S&T Enterprise

Missouri S&T

Peter Moore

Dr. Donald Wunsch II <dwunsch@mst.edu>

INTRODUCTION

The Missouri University of Science and Technology (Missouri S&T) Robotics Competition Team is proud to present *S&T Enterprise* as an entry in the 2014 Intelligent Ground Vehicle Competition. The robot will be making its second appearance at the IGVC as the tenth consecutive entry from the Missouri S&T Robotics Competition Team. S&T Enterprise is a very durable platform, utilizing two-wheel drive and a two castor system. As a veteran of IGVCs, the robot serves as a teaching tool for new members. The study of its strengths and weaknesses has proved invaluable as the team works to build better projects and train new generations of students in the field of robotics. The following report is an analysis of S&T Enterprise's design, a description of the significant changes made since the 2013 IGVC, and an overview of the S&T Robotics Competition Team structure.

Team Structure

The Missouri S&T Robotics Competition Team operates through the S&T Student Design and Experiential Learning Center (SDELC), which provides resources and support to all fourteen of the school's student-run design teams. The team is comprised of 29 Missouri S&T 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

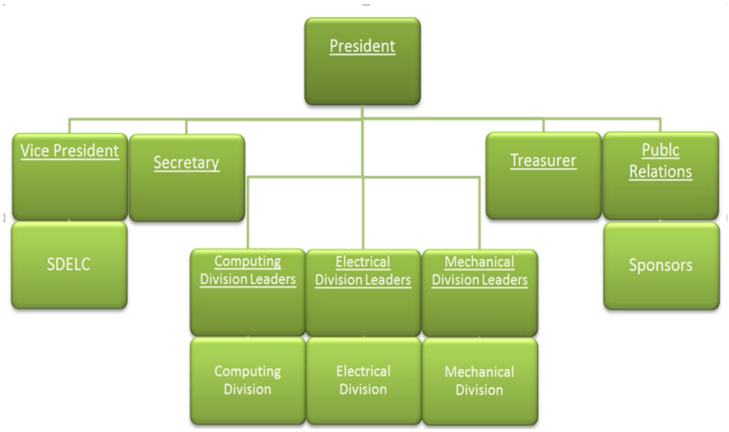


Figure 1 Team Hierarchy

robotics into more specific disciplines for members wanting to focus their interests. Figure 1 shows the overall team structure.

DESIGN PROCESS

Goals and Planning

S&T Enterprise's performance at the 2013 IGVC was not as strong as the previous year. This was due to software and electrical problems. S&T Enterprise's motor controllers did not function as planned, and last minute adjustments had to be made at competition. There were also problems with the emergency stop system that had to be fixed before the robot could qualify. There were additional issues with the software, we were unable to properly interface and address our motor controller drivers. The competition highlighted the robot's weaknesses, making clear what would need to be fixed in the 2013-2014 project cycle. The team had two main goals for S&T Enterprise at the beginning of the 2013 fall semester:

1. Create a more reliable electrical system
2. Rewrite navigation code to be more reliable in competition situations.

Execution

The upgrade process is cyclical. Solutions are 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. Permission was granted to paint white lines in regularly mowed fields at a public park. Construction cones and barrels were then placed within the lines and S&T Enterprise was sent through the obstacle course several times.

VEHICLE UPRGRADES

By building on the lessons of the previous year the team has created a robot fine-tuned to the needs of this competition. The problems of the 2013 model have been replaced with new capabilities and increased reliability. Underneath the fiberglass shell of the 2014 model there have been major innovations in both hardware and software.

Electrical Upgrades

Two major upgrades were implemented for the electrical system in order to bring S&T Enterprise to a fully-functioning and safe status. Aside from the normal maintenance of replacing a few bad cables and connectors, new motor controllers and a wireless emergency stop transmitter and receiver were constructed.

An Arduino Uno microcontroller was selected to communicate between the onboard computer and the two motors. The computer and microcontroller setup a serial connection using an usb cable to control the both the speed and direction of each motor. The standard arduino programming language is used to send a PWM signal to the speed of each motor to speed up or slow down the robot. A small switching circuit consisting of a few transistors allow the small microcontroller to swap between sending the direction cables 0 V or 24 V from the battery. The motor controller is placed into a small project box to prevent damage to the circuit and controller. This controller is also in charge of switching the onboard status light. This small function was easily added to replace the use of another microcontroller. The status light is powered with 12 V and swaps between a constant on position and a pulsing light.

The wireless emergency stop button (E-Stop) is a required component that was redesigned to create a reliable and safe prevention system. Two XBee 1mW modules were chosen to perform the transmission of the emergency stop command, with a range of 100m. The transmitter is powered by two AA batteries and sends kill signal whenever the XBee loses power. A status light turns on whenever the E-Stop has been issued. If the batteries were to ever lose power then the robot would automatically be stopped. The receiver holds a 3.3V relay closed whenever receiving the signal and opens the relay whenever the signal is lost. A bypass switch is used when testing the robot without requiring the remote E-Stop transmitter to be on.

Software Upgrades

The potential field navigation system was completely replaced with a new system known as A*. The A* algorithm is widely used for path finding and graph traversal, the process of plotting an efficient path between points called “nodes.” Unlike the potential field system which was designed to mimic gravity in a way, A* uses a best-first search and finds a least-cost path from a given initial node (the robot) to a goal node (the next GPS waypoint). As A* traverses the graph, it follows a path of lowest expected total distance, keeping a sorted priority queue of alternate path segments along the way. However, A* can’t see the entire map at once, which means the map and route must be continuously modified as position changes.

MECHANICAL DESIGN

Frame and Shell

S&T Enterprise’s frame is made out of square inch aluminum tubing and is 5 feet long. Aluminum was chosen for light weight while square tubing allows for easy installation of brackets and fixing components. The frame is divided into an upper and lower half. The upper half is attached to the bottom using a hinge along the starboard side with two hydraulic springs to assist opening. This allows easy access to the bottom portion of the robot for placement and removal of large components. S&T Enterprise’s frame size was based around IGVC regulations. At the beginning of the design process, it was understood that the frame had to meet all

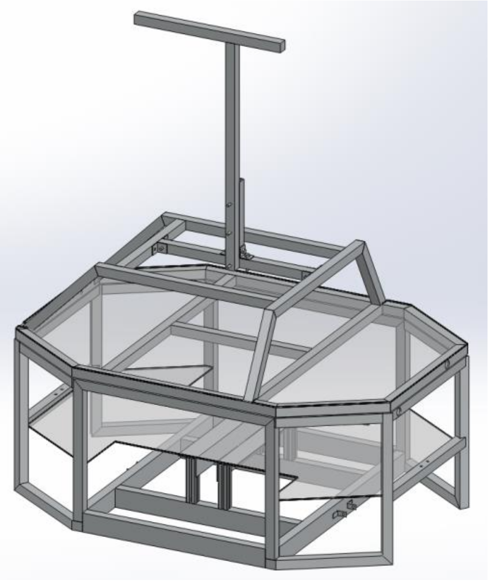


Figure 2 Solidworks Frame Design

regulations while still being able to travel through doorways and openings at least 30” wide. The efficient placement of components inside allows the frame to have significant cargo capacity providing more than enough room for the standard package. The inner components are protected by an external shell and wheel wells made from rigid epoxy. This provides thorough protection from rain or other adverse weather and course conditions.

Drive Train

S&T Enterprise features two independently driven wheels for zero turning capability. For either wheel a large AC motor is coupled to a gearbox machined and manufactured in house. The output of this is then connected to the wheel shaft by a couple.

Figure 3 Robot with Outer Shell Removed

Camera Mount

The camera mount reaches up to a height of 5 feet in order to give the camera a top view perspective. The camera mount was designed to be very adjustable. The mast features is attached rigidly to the frame via a simple four bar mechanism, and with the locking pin removed can be safely folded flat for transport. The camera mount was designed to satisfy the needs of the previous year’s software vision setup and to accommodate any changes that could be needed in the future.

ELECTRICAL DESIGN

S&T Enterprise features an electrical system with improvements based on several previous years of experience. This system is designed to serve the team for years to come without a foreseeable need of service or significant changes that often were required in the past.

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.

In order to prevent the batteries from ever running out of power, a charger was mounted onto the robot to allow the system to charge whenever near a standard wall outlet. A 24VDC 25A automatic smart charger was chosen to quickly and safely charge the lead acid batteries. The integration of the charger also allows development on the onboard computer without the requirements of the batteries to be in place.

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.

E-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 off. Two red, onboard, emergency stop buttons were placed onto the robot to prevent a situation where the robot is cannot be stopped.

The system is triple redundant, and the robot may be stopped using a special remote control, by software, or by 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 was to get away, no one has to chase the robot down. Instead, a simple button press will cause the robot to stop. Since the estop 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 estop is reset, the robot will not try to continue 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. Especially 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 while moving. The buttons are also the only guaranteed method of stopping the robot. These switches are physically tied into the system. A remote computer shutdown and the remote control are not.

Sensors

Vision Sensors. S&T Enterprise uses a single Point Grey Firefly MV camera. This camera operates at a resolution of 0.3 megapixels and is able to supply VGA (640x480) images at 30FPS over a standard IEEE 1394a “Firewire” connection. Standardization of 1394 cameras provides access to all internal setting registers for camera configuration. A removable lens with a 2.2mm focal length provides a 130-degree field of view.

Position Sensors. The robot utilizes a Microbotics MIDG-II INS/GPS as its primary position / pose sensor. This device includes a WAAS compliant GPS, a 3-axis accelerometer, a 3-axis rate gyro, and a 3-axis magnetometer. The device is capable of integrating positional information through an on-board Kalman filter and sending revised position / pose information via a serial interface at 50 hertz.

Additional positional information can be derived from the motor controllers, which maintain a running position based on the wheel encoders. This derived positional information is relatively accurate at short time scales, but tends to drift over time due to wheel slippage. The combination of the GPS for long-term absolute accuracy, accelerometers for intermediate accuracy, and wheel encoders for short-term accuracy is used to determine the most probable position at any point in time.

Computing

S&T Enterprise carries a full desktop computer to handle all of the vision, mapping, and navigation tasks the team requires for competition. The computer has an AMD 8 core processor and a graphics card (GPU) to aid in vision processing. The

graphics card gives the team the ability to perform parallel processing on images, providing a large speed increase. The computer uses a solid state drive to store data allowing memory to be safely accessed while the robot is in motion.
To aid in software development and to give the operator more feedback about the state of the robot, S&T Enterprise carries a “software workstation” complete with monitor, keyboard, and mouse. This allows the programming team to make changes to the software directly on the robot. This also allows them to view realtime video from the camera and change control parameters on the field. The computer provides more than enough processing power to compute the team’s complex vision algorithms. The GPU gives the team the ability to perform parallel processing on images, providing a large increase in speed. The computer can be accessed through the robot’s wireless network or via the workstation. The computer runs Ubuntu Linux, which allows software to be edited, compiled, and tested onboard. The onboard wireless router is configured to connect to external wireless networks, allowing software changes to be pushed to the team’s GitHub repository.

SOFTWARE STRATEGY

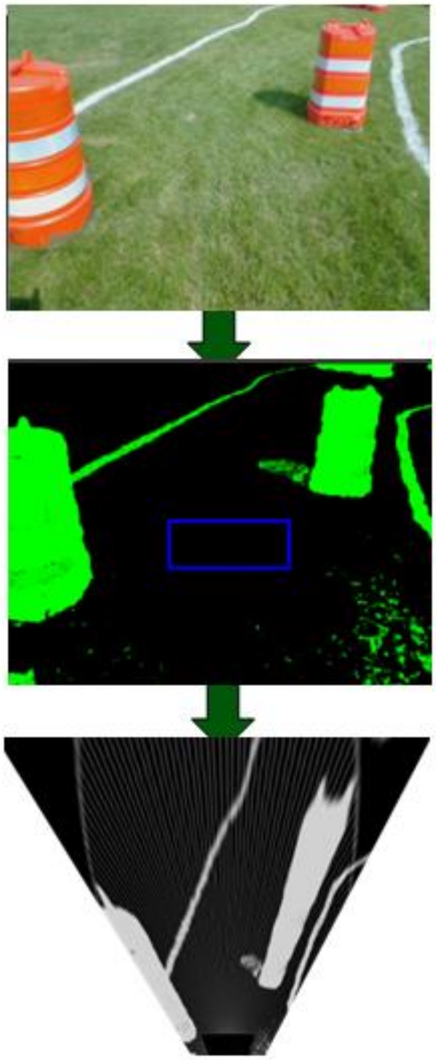


Figure 4 Image Pipeline

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.

A simplified overview of the current software architecture may be found in Figure 6.0.1. 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 a Wiimote wireless controller. The team uses Git revision control software to track changes. All software is available under the Open Source GPL v3 license. The software may be found in the team’s GitHub repository at https://github.com/MST-Robotics/S&T Enterprise_IGVC.

Vision

The primary sensor of S&T Enterprise is 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 4.

17. Current Goal (Pose)
Displays a geometry_msgs::PoseStamped message. More Information.

Topic: move_base_simple
Topic: initalpose

Wall Time: 1335383759.405459 Wall Elapsed: 399.173083 ROS Time: 1335383759.405436 ROS Elapsed: 399.173082

Figure 5 Software Map of The Lab

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.

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.

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.

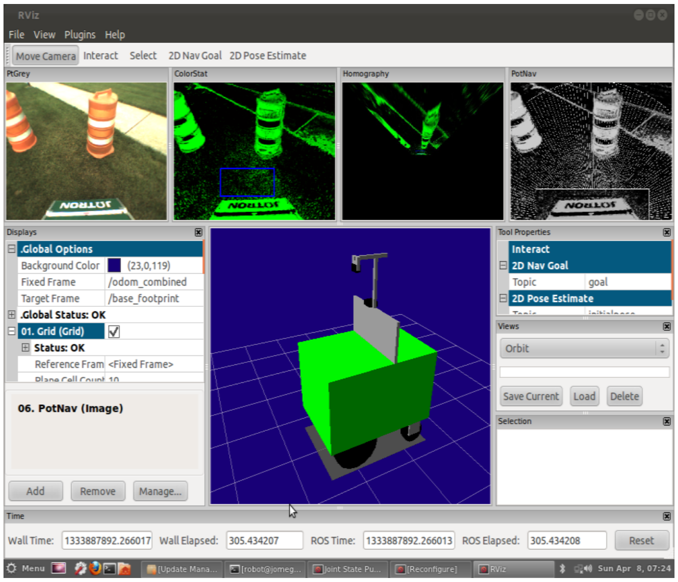
A Algorithm.* As mentioned earlier, the A* algorithm replaces the potential field model as the main form of robot navigation.

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.

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 a Wiimote controller to be connect 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 Wiimote 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. 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>
06. PotNav (Image)
Interact
2D Nav Goal
Topic goal
2D Pose Estimate
Orbit
Wall Time: 1333887892.266017
Wall Elapsed: 305.434207
ROS Time: 1333887892.266013
ROS Elapsed: 305.434208

Figure 6 Rviz Robot Dashboard

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.

SYSTEM INTEGRATION

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.

PERFORMANCE EXPECTATIONS

The robustness of S&T Enterprises 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 1.

Table 1: Performance Comparison Table

Characteristic	**Design Goal**	**Demonstrated in Field Test**
Max Speed	5 MPH (2.237 M/s)	3.2 M/s (limited to 2.2 M/s in motor controller firmware)
Ramp Climbing Ability	15 degrees	22 degrees
Reaction time - processing rate (sense-think-act loop)	4 hertz	7 hertz
Battery Life	1 hour	1.6 hours
Distance at which obstacles are detected	• Web Cameras : ○ 4 M forward ○ 3 M side – looking ○ 5 M Diagonal • Stereo Camera: ○ 10 M forward ○ 60 degree FOV	• Web Cameras : ○ 4.5 M ○ 3.2 M ○ 5.52 M • Stereo Camera: ○ 12 M ○ 65 degrees
Accuracy of arrival at way points	2 M	1.5 M
Microbotics INS/GPS	$0	$5,710
Wireless Router	$60	$60
Totals	***$60***	***$5,770***

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 S&T Enterprise 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.

SAFETY

Emergency Stop

With all of the power S&T Enterprise 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.

S&T Enterprise hardware limits its speed to just less than 5 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 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.

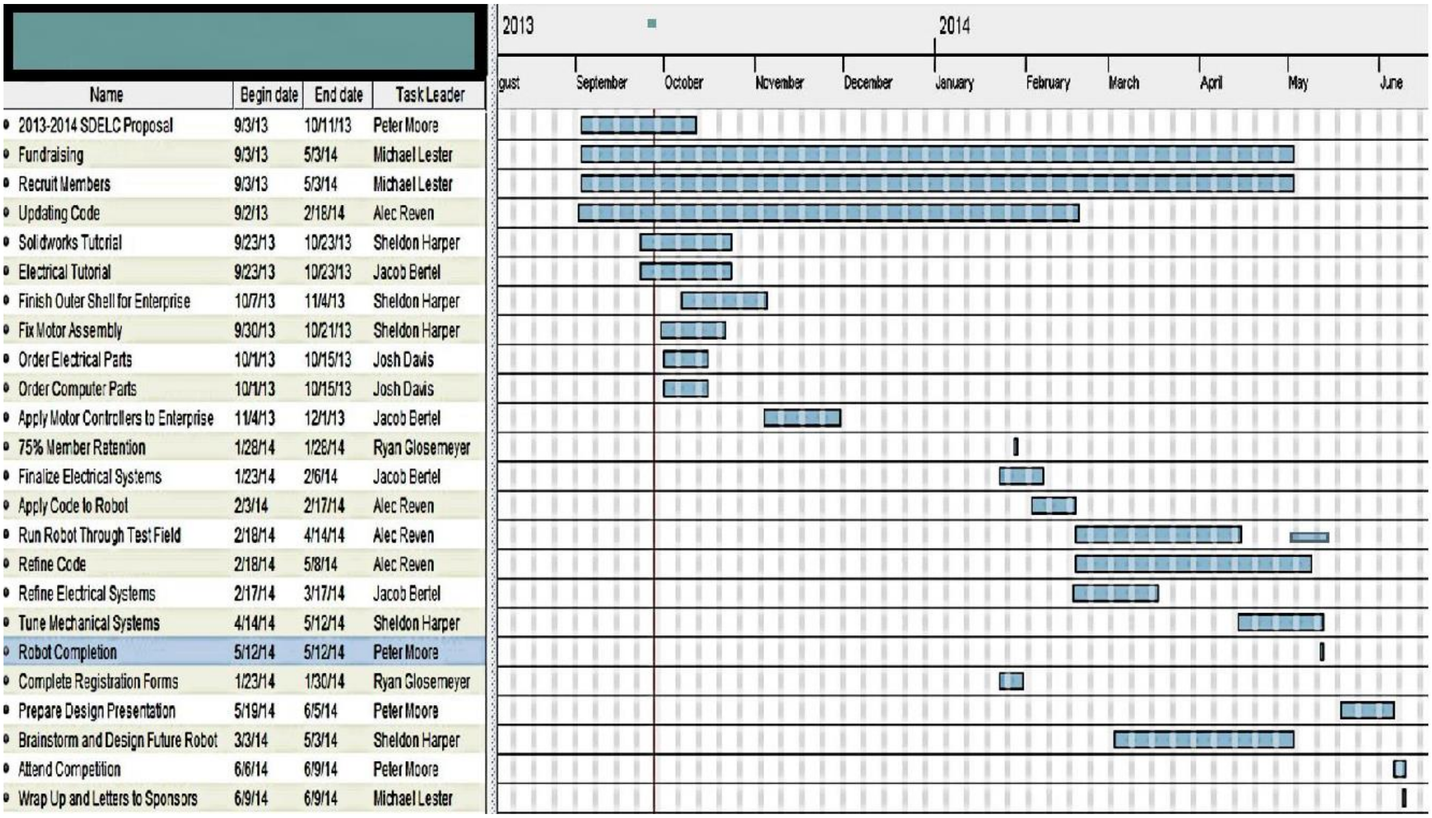
COST

S&T Enterprise's design and build process began during the fall semester of 2012 and was completed in spring of 2014. Since we are re-entering S&T Enterprise and all upgrades were made with materials that were had on hand, no additional costs were incurred.

APPENDIX A: 2013-2014 MEMBER ROSTER

Name	Position	Name	Division	Name	Division
Dr. Donald Wunsch	Main Advisor	Ryan Baysinger	Electrical	Samuel Pester	Mechanical
Dr. Douglas Bristow	Technical Advisor	Janson Beard	Electrical	John Peterson	Electrical
Peter Moore	President	Shae Bolt	Comp Sci	Christopher Siebert	Electrical
Sheldon Harper	Vice President	David Bubier	Electrical	Travis Stuart	
Joshua Davis	Treasurer	Andrew Donaldson	Electrical	Jackson Cwach	
Ryan Glosemeyr	Secretary	Christopher Durand	Comp Sci		
Michael Lester	Public Relations	Brock Ebert	Public Relations		
Alec Reven	Computing Lead	Emily Harnandez	Electrical		
Jacob Bertels	Electrical Lead	Zackary Hudgens	Electrical		
Caleb Wilcxynski	Mechanical Lead	Nicholas Johnson	Electrical		
Mathew Anderson	Comp Sci	Nicholas Marik	Mechanical		
Sean Basler	Comp Sci	Anthony Nguyen			

APPENDIX B: 2013-2014 SCHEDULE



Name	Begin date	End date	Task Leader
2013-2014 SDELC Proposal	9/3/13	10/11/13	Peter Moore
Fundraising	9/3/13	5/3/14	Michael Lester
Recruit Members	9/3/13	5/3/14	Michael Lester
Updating Code	9/2/13	2/18/14	Alec Reven
Solidworks Tutorial	9/23/13	10/23/13	Sheldon Harper
Electrical Tutorial	9/23/13	10/23/13	Jacob Bertel
Finish Outer Shell for Enterprise	10/7/13	11/4/13	Sheldon Harper
Fix Motor Assembly	9/30/13	10/21/13	Sheldon Harper
Order Electrical Parts	10/1/13	10/15/13	Josh Davis
Order Computer Parts	10/1/13	10/15/13	Josh Davis
Apply Motor Controllers to Enterprise	11/4/13	12/1/13	Jacob Bertel
75% Member Retention	1/28/14	1/28/14	Ryan Glosemeyer
Finalize Electrical Systems	1/23/14	2/6/14	Jacob Bertel
Apply Code to Robot	2/3/14	2/17/14	Alec Reven
Run Robot Through Test Field	2/18/14	4/14/14	Alec Reven
Refine Code	2/18/14	5/8/14	Alec Reven
Refine Electrical Systems	2/17/14	3/17/14	Jacob Bertel
Tune Mechanical Systems	4/14/14	5/12/14	Sheldon Harper
Robot Completion	5/12/14	5/12/14	Peter Moore
Complete Registration Forms	1/23/14	1/30/14	Ryan Glosemeyer
Prepare Design Presentation	5/19/14	6/5/14	Peter Moore
Brainstorm and Design Future Robot	3/3/14	5/3/14	Sheldon Harper
Attend Competition	6/6/14	6/9/14	Peter Moore
Wrap Up and Letters to Sponsors	6/9/14	6/9/14	Michael Lester