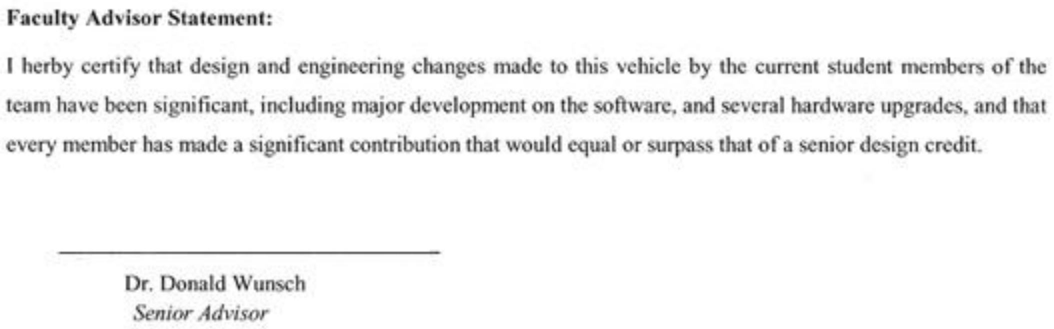
MISSOURI S&T University of Science & Technology ROBOTICS COMPETITION TEAM

"Helping Robots Help Themselves"

The Missouri University of Science and Technology Robotics Competition Team presents:

S&T ENTERPRISE

For entry in the 2013 Intelligent Ground Vehicle Competition



Faculty Advisor Statement:

I herby certify that design and engineering changes made to this vehicle by the current student members of the team have been significant, including major development on the software, and several hardware upgrades, and that every member has made a significant contribution that would equal or surpass that of a senior design credit.

Dr. Donald Wunsch
Senior Advisor

Contents

1. Introduction

The Missouri University of Science and Technology (Missouri S&T) Robotics Competition Team is proud to enter *S&T Enterprise* into the 21st Intelligent Ground Vehicle Competition. S&T Enterprise is a very durable platform, utilizing a two-wheel drive and a two castor system. Over the past year, S&T Enterprise has become an innovative platform for testing new code, hardware, and mechanical components. The following report is an analysis of S&T Enterprise's design, a description of innovations applied to the robot, and an overview of the S&T Robotics Competition Team structure.

1.1 Team Structure

The Missouri S&T Robotics Competition Team operates through the S&T Student Design and Experiential Learning Center (SDELC). The team consists of roughly 30 undergraduates from a variety of disciplines. A full member list can be found in Appendix A. The team is run by an executive board consisting of a president, vice president, secretary, treasurer, and public relations manager. A new executive board is elected each spring and officially takes office immediately after the IGVC. The executive board appoints three division leaders to oversee the mechanical, electrical, and computing divisions of the team. The use of divisions has allowed the team to break down the wide field of robotics into more specific disciplines for members wanting to focus their interests. Figure 1.1.1 shows the overall team structure.

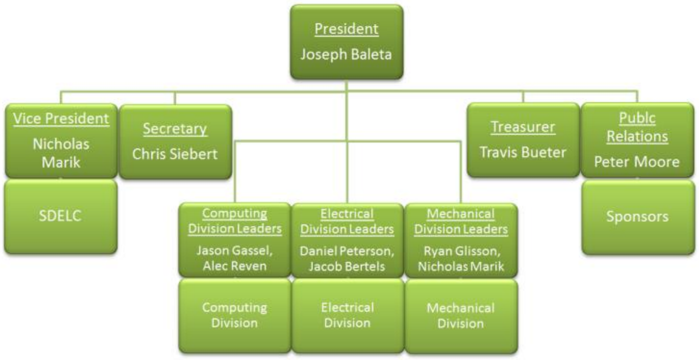


Figure 1.1.1: Robotics Competition Team Heirarchy

2. Design Process

2.1 Goals and Planning

The decision to build a new robot was made at the end of the previous spring semester. A new project was needed to attract new members and teach them about the design and manufacturing processes. As a member the SDELC, the team was required to submit a detailed proposal for the 2012-2013 project cycle. By August 2012, the budget and building schedule for S&T Enterprise were finalized. The team had four main goals for the robot:

1. Rewrite current code to be more robust.
2. Implement a more reliable power system.
3. Design a more stable and durable platform.
4. Test and ensure that all systems work in sync with one another.

2.2 Execution

S&T Enterprise was developed using the seven-step process outlined in Figure 2.2.1. After deciding to build the new robot, the requirements to achieve it were defined and a design was made. After fabricating the design and integrating the various systems, a rigorous test cycle began. Concepts were not minimally tested and then assumed to work entirely, but put through the entire test process again which involves trial runs in a competition setting. A practice field was created with painted lines and barrels to test S&T Enterprise in the same way that Jomegatron was tested.

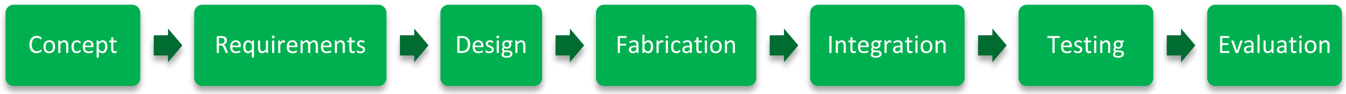


Figure 2.2.1: Design process for S&T Enterprise

With Jomegatron's design and construction, goals were never fully defined at the beginning of the project. There was a general understanding of what needed to be done to make the robot compete, but there was no list of detailed, measurable criteria driving its design. The robot's design was changed multiple times throughout the build process as overlooked problems became apparent. It was clear that this needed to change if S&T Enterprise was to become a reality. By themselves, the four main goals listed earlier are vague. So each goal was further defined, creating a set of requirements for the new robot. A set of clear, descript requirements made the design and build processes go much faster. Less time was spent trying to fix overlooked problems or change the design to fit a new function.

2.3 Innovations

S&T Enterprise includes several innovations, some of which were experiments the team took a risk in trying. Listed below are the greatest improvements to the robot, which are described in more detail throughout the rest of the report.

- Dampening suspension for rough terrain
- A waterproof composite shell and composite wheel wells
- CNC machined, fully enclosed gearboxes
- A new 24 volt electrical system
- A newly implemented navigation algorithm, A*.

3. Mechanical Design

Several mechanical lessons were learned from Jomegatron. The older robot's strengths were its ruggedness, easy accessibility to inside components, computer workstation for quick coding, and ability to fit through doorways. However, Jomegatron suffered from design flaws such as the skid-steer drive system it originally had, ineffective waterproofing, cumbersome Lexan shell panels, and awkward turning. S&T Enterprise's mechanical design was based on criteria aimed at solving the mechanical flaws of Jomegatron while meeting IGVC requirements. The following goals determined the robot's physical structure:

- Create a frame that meets all IGVC size requirements while being able to fit through doors
- Allow the robot to pivot about two center drive wheels
- Place the center of gravity between the drive wheels to maximize drive wheel traction
- Install a suspension on the front and back caster wheels to ensure constant wheel contact with the ground and decrease camera shake
- Improve waterproofing and accessibility over Jomegatron with a composite shell
- Develop a better transmission to gear down motors
- Make heavier components (batteries) easy to remove and replace
- Have a computer workstation on the back of the robot, complete with monitor, keyboard, and mouse.
- Have a designated compartment for the competition payload

3.1 Frame

A space frame design similar to Jomegatron's was chosen for S&T Enterprise to maximize accessibility and space for components. Square aluminum tubing was chosen for light weight and flat surfaces for easy installment of brackets, hooks, hinges, and pistons. The overall size of the frame was dictated by IGVC size requirements (three feet long by two feet wide). The width of the robot had to be as wide as possible to increase the distance between the drive wheels for better pivoting while still small enough to fit through a 36" wide doorway. A final width of 34" was chosen. The frame was divided into a top and bottom half connected by two pistons and a piano hinge, allowing it to open up for placement or removal of large parts inside.

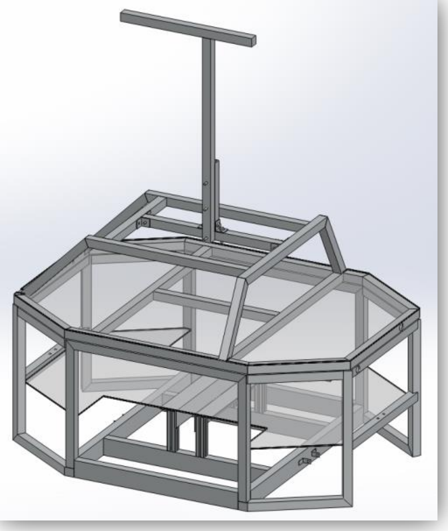


Figure 3.1.1: Frame modeled in SolidWorks 2012

3.2 Suspension

S&T Enterprise rolls on a set of two center wheels stabilized by swivel casters at the front and back ends. This configuration allows the robot to pivot about its center. A swingarm suspension was added to the stabilizing casters for two reasons:

1. With stationary casters, if the height of either caster was even slightly off, the robot would have a tendency to rock back and forth on the center wheels.
2. Jomegatron's camera suffered from severe shaking over rough terrain which affected the robot's performance. The suspension helps solve part of the problem, allowing S&T Enterprise to handle hills and bumps better than its predecessor.

RC car shocks were chosen for their small size, excellent damping characteristics, and ability to handle the smaller loads taken by the casters. With the majority of the weight centered along the drive wheels, each caster was expected to take no more than 40 pounds of load at any given time, although they could handle more. The swingarm design was much cheaper and easier to implement than linear bearings sliding along guide rails. Since the travel of the shocks is relatively small, the swingarm's rotation is small and the caster's path can be approximated as linear.

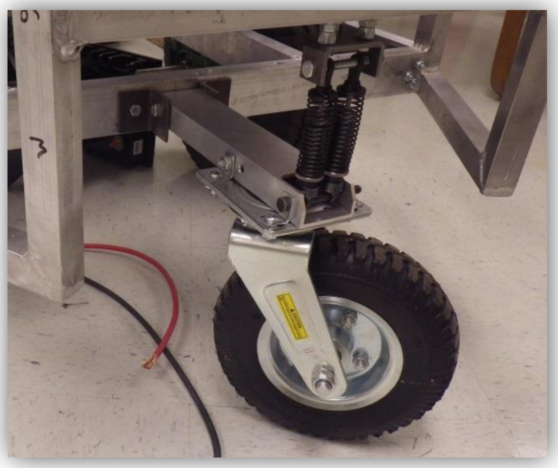


Figure 3.2.1: Caster swingarm suspension

3.3 Composites

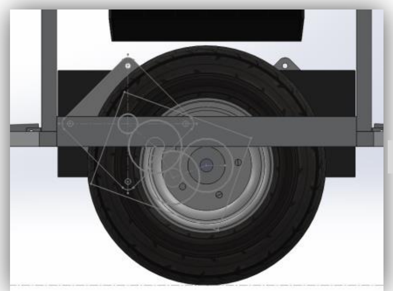
The use of composite materials was first tested on Jomegatron. The fiberglass wheel wells on the robot at the 2012 IGVC fit the contour of the wheel, proved to be waterproof, and looked much nicer than previous Lexan wheel wells. S&T Enterprise has the same type of drive wheels that Jomegatron had, so the same mold was used for composite wheel wells on the new robot.

The Lexan shell on Jomegatron suffered from several flaws. Panels could be easily knocked off, there were several panels that had to be accounted for, and the shell was never truly waterproof. It was clear that a new design would be needed to protect electrical equipment from the environment. The new shell needed to be better at blocking moisture, have fewer pieces, be quickly removable, look more professional and streamlined, and have plenty of room for sponsor logos. For such a large waterproof shell with smooth contours, composite material was an obvious choice. Fiberglass was chosen for its moisture resistance, low cost, and relative sturdiness. Pieces of machining foam were glued together into one block large enough for a half-shell mold. The block was run through a CNC mill to cut out the mold contour. After a coat of primer, light sanding, and coat of releasing agent, the mold was ready to be used. Two layups on the mold can create the symmetrical shell.

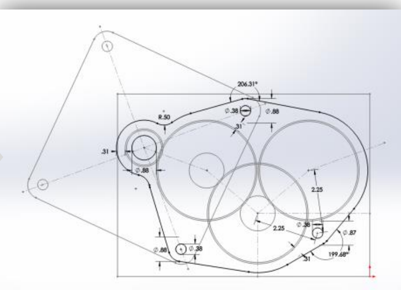
3.4 Gearboxes

The motors donated to the team for S&T Enterprise were originally meant for air compressor systems. A transmission was needed to gear down the high speed, low torque motors. The gearboxes used in the S&T Enterprise were designed with three goals in mind:

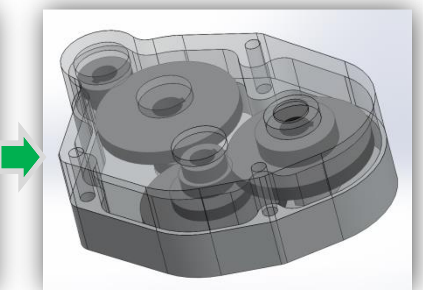
1. Gear down the motors by approximately 20:1.
2. Be as small as possible.
3. Be easily removable for maintenance.



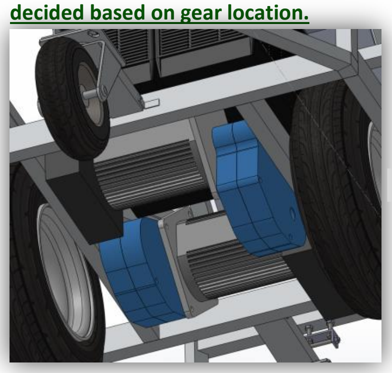
1. The gears are sketched onto the robot model and motor placement is decided based on gear location.



2. The sketch is imported into a new file to model the housing. Important parts such as the motor plate are included.

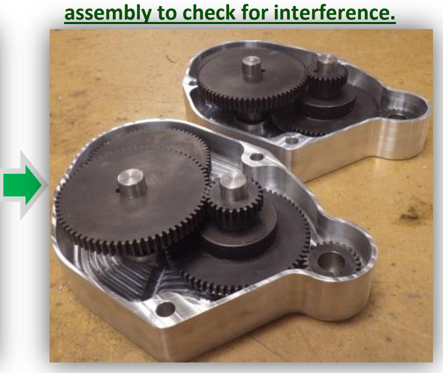


3. The housing is assembled. The gears are modeled and inserted into the new assembly to check for interference.



4. The gearbox assemblies are inserted into the robot model for a final interference check.

5. The gear housing is CNC machined and shafts are lathed. The gears are bored and keyed.



6. The gearboxes are assembled and tested before being installed on the robot.

Each gearbox contains three pairs of meshed gears. Each pair has a 2.8:1 ratio, creating a final gearbox ratio of approximately 22:1. The housing is split into two CNC machined halves that bolt to the motor plate. The gearboxes can be secured to the motor and easily opened for maintenance. Since the housing was not completely airtight, the mechanical division chose heavy lithium grease over a full oil bath to lubricate the gears.

4. Electrical Design

4.1 Power System

S&T Enterprise sports a 24 Volt electrical system. Two large lead acid batteries are used to supply 2.52 kilowatts of power. The batteries are arranged in series to increase the voltage. In the past, the team has had a higher voltage system to support more powerful motors; however, the motors in this system provide enough torque on the 24 Volts. Higher voltage systems proved to be difficult to find components for. Lowering the voltage to 24 Volts makes finding and replacing parts easier and cheaper.

A master switch allows the team to easily turn the robot's power on and off as needed. Power is distributed by a standard automotive fuse block. This style of block was chosen because fuses/breakers are commonly available in case a replacement is required. This fuse block supplies power to every electrical part of the robot along with the availability of expansion for adding future components.

S&T Enterprise carries a desktop computer for processing vision software and controlling the motors. While the computer is custom built, off the shelf parts were used. This means that a standard computer power supply was used. To supply power to the computer, and any other 120VAC devices, the robot carries a 900W power inverter. Rather than deal with having to order and maintain a custom, low voltage power supply, the team decided that a power inverter and standard computer power supply would be better. This way, if any failures occur in the computer power system, new components can be replaced very quickly.

4.2 Motors

Enterprise drives using two brushless DC (BLDC) motors, which can be powered by the 24 Volt power system. The motors were donated by Amtec Ltd. Each can produce 3 ft-lb of torque and spin up to 2300 RPM. With a gearbox, Enterprise should be able to hit a top speed of 15 MPH. The team does not plan to run the robot this fast, but this means that the motors should never be required to operate near their limits, prolonging their life. The motors include an on board controller designed specifically for these motors. This way, the team does not have to worry about mismatching a motor and controller.

These motors were originally designed to run air conditioner compressors in Greyhound buses, and as such, were designed to run at a constant speed in one direction. The team contact at Amtec, Rod Hower, happened to be the engineer who designed the on-board controller. With his help, two wires were run from the motor controller. A speed line now allows the motor's speed to be varied continuously from zero to full speed. A second line was also pinned out for direction reversal. Pulling this line low spins the motor in one direction, and pulling the line to power supply voltage will spin the motor in the opposite direction.

4.3 Emergency Stop System

The Emergency Stop (E-stop) system on S&T Enterprise is similar to the E-stop systems used on past Missouri S&T robots. The system is designed to run off of the 24 volt supply power so no power converters are required. Two Double Pole Double Throw (DPDT) relays are used to connect the motor controllers to the main batteries. If the relays loose power (robot is turned off or an E-stop signal is sent), the relays switch the motor power input rails into a bank of resistors instead. By doing this, the team ensures that the robot will not continue moving under power, but also that the robot will come to a quick, smooth stop even if traveling down hill.

The system is triple redundant and the robot may be stopped by remote control, software, or one of two switches mounted on top of the robot. The system is designed in push-to-break configuration. In other words, for the robot to continue moving, power must be sent to the motors. The remote control is not only a competition requirement, but also a good idea. If for some reason the robot were to get away, no one would have to chase the robot down. Instead, a simple button press will cause the robot to stop. Since the E-stop board is tied into the computer, the board can report and estop to the running program, and cause the code to stop sending motor control commands. This way, when the estop is reset, the robot will not try to contintute to move, but will instead wait for further instructions. This also allows a remote user to stop the robot's movement without needed the remote control. Of course, the physical switches on top of the robot may be used as well. Escecially when in close quarters (i.e. Public Relations events) the team needs to be able to stop the robot very quickly in case someone walks in front of it while it is moving. The buttons are also the only gaurenteed method of stopping the robot. These switches are physcally tied into the system. A remote computer shutdown and the remote control are not.

5. Software Strategy

S&T Enterprise's software was programmed in C++ and designed around ROS (Robot Operating System). ROS provides a dynamic and robust transport layer for the robot. The system allows code modules to be linked at runtime, making it easy to edit or replace a single module without the user being required to comprehend the program as a whole. The software stack is designed to map the environment and navigate to GPS (Global Positioning System) waypoints using the input from a single monocular camera. The software provides a GUI (Graphical User Interface) to display debugging data and allows users to provide input via the GUI or an Xbox wireless controller. All software is available under the Open Source GPL v3 license.

5.1 Vision

S&T Enterprise is equipped with a 640x480 resolution wide angle camera. The first step of the vision pipeline is to identify obstacles. The team has developed two primary methods for finding obstacles which are described below. Both methods attempt to identify obstacles based on their color and output an image marking all of the obstacles within each frame.

- ***Per Pixel Based***

The per-pixel based method of image segmentation identifies obstacles based on their color characteristics. The user specifies the colors of various obstacles in the frame as well as the color of the grass. The software creates normal distribution curves based on the input color chromaticity and hue components. When images are published from the camera driver every pixel in the image is given a probability of being an obstacle based on where it lies on the distribution curves. The module publishes a grayscale image defining each pixel by the probability that it is an obstacle.

- *Gradient Based*

The gradient based method of image segmentation attempts to first segment the image into regions of continuous color and then uses the statistics of all the pixels in a region to determine obstacles. To do this, the module creates runs of pixels in the X and Y directions that have a consistent change in gradient. The module looks at the second derivative of the image to determine the start and stop of runs. The runs are then linked together into regions defining areas with similar gradients. The statistics of all the pixels in these regions are then compared to the statistics of the training obstacles to determine obstacles within the image. The module publishes a binary image defining the pixels that make up all obstacles within the image. The modules may be launched separately or may be used together with their outputs combined. The blue and red flags of the competition are handled by creating virtual walls to the right of the red flags and left of the blue flags after performing segmentation. Once an image has been found with all of the obstacles marked, a homographic transform is applied to the image. The homographic transform attempts to create a bird's eye view of the area around the robot, correlating obstacles on the ground plain to their positions in the world. Ray-casting is then performed on the transformed image to give an array containing the distance to the closest obstacle along each angle. Both the homographic image and ray-cast are published. The various stages of the image pipeline may be seen in Figure 5.1.1.

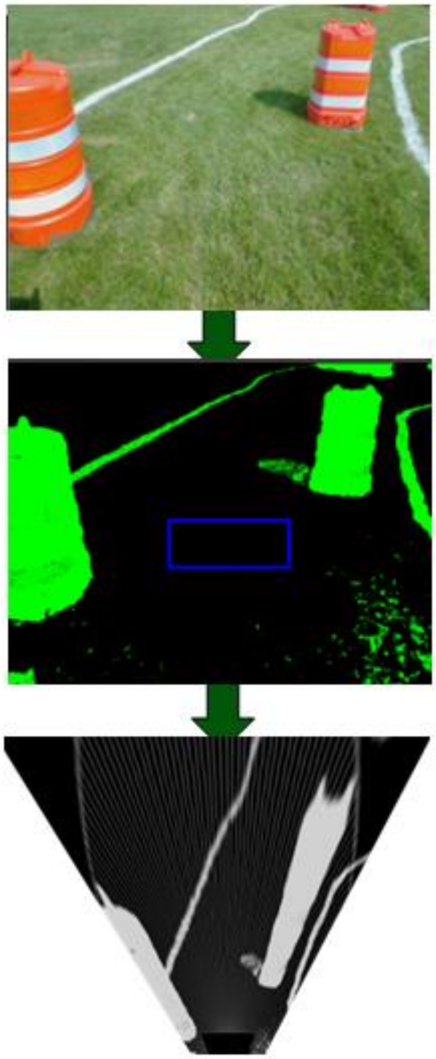


Figure 5.1.1: Image Pipeline

5.2 Position

The position module is in charge of maintaining an accurate account of the robot's position in the world. The module subscribes to the position information being published by the GPS/INS unit and the wheel odometer. The software combines all position information using a Kalman filter to maintain the most accurate position. The module is also in charge of maintaining a list of GPS waypoints. The waypoints may be loaded from a file or may be input by the user via the team's GUI or using the JAUS (Joint Architecture for Unmanned Systems) protocol. The module decides the robot's current target based on priority. If two waypoints are given the same priority, such as those in no-man's land, the program will choose the closest. The user may set time limits on each priority to be sure the robot has enough time to finish the course.

5.3 Model

The model module attempts to create an accurate map of the world. The team's current method uses the gmapping stack which may be found in the ROS repository. The gmapping stack uses a SLAM (Simulations Location and Mapping) algorithm to map the environment. The module subscribes to the ray-cast output by the vision pipeline and the combined position. The software places local obstacle information onto the global map by using the input position and tracking features from frame to frame. The software uses the tracking features to create a more accurate position and aid in future mapping. The module outputs the corrected position as well as a local and global map of obstacles.

5.4 Navigation

The navigation module is responsible for deciding the movement of the robot based on the obstacle map and the current target. The team has two methods of determining the robot's movement which are described below. The methods output desired forward and rotational velocities for the platform.

- *ROS Navigation stack*

S&T Enterprise's software was designed to be compliant with the ROS navigation stack. The navigation stack looks at the local obstacle map around the robot and determines the path needed to avoid close obstacles. The software then looks at the global map and attempts to find a path that will lead to the next waypoint given by the position node or by a user.

- *A* Algorithm*

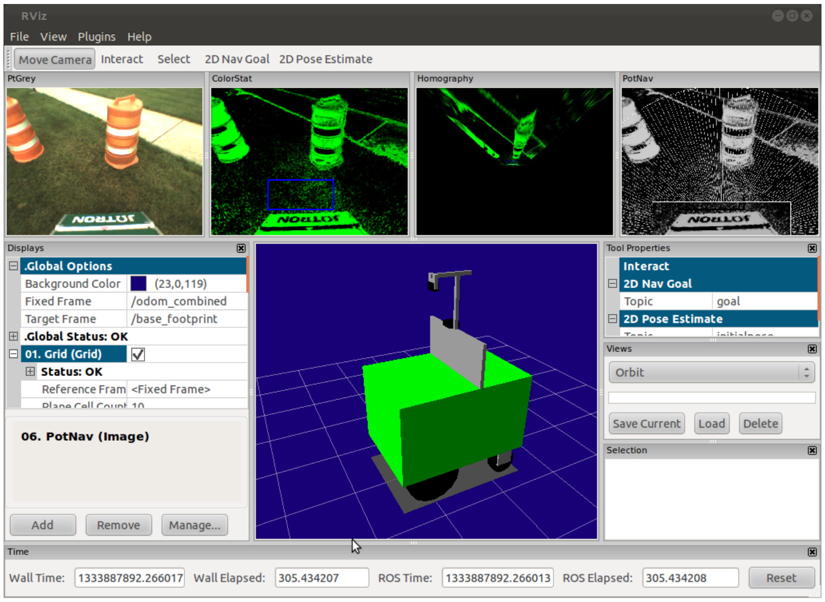
The potential field model used on Jomegatron caused several problems at the last IGVC. It became apparent that either the potential field code would have to be drastically changed or completely replaced for S&T Enterprise. The latter was done in the hopes that the new A* system would provide the robot with more direction when dealing with short-term destinations.

5.5 JAUS

The JAUS module was designed by the team to convert JAUS messages into ROS messages. The module was designed to be as general as possible and can easily be used on other robots. The software supports all of the JAUS capabilities and allows the user to pull information from the software and input controls.

5.6 Control

The control node has the final control over what velocity commands are sent to the hardware interface module. The node has several modes of operations that decide the behavior of the robot. The module starts in standby mode and waits for an Xbox controller to be connected or for the JAUS node to take control. Once the user connects, the user will have the ability to place the robot into either user controlled mode or autonomous mode. In autonomous mode the node will pass the velocity commands published by the navigation software through to the motor controller node, giving the software control over the robot. In user controlled mode the software will interface with the Xbox controller or JAUS module and compute velocity commands based on user inputs. The software also launches a ROS tool named RViz. RViz is a visualizer that allows the user to view combined information about the robot's inputs in a three-dimensional virtual environment. The output of the RViz display may be seen in Figure 6.6.1. The display is customizable at run time so users may view any debugging information that is being published. Users are also able to subscribe to this data over a network allowing for remote operation. The node interfaces with a text-to-speech library to provide feedback about the current state of the robot.



⊟ .Global Options
Background Color (23,0,119)
Fixed Frame /odom_combined
Target Frame /base_footprint
⊞ .Global Status: OK
⊟ 01. Grid (Grid) ☑
⊞ Status: OK
Reference Fram <Fixed Frame>
Plane Cell Count 10

06. PotNav (Image)

Interact
⊟ 2D Nav Goal
Topic goal
⊟ 2D Pose Estimate
Topic initialpose

Orbit

Time
Wall Time: 1333887892.266017 Wall Elapsed: 305.434207 ROS Time: 1333887892.266013 ROS Elapsed: 305.434208

Figure 6.6.1: RViz Robot Dashboard

5.7 Hardware Interface

The hardware interface node converts the desired robot velocity output into wheel velocities. The software uses the computed wheel velocities to create serial commands which are then sent to the motor controllers. The module reads back the encoder information and publishes the wheel odometer. The software also interfaces with the e-stop board, giving the state of the robot and control over the safety light.

6. System Integration Plan

S&T Enterprise's control software was developed by several student members of the team, all working on different modules and levels. Early in the design process, emphasis was placed on higher-level module functional descriptions and interface specifications. After the interfaces were designed and the desired functionality achieved, sub-teams were free to start coding the internals of each module.

Testing was performed at both the module level using test drivers and at the system level using lab and hallway operational tests. Once acceptable behaviors were reliably demonstrated in the lab environment, outdoor operational tests were conducted on a local field designed to replicate the IGVC course.

As several team members participated in past IGVC events, the team was able to re-create all of the challenging features typically found at the IGVC, including solid and dashed white painted lines, various densities of grass / dirt, shadows, sun glare, ramps, cones of various types and colors, snow fencing, plank saw-horses, switchbacks, center islands, dead ends, traps, potholes, and sand pits.

During the spring semester, the team scheduled numerous outdoor tests, each of which focused on a particular set of issues. After each test, the testing sub-team reviewed the results, took notes, and made plans to address any deficient performance observed.

7. Performance Expectations

The robustness of S&T Enterprise's algorithms have been proven multiple times in simulations. The new A* navigation code is much more reliable than the potential field model. The team's predictions along with the design's demonstrated values can be found in Table 7.0.1.

Table 7.0.1: Performance Comparison Table

Characteristic	**Design Goal**	**Demonstrated in Field Test**
Max Speed	5 MPH (2.24 M/s)	4 MPH (1.79 m/s)
Ramp Climbing Ability	15 degrees	25 degrees
Reaction Time	4 hertz	5 hertz
Battery Life	2 hours	3 hours
Distance of Detection	• Web Cameras : ○ 4 M forward ○ 3 M side – looking ○ 5 M Diagonal	• Web Cameras : ○ 4.5 M ○ 3.2 M ○ 5.52 M
Accuracy at Waypoints	2 M	1.75 M

7.1 Complex Obstacles

The control software detects and handles the following special situations. Specific detection / handling methods are described below:

- *Switchbacks*

When a switchback situation is encountered, S&T Enterprise will seek the path of least resistance. When no such path is obvious within 190 degrees of the front view, S&T Enterprise will turn 180 degrees and examine the rear environment for a potential exit path. The limited obstacle model memory will discourage Jomegatron from repeatedly taking the same path.

- *Dead Ends*

S&T Enterprise retains a short-range memory of objects visited in the past few dozen cycles. If S&T Enterprise encounters a dead end, it will rotate 180 degrees (as in the switchback case above) to look for a more promising path.

- *Traps*

To negotiate traps, S&T Enterprise employs a method similar to that used to detect and navigate out of dead ends.

- *Potholes*

S&T Enterprise will avoid all potholes provided they are a sufficiently different color than the grass.

- *Dashed Lane Lines*

S&T Enterprise's particle-based vision produces notable artifacts when it encounters a partial lane line, allowing it to detect it as a dash and not an opening. These artifacts are detected and subsequently avoided to prevent the robot from entering the opening.

8. Safety

8.1 Emergency Stop

With all of the power Jomegatron can supply, there needs to be a way to stop the robot in case the operator loses control. S&T Enterprise is equipped with a triple redundant emergency stop (E-stop) system. There are two buttons located at hand height on the main frame of the robot. If either button is pressed, the motor controllers shut off and relays switch the motor power lines into a bank of resistors to bring the robot to a quick and easy stop. In addition to the buttons, an AVR micro-controller takes commands from the computer and from a remote control to stop the robot remotely. The computer must reset a count on the micro-controller every second to keep the robot moving. This prevents the robot from running off in the event of a computer glitch or total computer failure. Finally, all software E-stops may be bypassed so that the robot can be driven directly in case of E-stop board failure. The buttons are hardwired, however, and cannot be bypassed so that the robot can be stopped for sure in this way.

Power tracking and E-Stop systems can be handled by an independent AVR microcontroller, ensuring that these critical systems continue to function in the event of a computer failure or when the robot is in a low-power state. Power tracking via current and voltage sensors enables the robot to provide an accurate estimate of its remaining battery life and give warnings when battery levels are dangerously low.

Even though the Wii gaming console's Wiimote has proven effective at controlling the robot for the past two years, we have decided to utilize an Xbox controller for programming ease. The team can link the Xbox controller, drive manually to testing areas, then place the Xbox controller in standby mode for autonomous testing. S&T Enterprise's hardware limits its speed to just less than five miles per hour, and the fuses installed on the motors ensure that they receive no more than forty amps. The robot is also programmed to stop upon the loss of a Xbox controller or Wireless E-Stop connection or in the event of a crashed program. If the motors module has not received a request in the last 3 seconds, it will safely stop and turn off the motors. This prevents a single module crash from causing a runaway situation.

9. Cost in Dollars/Hours

S&T Enterprise's design and build process began during the fall semester of 2012 and was finished the spring semester of 2013. S&T Enterprise's total cost is show below in Table 1.

S&T Enterprise's construction and programming required a large amount of man hours. Each team member spent an average of four hours a week working for the (design, construction, funding, etc.). Given 22 active members and 32 weeks of work over the academic year, S&T Enterprise cost approximately 2800 man hours.

Table 9.0.1: S&T Enterprise expenses

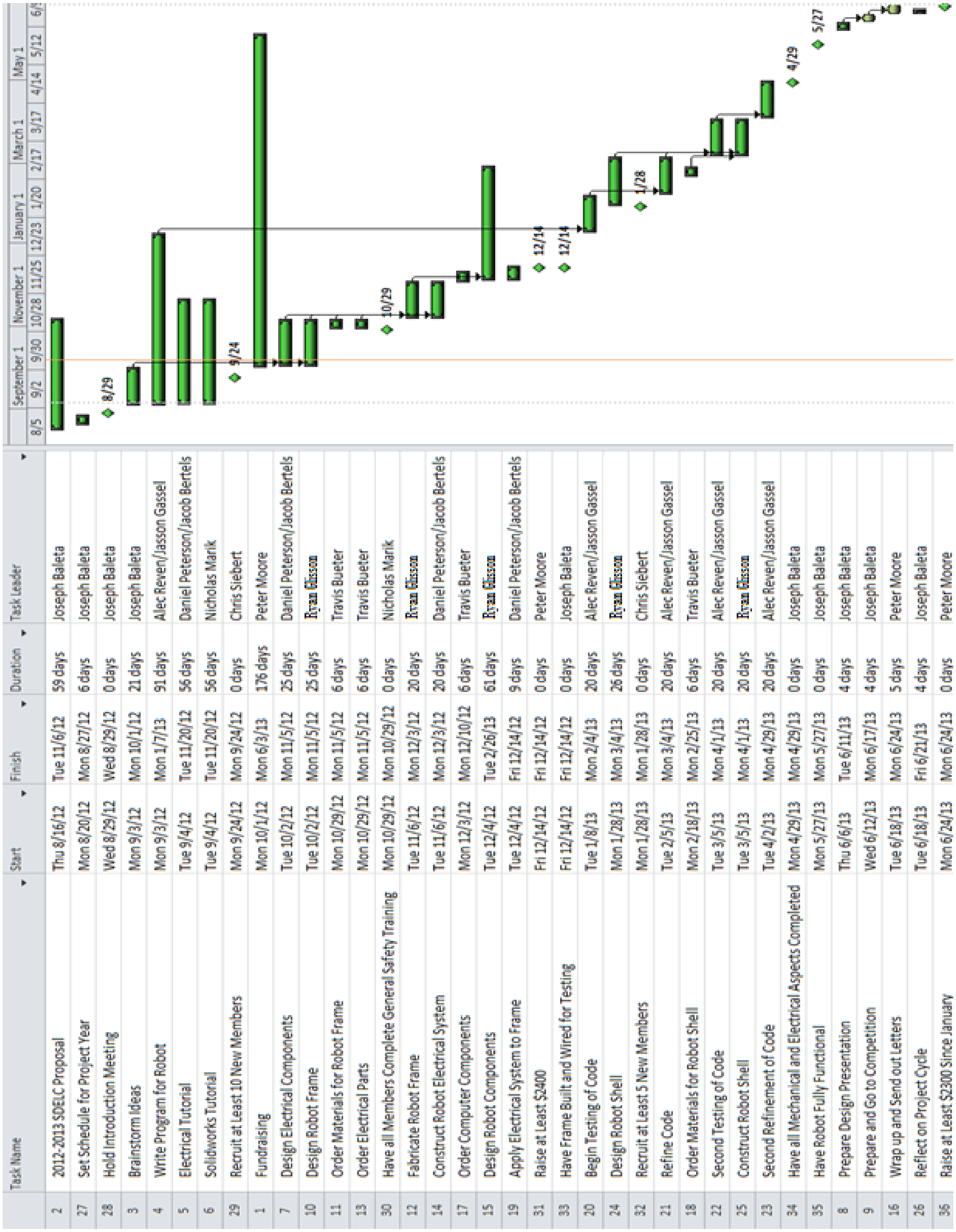
Comparison of retail cost versus what the team paid

		Actual Cost	Retail Cost
Mechanical Division			
	Frame	$ 300	$ 300
	Motors	$ -	$ 1,600
	Gear Boxes		
	Shell		
	Wheels	$ 100	$ 100
	Bearings	$ 33	$ 33
	Swivel Casters	$ 49	$ 49
	Shocks	$ 27	$ 27
	Round Bar	$ 36	$ 36
	Bearings	$ 33	$ 33
	Lexan	$ 140	$ 140
	Misc. Hardware	$ 132	$ 132
Electrical Divison			
	Motor Controllers		
	Batteries	$ 280	$ 280
	Charger	$ 305	$ 305
	Power Supply		
	Motherboard	$ 69	$ 69
	Caution Light	$ 69	$ 69
	Computer		
	Misc. Hardware	$ 160	$ 160
Computer Divison			
	GPS	$ 68	$ 68
	Xbox Controller	$ 45	$ -
	Point Grey camera	$ 300	$ 300
	Camera Wide Lens	$ 260	$ 260
Totals		$ 2,406	$ 3,961

Appendix A: 2012-2013 Member Roster

Name	Position	Name	Division	Name	Division
Dr. Donald Wunsch	Main Advisor	Jacob Bertels	Electrical Lead	Sheldon Harper	Mechanical
Dr. Douglas Bristow	Technical Advisor	Ryan Glisson	Mechanical Lead	Valentine Mbah	Mechanical
Dr. Zhaozheng Yin	Technical Advisor	Samuel Pester	Mechanical	Rory Hayes	Computing
Joseph O. Baleta	President	Michael Lester	Computing	Emily Hernandez	Electrical
Nicholas Marik	Vice President	Ashley Painter	Mechanical	Jessica Greathouse	Electrical
Travis Bueter	Treasurer	Jason Jarquio	Mechanical	Josh Davis	Electrical
Christopher Siebert	Secretary	Dustin Pieper	Electrical	Matthew Anderson	Computing
Peter Moore	Public Relations	William Roussin	Electrical	Alan Gallegol	Computing
Alec Reven	Computing Lead	Ryan Glosemeyer	Electrical	Dzung Tran	Mechanical
Jason Gassel	Computing Lead	Zach Kreuer	Mechanical		
Daniel Peterson	Electrical Lead	Alex White	Mechanical		

Appendix B: 2012-2013 Schedule



	Task Name	Start	Finish	Duration	Task Leader
2	2012-2013 SDELC Proposal	Thu 8/16/12	Tue 11/6/12	59 days	Joseph Baleta
27	Set Schedule for Project Year	Mon 8/20/12	Mon 8/27/12	6 days	Joseph Baleta
28	Hold Introduction Meeting	Wed 8/29/12	Wed 8/29/12	0 days	Joseph Baleta
3	Brainstorm Ideas	Mon 9/3/12	Mon 10/1/12	21 days	Joseph Baleta
4	Write Program for Robot	Mon 9/3/12	Mon 1/7/13	91 days	Alec Reven/Jasson Gassel
5	Electrical Tutorial	Tue 9/4/12	Tue 11/20/12	56 days	Daniel Peterson/Jacob Bertels
6	Solidworks Tutorial	Tue 9/4/12	Tue 11/20/12	56 days	Nicholas Marik
29	Recruit at Least 10 New Members	Mon 9/24/12	Mon 9/24/12	0 days	Chris Siebert
1	Fundraising	Mon 10/1/12	Mon 6/3/13	176 days	Peter Moore
7	Design Electrical Components	Tue 10/2/12	Mon 11/5/12	25 days	Daniel Peterson/Jacob Bertels
10	Design Robot Frame	Tue 10/2/12	Mon 11/5/12	25 days	Ryan Glisson
11	Order Materials for Robot Frame	Mon 10/29/12	Mon 11/5/12	6 days	Travis Bueter
13	Order Electrical Parts	Mon 10/29/12	Mon 11/5/12	6 days	Travis Bueter
30	Have all Members Complete General Safety Training	Mon 10/29/12	Mon 10/29/12	0 days	Nicholas Marik
12	Fabricate Robot Frame	Tue 11/6/12	Mon 12/3/12	20 days	Ryan Glisson
14	Construct Robot Electrical System	Tue 11/6/12	Mon 12/3/12	20 days	Daniel Peterson/Jacob Bertels
17	Order Computer Components	Mon 12/3/12	Mon 12/10/12	6 days	Travis Bueter
15	Design Robot Components	Tue 12/4/12	Tue 2/26/13	61 days	Ryan Glisson
19	Apply Electrical System to Frame	Tue 12/4/12	Fri 12/14/12	9 days	Daniel Peterson/Jacob Bertels
31	Raise at Least $2400	Fri 12/14/12	Fri 12/14/12	0 days	Peter Moore
33	Have Frame Built and Wired for Testing	Fri 12/14/12	Fri 12/14/12	0 days	Joseph Baleta
20	Begin Testing of Code	Tue 1/8/13	Mon 2/4/13	20 days	Alec Reven/Jasson Gassel
24	Design Robot Shell	Mon 1/28/13	Mon 3/4/13	26 days	Ryan Glisson
32	Recruit at Least 5 New Members	Mon 1/28/13	Mon 1/28/13	0 days	Chris Siebert
21	Refine Code	Tue 2/5/13	Mon 3/4/13	20 days	Alec Reven/Jasson Gassel
18	Order Materials for Robot Shell	Mon 2/18/13	Mon 2/25/13	6 days	Travis Bueter
22	Second Testing of Code	Tue 3/5/13	Mon 4/1/13	20 days	Alec Reven/Jasson Gassel
25	Construct Robot Shell	Tue 3/5/13	Mon 4/1/13	20 days	Ryan Glisson
23	Second Refinement of Code	Tue 4/2/13	Mon 4/29/13	20 days	Alec Reven/Jasson Gassel
34	Have all Mechanical and Electrical Aspects Completed	Mon 4/29/13	Mon 4/29/13	0 days	Joseph Baleta
35	Have Robot Fully Functional	Mon 5/27/13	Mon 5/27/13	0 days	Joseph Baleta
8	Prepare Design Presentation	Thu 6/6/13	Tue 6/11/13	4 days	Joseph Baleta
9	Prepare and Go to Competition	Wed 6/12/13	Mon 6/17/13	4 days	Joseph Baleta
16	Wrap up and Send out Letters	Tue 6/18/13	Mon 6/24/13	5 days	Peter Moore
26	Reflect on Project Cycle	Tue 6/18/13	Fri 6/21/13	4 days	Joseph Baleta
36	Raise at Least $2300 Since January	Mon 6/24/13	Mon 6/24/13	0 days	Peter Moore