constructed bearing system. Shocks directed upwards through the pod are absorbed by the surrounding frame, consequently reducing stress on the turning mechanism. The structural integrity of the design for the drivetrain was verified through both hand calculations and actual testing. First, the design was analyzed to determine whether it could withstand hitting a bump at top speed. Upon professional recommendations of several professors (credit: Carl Martin, Robert Witt) this was modeled as applying the dead load of Singularity's own weight in 2g conditions. Finite element analysis performed on this

component using conservative estimates, as illustrated in Figure 4.2, showed the part met at worst 62% of yield, confirming the safety and reliability of the design. Additionally, after the first pod was constructed, it was impact tested to determine whether it would withstand the force of the entire robot dropped from a height of 6“.

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 is 39” tall when the camera mount is collapsed, allowing the robot to fit inside a minivan without disassembly. The camera mount adds an additional 16” of height to provide a large range of visibility for lane detection.

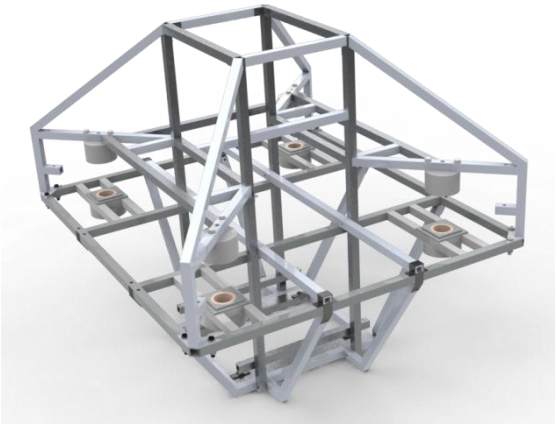

Figure 4.4 Singularity’s original hybrid aluminum-steel frame design.

The frame needed to be strong enough to withstand complex loading from the omnidirectional drivetrain, provide the stability necessary to reduce vibrations felt of the sensors and cameras, and be as lightweight as possible. Initially, the chassis was designed with a hybrid of high-strength steel and lightweight aluminum to reduce the overall weight of the robot without sacrificing structural integrity. However, the tight corners and complex geometry of the frame posed major challenges for the welded aluminum portion of the structure. For example, it was determined that the aluminum battery compartment design would not be able to withstand the force of 100-lbs of batteries being loaded and unloaded. In order to meet the desired structural integrity requirements, the majority of the frame was redesigned with steel square-tube. This had the additional benefit of allowing the frame to be welded using the Metal Inert Gas (MIG) process, which is faster than the Tungsten Inert Gas (TIG) process used to weld aluminum.

In order for the embedded and software teams to have enough time to test before competition, Singularity’s chassis was streamlined. The simplified geometry reduced the pressure on the welders and allowed for a rapid turnover of the robot to the team members in charge of configuring the electrical components. Additionally, the current chassis uses less material than the old design, leading to a cleaner look and leaving space for future modifications and improvements.

4.3 Serviceability

Singularity was designed to provide easy access to its internal components for maintenance and upgrades. The batteries and two of the laser range finders are attached to a pull-out tray. This allowed the size of the battery compartment to be reduced compared to its predecessor while speeding up the process of replacing the batteries. Similarly, all of Singularity’s circuit boards are attached to easily-accessible panels mounted underneath

Singularity's outer shell. The catadioptric assembly on the top of the robot can be removed and stored in the payload area, which allows Singularity to fit in the back of a mini-van for long-distance transportation.

4.4 Sensor Placement

Singularity implements an omnidirectional optical system for lane 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 surrounding several feet 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. However, 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 areas.

In addition to the omnidirectional vision, there is an LRF mounted on each side of Singularity. These serve to detect obstacles through 180° ranges, which were placed to minimize blind spots. Also, due to interference, it was necessary to position the LRFs in a way such that all the scans they would make are vertically offset from each other by 20 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 of the magnetic field.

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 deliver efficiency, functionality, and safety. In addition to handling all of the low-level sensor interfacing, the electrical system provides power to all system components.

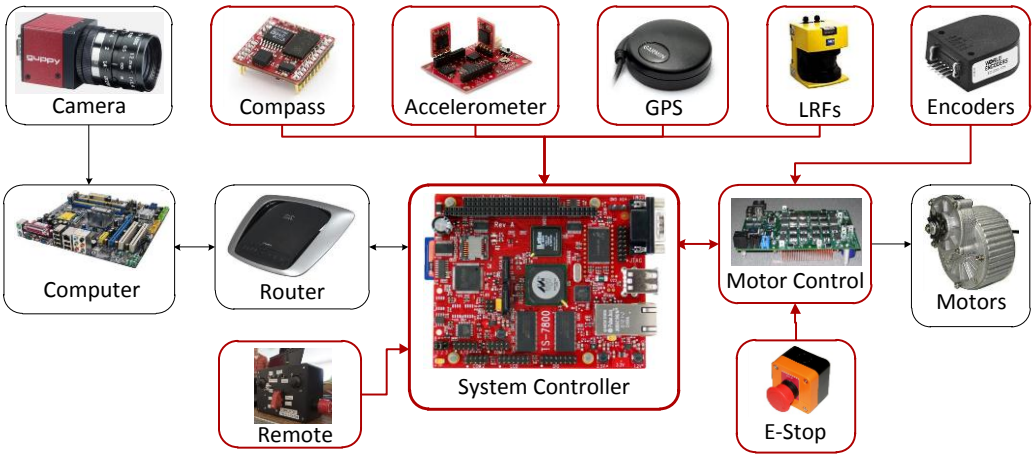

Figure 5.1 Embedded system diagram. Red-outlined components indicated the embedded system.

5.1 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. This system provides the simple interface between the main computer and the rest of the electrical system. It communicates with the main computer via Gigabit Ethernet connection, enabling simple, reliable high-speed data transfer.

5.2 Main Computer

Singularity uses a custom-built micro-ATX computer as its main computational platform. It provides substantially more processing power than an equivalently priced laptop and is tightly integrated with the rest of the robot, which reduces the risk of damage to the computer and peripherals during operation in tough environments. The computer is outfitted with a 2.5GHz Intel Core 2 Quad Processor, 4GB DDR2 RAM, and 4GB of solid-state permanent storage. A 23W passively cooled nVidia Quadro graphics card is also installed for parallel data processing (GPGPU). This allows Singularity to quickly analyze sensor data and react to changes in its environment.

5.3 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 Firewire 400 connection. It receives power via this connection as well. Singularity also has four SICK PLS101 laser range finders, providing a complete 360 degree 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, a second accelerometer to provide accurate rotational data, and quadrature and absolute drive pod encoders.

5.4 Motion Control

Singularity's drive system implements a brand new method of motion control. Four team-designed motor controllers drive eight motors (four steering, four drive) on Singularity completely independently. Each motor controller commands a single drive pod, and receives feedback from quadrature encoders, absolute encoders, and potentiometers mounted on the drive pods. This control scheme allows Singularity to accurately drive on uneven terrain. The motors can be given commands to simulate several common driving modes, such as Ackermann steering, differential drive, and synchro drive.

5.5 Remote Control

Singularity uses a Linksys WRT320N wireless router to enable remote connectivity between the internal systems and the outside world. Using this router, Singularity can communicate with any device capable of connecting to an 802.11g (Wi-Fi) network. The wireless system is used to provide shell access to the main and embedded systems. Additionally, this allows the robot to be controlled remotely via the team's custom-designed, JAUS-based user interface.

A team-designed, hand-held Operator Control Unit (OCU) allows for manual control of the robot. The OCU communicates with the robot using a pair of 900MHz ZigBee modules, which offer excellent reliability and performance. The system supports automatic channel hopping, power modulation, and 128-bit AES encryption of the wireless transmissions. Also, the modules have a maximum line-of-sight range of up to six miles, which is far greater than any reasonable usage of the robot.

5.6 Electrical Safety Features

The wireless emergency stop system is integrated directly into Singularity's OCU. The system uses a side channel from the ZigBee link to completely bypass the embedded software system when sending emergency stop commands. This dedicated data line is more reliable than sending an emergency stop packet from the remote, as all software on both sides of the ZigBee link is bypassed. The emergency stop is also triggered when the ZigBee link is broken. The emergency stop cuts power to the drive motor circuitry, bringing Singularity to a halt in less than one foot of travel.

Singularity also includes a warning light and an optional 110dB air horn to provide visual and auditory warnings during operation. The embedded team decided to expand upon the basic safety features required in the LED warning system: besides switching to a flashing pattern while in autonomous mode, the LED lights around Singularity will also repeat a cascading pattern to indicate the direction the robot is traveling All of the electrical systems on Singularity are protected by fuses in order to prevent the failure of one component from affecting other components. In addition, the power converters incorporate over- and under-voltage protection, as well as short-circuit and electrostatic discharge protection, making the power system robust under a wide variety of difficult conditions.

5.7 Power System

Table 5.1 Power System Requirements

		Normal		Maximum	
Device	Volts	Amps	Watts	Amps	Watts
TS-7800 Single Board Computer	5	0.4	2	0.8	4
Warning Lights	3.3	0.6	1.98	1.2	3.96
Misc Electronics	5	0.5	2.5	1	5
Linksys WRT54G Router	12	0.4	4.8	0.4	4.8
Garmin GPS 18x-5Hz	5	0.05	0.25	0.1	0.5
Guppy F-080 Firewire Camera	12	0.2	2.4	0.2	2.4
Main Processor DC-DC supply	24	3	72	4.5	108
Sick PLS101 LRF (4)	24	0.8	19.2	1	24
Turning Motors (4)	24	2.75	66	24	576
Scooter Motors (4)	24	20	480	85	2040
Total Watts			**651.13**		**2768.66**

Singularity's power system is designed to maximize vehicle run time and make software development and testing as easy as possible. Power is derived from two 12V deep-cycle lead acid marine batteries that form a 24V nominal battery pack with 75AH capacity. This battery system provides power for up to three hours of operation under normal conditions and up to ten hours in standby mode. The long battery life and integrated charging port allow

Singularity to run nearly continuously. In addition, depleted batteries can be replaced in minutes to maximize run time in the field.

Power conversion using team-built DC-DC converters provides 24V, 12V, 5V, and 3.3V power to the various systems on the robot at 85-95% efficiency. A separate LRF power supply allows all four LRFs to be powered from a single circuit board. Efficiency is approximately 93% when each LRF draws 1A. One distinctive feature added to this power supply is a combination of both a hardware push button reset and a software reset, which allows the LRF's to be power cycled manually or from software in the unfortunate event of a failure while in autonomous mode.

6 Software Design

Singularity's software system is an evolved version of the Robotics Simulation and Control Lab (RSCL). RSCL is a JAVA framework originally developed by the team in 2005. It was designed with the principles of simplicity, modularity, and robustness in mind. The team sought to continue this tradition because the design aspects closely aligned with the goals of our team: providing a training platform for inexperienced members that also meets IGVC performance requirements. The RSCL platform is easily accessible to undergraduates because it was originally designed by fellow undergraduates, and because it is written in JAVA, UW-Madison's primary instructional programming language. Finally, RSCL was chosen because it has proven to be a versatile and robust platform at past IGVC competitions.

A marked deviation in our objectives from last year is an emphasis on a more reactionary approach compared to last year's heavy mapping and path planning approach. This was done to reduce software complexity and increase robustness. Through careful deliberation, and through experience from last year's attempts, it was determined that the additional complexity brought on by a simultaneous localization and mapping (SLAM) system was not adequately justified by the requirements of the competition. It was deemed that all objectives could be completed with the less complex system.

6.1 Structure

Singularity's software architecture is comprised of 3 distinct layers: the sensor/effector daemon layer, the observable layer, and the observer layer. The sensor/effector daemon layer is comprised of servers which interface directly with the sensors/effectors and transmit received data or commands to or from a software client via TCP/IP sockets. Implementing the sensor interfaces in this way increases modularity and the scalability of the system for future projects. This year we incorporated an open source software package called daemontools which monitors all of our daemons and immediately restarts them in the event that they encounter a catastrophic failure. This has greatly improved the robustness of our robot, increasing testing hours and providing insurance should something fail in the middle of a live run.

Our second layer, the observable layer, consists of asynchronous clients that run as part of RSCL, each running in its own thread. They connect to the aforementioned servers and provide data asynchronously to upper-level layers using an event-based subscriber model. We chose this implementation to provide greater modularity by

freeing higher layers from the burden of implementing and being tied to specific server connections. The asynchronous event model ensures that higher layers always have access to the most recent data, ensuring the highest possible temporal accuracy of our models. These clients are implemented using a singleton programming paradigm to reduce the presence of redundant network traffic travelling to higher layers.

The third layer, the observer layer, consists of high-level data modules such as maps, goals, path planners, pose estimators, and operating modes. The high level of abstraction in these modules allows them to be easily ported to different robots with minimal modification.

6.2 Obstacle & Lane Detection

The main sensors used for obstacle and lane detection are the four laser range finders (LRFs) and camera. Raw data received from the LRFs and camera are synthesized into a set of objects (obstacles and lanes) and are overlaid onto a local object map. This object map is then processed into a set of drivable regions surrounding the robot. This approach remains robust in the face of faulty data because it throws out all map data after a single iteration of the process. Only a small set of hints are maintained in order to inform the next iteration of likely object locations.

Our camera is used primarily for Lane Detection, though it also serves as our primary flag and pothole detection sensor. In developing the vision system it was necessary to balance the field of view the camera provides with the need to maintain sufficient image resolution for our vision algorithms to run effectively. It was determined through a set of calculations and experimental results that a maximum field of view ranging from 9 to 20 feet in a single direction would produce the best results. Given the maximum speed of the robot and speed of our algorithms and sensors, it was determined that a maximum sensor range of nine feet would be sufficient to properly navigate the robot through any potential situation. The LRF ranges are limited to 12 feet from a maximum range of 150 feet for this same reason.

6.2.1 Vision Processing

Singularity uses a catadioptric vision system with a cone-shaped mirror configuration in order to provide a 360-degree camera field of view. A cone mirror was selected because it provides all-encompassing visibility, which allows Singularity to maneuver in its environment without exiting the lanes. Additionally, they create an opportunity to perform omnidirectional stereo vision, which the team is considering implementing in the future. First, images taken from the camera off of the cone mirror are processed to detect points indicating lanes. Then, these points are un-warped using an algorithm that maps points in the image-space onto the ground-plane. Experimental results showed

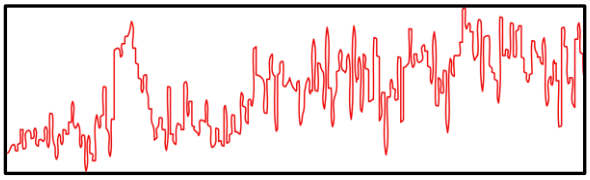

Figure 6.2 Intensity values from a row of the input image.

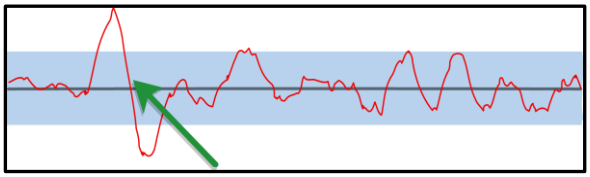

Figure 6.2 The same row after the edge filter has been applied. The green arrow indicates the detected line segment.

that running the algorithms in this order provided the best performance in terms of both efficiency and lane detection accuracy.

The lane detection algorithm first applies a noise-invariant, high-contrast filter to the input image, detecting paired high contrast edges. This filter works by calculating the difference between the averages of either side of each pixel. Then, corresponding adjacent high and low relative extrema, beyond a specific threshold, are selected as segments, rows of pixels bounded by high contrast edges. Segments are clustered into groups based on proximity; then, a shape analysis algorithm selects the groups that most closely approximate lines. Finally, curves are fitted to the selected groups of segments. These curves are then un-warped to ground-space coordinates using the unwarping scheme mentioned above and transmitted to the next layer.

This customized lane detection algorithm provides several benefits over more conventional lane detection algorithms. All filters applied to the images are separable, which reduces the runtime complexity of the processing from $O(n^2)$ to $O(n)$. In practice, our image processing algorithm can process a 1024x768 image in less than 30ms. This speed allows for more precise and temporally accurate data for implementing path planning algorithms. It is also tailored to deal with the high degree of noise detected in images of grass. Finally, the algorithm runs using image intensity values, as opposed to specific color channels. This makes the algorithm generalizable to a wide variety of applications such as object detection.

Once the detected curves reach the higher layer, they are further filtered based on their direction, size, position, and jaggedness. Additionally, they are compared with predictive transforms of the previous scan's curves in order to rule out false positives.

6.3 Autonomous Navigation

Singularity implements a hierarchical subsumption architecture for mapping and path planning. First a long term goal is calculated using the robot's previous path, lane direction, and waypoints. This long term goal, in conjunction with an obstacle map of nearest obstacles, is then used to find the direction closest to the long term goal that avoids obstacles. Finally, a set of commands is determined using inverse kinematics to get us as close to our midterm goal as possible.

The lane manager determines Singularity's long term goal by projecting out the center and direction of the detected lane. This is done in a robust manner by weighting how much vision-detected lines contribute to the calculation of the lane geometry based on how much they correlate with the robot's previous path (a good

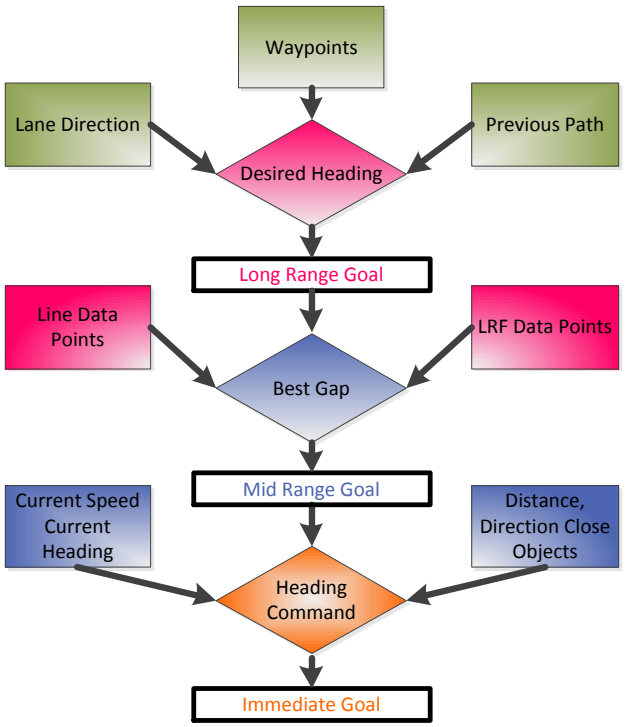

Figure 6.4 Path planning diagram

indicator of the actual lane). The long term goal alleviates the problem of following dashed lines because the goal is extrapolated from previous lanes and partial lane geometry. This is a marked improvement over previous years' algorithms which extrapolated lines as obstacles, creating problematic situations in the case of false positives.

The midterm goal is determined in accordance with input from the lane manager, replacing the lane manager's goal if it is not directly achievable. This is accomplished by gathering the closest objects in each direction into an array. This array is processed into a set of drivable gaps. The final midterm goal is chosen as the angle within the drivable gaps that is sufficiently distant from obstacles and most in the direction of the long term goal. Driving toward this midterm goal achieves obstacle avoidance (making complex obstacles such as center islands trivial to surpass) while still leading us in the direction of the chosen goal. One additional measure is taken: placing an obstacle on the map at the robot's previous position. This has the effect of propelling the robot forward in the case of switchbacks. If no valid gaps are detected, this obstacle is removed, and the midterm goal is recalculated, allowing the robot to escape dead ends and traps.

The short term goal provides a motor command that moves the robot as close to the midterm goal as possible. A table is generated mapping motor commands to predicted positions. The considered motor commands are within a certain nucleus of the current motor command, simultaneously limiting the robot's acceleration and the size of the calculated table. A fixed time interval, several times larger than the robot's reaction rate is used for position prediction. Thus the robots path is defined as the single motor command executed over the fixed time interval. This provides the robot with a steady, consistent path to the midterm goal, yet still allows the robot to quickly modify its path should its midterm goal change.

This path has several enticing properties. The implicit acceleration limit and preference for constant velocity reduces strain on the mechanical platform by avoiding rapid changes in speed. This path naturally reduces any tendency to turn in place, reducing both wear on the driving system and excessive power consumption. Finally, the consistent path coupled with a high reaction rate makes this algorithm ideal when the robot is operating at high speeds. Furthermore speeds are naturally throttled in the presence of close obstacles.

6.4 Navigation Strategy

The modularity of the software system makes the implementation of the navigation challenge code simple. The long term goal determined by the lane manager is simply replaced with the specific destination GPS waypoint. A simple greedy algorithm determines the waypoint traversal order. Though a Differential GPS system was outside the scope of the team's budget, an extended Kalman filter is used, combining GPS, encoder, and IMU data to provide us with accurate localization up to 10 cm.

This year the team developed a global obstacle map implemented using a range tree data structure. Obstacles are placed onto the robot's local map on every path planning cycle by making a query to the range tree. These obstacles are placed to tailor the robot's path and set up restricted driving areas. This will be used to force the robot to look at certain positions for a break in the fence separating the Valley and the Mesa.

6.5 Simulation

In order to facilitate testing of the software without using the physical robot, a simulation component was designed. It is composed of two parts, SimulateC (SimC) and SimulateD (SimD), as well as a GUI for each to simplify their usage. Modularity allows the simulator to be transparently inserted between the second and third layers of the system. SimC records raw data from the sensors on the robot, and SimD plays this data back into the software algorithms. This allows the team to test new algorithms more quickly and efficiently, checking their responses to previously encountered situations.

6.6 JAUS Integration

A single, light-weight framework was developed as the base of our JAUS compliance code. Our framework acts as a central kernel for our client-side and server-side JAUS implementation. This framework provides all message encoding, decoding, sending, and receiving functionality. All seven core services of the JAUS specification have been implemented, and six of the mobility services have been implemented as well. Building off of this framework, we have fully integrated our robot to be JAUS compliant, developed a graphical user interface that controls the robot entirely through JAUS messages, and developed a similar Android application that is capable of controlling the robot.

7 Performance

Singularity's rugged construction and powerful motors allow it to navigate over a variety of terrains at up to 10mph and climb steep slopes, while the omnidirectional drivetrain provides efficient maneuverability. The omnidirectional camera and LRFs detect obstacles several meters away which the powerful on-board computer can quickly react to using robust localization and mapping algorithms. Two large deep cycle lead-acid batteries provide ample run time for testing and should last all day under intermittent use.

Table 7.1 Performance characteristics

Performance Parameter	Prediction	Result
Top Speed	10mph	-
Ramp Climbing	30°	-
Reaction Time	200ms	85ms
Battery Life	3 hours	-
Obstacle Detection Distance	14 feet	12 feet
GPS Waypoint Accuracy	1.0m	1.0m

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 work. 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, power supplies, operator control unit, and motor controllers. Most of the software is written entirely by team members. In many cases, code originates from various open source projects and is updated or improved upon.

Table 8.1 Estimated parts cost for Singularity

System	Item	Qty	Cost	Our Cost
Mechanical	Structural Components	-	$783	$783
	Drive Components (Sprockets, Chain, etc.)	-	$252	$252
	Bearings	-	$238	$238
	Hardware	-	$111	$111
	Non-Structural Components (PVC, Shaft Collars, etc.)	-	$91	$91
	Motors	8	$703	$703
Computer	Main Board - Foxconn G33M-S Micro-ATX	1	$95	$0
	Processor - Intel Q9300 Quad Core	1	$280	$0
	Graphics Card - Quadro NVS 290	1	$120	$0
	Solid State Memory - 4GB	1	$65	$0
	Memory - 4GB DDR2 800	1	$61	$0
Vehicle Control	TS-7800 Single Board Computer	1	$270	$270
	Interface Board – Team Designed	1	$150	$150
	Motor Controllers – Team Designed	4	$600	$300
	Wireless Router – Linksys WRT320N	1	$60	$60
	Operator Control Unit – Team Designed	1	$300	$0
	Warning Lights – Team Designed	1	$80	$80
	Wire and Interface Hardware	-	$120	$120
Sensors	SICK PLS101 Laser Range Finder	4	$12,000	$300
	Garmin GPS 18x-5Hz	1	$160	$160
	OS4000-T Compass	1	$250	$250
	Accelerometer	1	$80	$0
	Drive Encoders	4	$200	$0
	Turn Encoders and Potentiometers	4	$100	$100
Power	Main Power Supply	1	$90	$90
	LRF Supply	1	$80	$80
	ATX Power Supply – M4-ATX 250W	1	$100	$0
	Batteries – 75Ah 12V Deep Cycle Lead-Acid	2	$120	$0
Total			**$17,559**	**$4,138**

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 2011 Intelligent Ground Vehicle Competition, and to highlight the strengths of the IEEE Robot Team. Singularity's modularity, versatility and efficiency should prove to be an ideal platform for autonomous vehicle research and development.