MISSOURI S&T University of Science & Technology

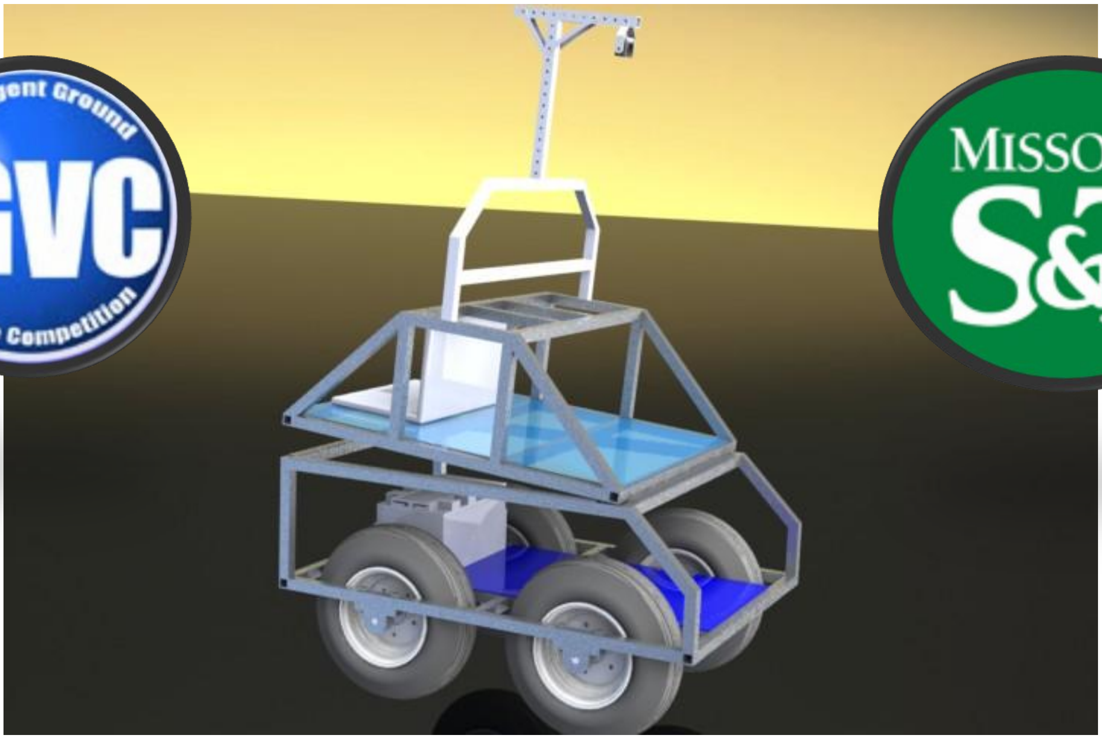
ROBOTICS COMPETITION TEAM

"Helping Robots Help Themselves"

The Missouri University of Science and Technology Robotics Competition Team

proudly presents:

JΩTRON



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 Wunsch

Dr. Donald Wunsch
Senior Advisor

1. INTRODUCTION

The Missouri University of Science and Technology (Missouri S&T) Robotics Competition Team is pleased to present "JΩtron" to the 2011 Intelligent Ground Vehicle Competition (IGVC). Joe-mega-tron, named after the Missouri S&T mascot Joe Miner, will make its first appearance at the IGVC during the seventh year that the Missouri S&T Robotics Competition Team has submitted an entry. JΩtron's design is based on improvements learned from the faults of previous robots, so JΩtron has become the robot of suggestions and imaginings. The simple and rigid frame makes JΩtron's structure ideal for the challenges presented by the IGVC. During this school year, the electrical, mechanical, and software designs were established based on the needs seen at past competitions and the desire to create the strongest robot possible. Although JΩtron is a new robot on the course, its reliability and intelligence should enable it to navigate almost any course encountered during the 2011 IGVC.

2. DESIGN PROCESS

2.1 TEAM STRUCTURE

The team operates through the Missouri S&T Student Design and Experiential Learning Center (SDELC), which provides logistical and financial support to all ten of the school's student-run design teams. The Robotics Competition Team is comprised of roughly 30 members from a variety of disciplines, and a full member list can be found in Appendix B. The team is run by five elected undergraduate officers who in turn appoint three additional officers to serve year-long terms. The team president, vice president, treasurer, public relations officer, and ecretary comprise the five elected positions. At the beginning of the school year, three division leaders are appointed, each of whom oversees and manages one of the team's following three divisions: mechanical, electrical, and computing. All of the other team members are a part of one or more of these divisions. This team structure (*Figure 1*) allows the general members of the team to operate without being distracted by the day-to-day logistics that come with running the team.

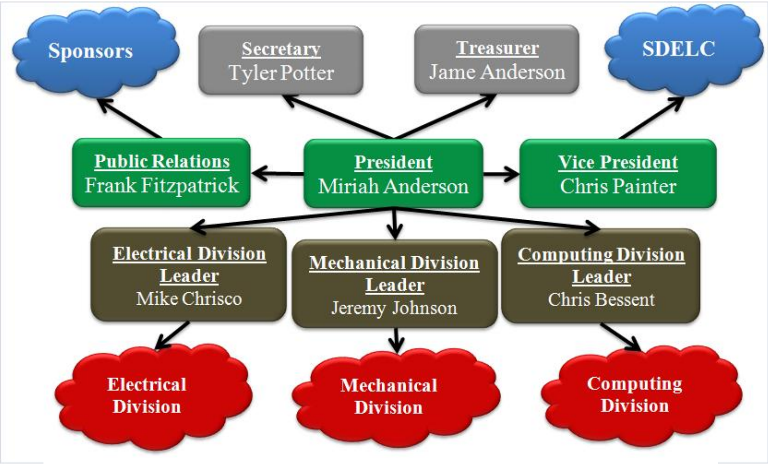


Figure 1: Team Structure

2.2 PLANNING

The 2010-11 school year started off with the creation of a design and a partial frame for JΩtron. The team's officers and advisors looked at our previous robot and decided that the best course would be to design and build a new robot that would be able to utilize much of the hardware the team already owned. Rather than basing this robot on a completely new design, the team found the previous robot's difficulties and flaws and based the new robot on improvements to the old design. JΩtron was created with the mechanical, electrical, and computing divisions working together to find the best combination for the IGVC. With the design established and motivation high, JΩtron became a reality in the 2010-11 school year. Since the new robot's architecture is similar to that of the old robot, the computing group was able to develop software for the previous robot that would run the same on JΩtron. A full schedule was laid out that included ample time to build JΩtron and test the software on both robots. Appendix A depicts the original schedule.

2.2 EXECUTION

Throughout the 2010-11 school year, the mechanical division worked on the fabrication and completion of the frame and drive train for JΩtron. Unfortunately, progress was not made as quickly as expected; the team spent time waiting on parts and trying to fix new problems that arose. As the computing division waited on JΩtron's completion, they were able to test and develop more robust software on a previous robot that could then be transferred to JΩtron. The electrical division stayed busy preparing an old robot for public relations events. About three quarters of the way through the year, the electrical division got control of the robot in order to wire it. They also ran into some difficulties as they waited for parts, but it was not long before the wheels were powered and JΩtron was alive, allowing the year to end with a few test dates where JΩtron could show its intelligence.

3. VEHICLE HIGHLIGHTS/INNOVATIONS

JΩtron incorporates previous designs but without their flaws. The new robot will bring robustness, stability, and power to this year's competition through its innovative design.

3.1 MECHANICAL UPGRADES

The previous robot experienced some issues due to the use of a rear caster, so our solution implemented four-wheel drive with custom designed gearboxes and chains. Thus, the robot grew in size as well as robustness. The previous robot also had accessibility problems, so for JΩtron, we created panels that can be removed easily with magnets, added a hydraulic spring to separate the top of the robot with a hinge in order to easily access the electrical and drive components, and placed the computer box on sliders that allowed quick and easy access to the computer parts on the top.

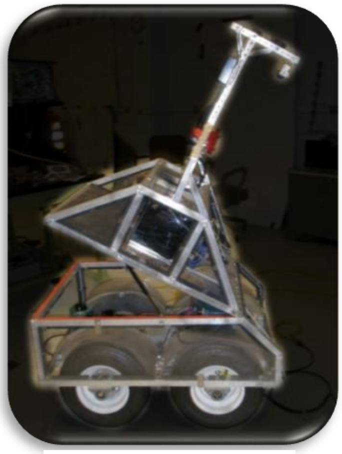


Figure 2: JΩtron

3.2 ELECTRICAL UPGRADES

With its completely new vehicle design, JΩtron features a new electrical system designed from the ground up. At its foundation, the new system employs four 12V sealed lead-acid batteries for a supply voltage of 48V. 20AH batteries give the robot about 960Whr of capacity. This is enough power for the robot to run continuously for well over an hour. The higher battery voltage is better suited for the PMDC brushless motors used in the robot's drive train. This also requires less current-handling capacity of the motor drives and wiring, resulting in less cost, size and resistive heating. The new electrical system also incorporates a power tracking system. Each of the robot's main circuits is monitored separately by a current sensor. This data, combined with battery voltage, allows a monitoring microcontroller to create a power profile for the robot. This data is fed to the robot's main computer to give precise estimates of remaining battery life. The remote E-Stop controller is also designed to provide feedback on the robot's status, including these calculated battery levels. This will ideally prevent the types of failures experienced in the past during competition and field-testing. JΩtron also features a built-in charger that can be used to quickly charge or help power the robot during testing. This improvement was also inspired by past experiences in the field. Wiring and component placement were designed to be as well organized, serviceable, and upgradable as possible. For safety reasons, because of the robot's larger size and new powerful drive system, the new electrical system also features a triple-redundant onboard and wireless E-Stop system.

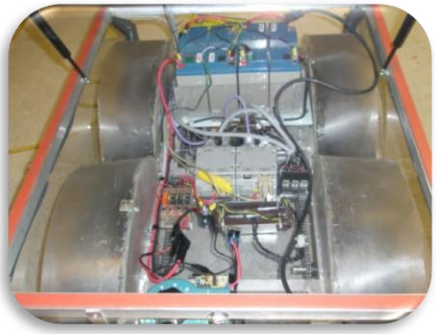


Figure 3: JΩtron's Electrical System

3.4 SOFTWARE UPGRADES

Due to JΩtron's similarities to the team's previous robot, Aluminator, most of Aluminator's software has been modified to accommodate JΩtron. The major changes feature a unique particle-based vision algorithm and a transition to using the ROS (Robot Operating System) platform.

4. MECHANICAL DESIGN

4.1 FRAME

JΩton's frame is made out of one inch by one inch aluminium square tubing and is 28" wide by 40" long. The frame itself is divided into two sections, the upper half and the lower half. The upper half is attached to the bottom using a hinge along the back side and two hydraulichydrauohydrolic springs, much like you would see on the hood or trunk of a car. This allows very easy access to the bottom portion of the robot, making it easy to change batteries, pull something out, or put something new in. The frame itself is welded together in order to ensure its endurance through many years of use. JΩtron's frame is also relatively large in order to ensure that there is enough room for everything that is essential to the robot; if and when upgrades are needed, space will not be an issue.

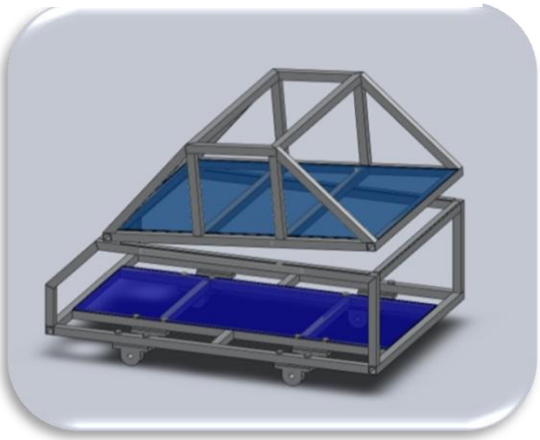


Figure 4: Solidworks Frame Design

The shell of the robot is attached using small, steel 'L' brackets that are pop-riveted at points all along the frame. The shell itself is made out of Lexan® with small magnets attached so that the magnets and 'L' brackets connect together to create a solid, easily-removable shell. This method ensures that there will always be a way to get into either the top or bottom half of the robot from any angle, even if opening the robot's hatch is not an option.

4.2 DRIVE TRAIN

For JΩtron's drive train, we utilized a "tank" style configuration using four wheels, two motors, two gearboxes and a chain system to connect the front and back wheels. The gear boxes were custom designed and fabricated by the team this year to fit the motors that were donated to the team by the team's sponsor, Elmo. Because we decided to use a chain to connect the two wheels, we had to design and install a chain tensioner to keep the chain tight and in place while the robot moves and changes direction. The chain tensioner, which sits on the bottom of the robot, and is composed of two aluminum plates with a bearing able to house a 3/8in shaft with a sprocket on the end. The chain tensioner can be adjusted for more or less tension by tightening the two bolts in a slotted hole.

4.3 CAMERA MOUNTS

Figure 5: Camera Mount

The bolted-down camera mount, which sits rigidly on top of the robot, is the same 1 by 1 inch aluminum square tubing, just like the rest of the robot. It reaches up to a height of 5 feet in order to give our camera an ea bird's-eye perspective. The camera mount consists of two parts: the upper and lower half. The lower half holds the stereo camera at a 45 degree angle, while the upper half holds the standard camera. This upper part was designed to be very adjustable, and to achieve this, there are holes every inch on the mount to guarantee the ideal camera angle, even if that angle needs modifications. The upper mount holds the camera in two places and allows the camera's downward angle to change but also keeps it locked in place while JΩtron is moving. The camera mount was designed to satisfy the needs of this year's software vision setup and to accomodate any changes that may be made in the future.

5. ELECTRICAL DESIGN

JΩtron features a completely new 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.

5.1 ACTUATORS

JΩtron features a completely new drive system compared to previous designs. The motors used on this platform are Brushless Permanent-Magnet DC motors provided by Elmo. These two motors produce up to 900W each and are driven by Elmo's 'Drum' type servo motor drive. This combination of power supply, advanced drives, and brushless motors gives this robot very high torque and maneuverability. The massive reliability and efficiency improvements over the previously-used brushed DC motors allow the robot to run longer with a limited battery life and with fewer failures like the ones experienced in the past with brushes and armatures. The new 48V power system was also chosen as the drive system's foundation because it requires less current handling capacity in the motor drives, provides longer battery life, and is better suited to the higher voltage required by our motors.

5.2 SENSORS

Many types of sensors and feedback are used on *JΩtron*. Low-level sensors handle emergency stops and power tracking on the robot. Higher-level sensors take in information about the robot's surroundings. Two types of vision sensors handle obstacle avoidance while 5 types of position sensors help determine the robot's location.

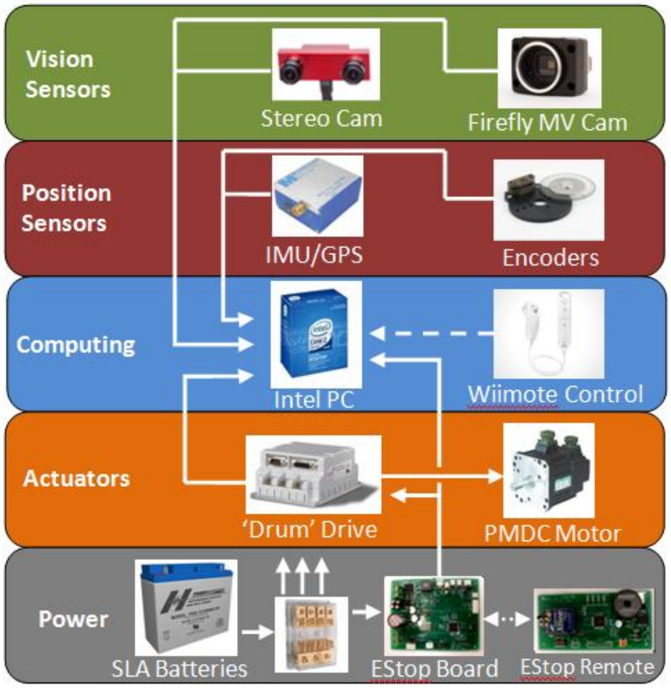


Figure 6: Simplified System Diagram

• STATUS SENSORS

At a low level, power tracking and E-Stop systems are handled by an independent AVR microcontroller. This ensures 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 warnings via multiple interfaces when battery levels are dangerously low. *JΩtron* also features a two-way wireless interface that provides E-Stop functionality as well as feedback about the robot's current status. Finally, the E-Stop system features triple redundancy to help guarantee safely controllable operation and testing. An E-Stop condition from the PC, onboard switches, or remote triggers a serial command as well as an electrical signal to the motor drives to shut down. This signal also triggers physical brake relays to be engaged to stop the robot quickly.

• VISION SENSORS

JΩtron employs a single Point Grey Firefly MV camera. This camera operates at a resolution of 0.3 megapixels and is able to supply VGA (640 * 480) 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.

JΩtron also employs a Videre Design STOC (Stereo On a Chip) stereo camera (*Figure 7*). This camera performs real-time image correlation using an on-board FPGA, and it runs at 30 FPS. The stereo camera also connects to JΩtron's computer via a Firewire (1394a) interface.

• POSITION SENSORS

JΩtron 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.

5.3 COMPUTING

All computing is done on JΩtron's onboard computer. JΩtron contains a full-size, custom-built desktop featuring a Core 2 Quad Q9400 2.66GHz processor and an nVidia Geforce GT 430 graphics processing unit. This computer provides more than enough processing power to compute complex vision algorithms and is essential for running software. The computer is linked to a wireless router through which any other computer can use S.S.H. to make changes. The computer runs Ubuntu Linux, which allows the code to be edited and compiled directly on the robot and also allows changes to be made to the team's Bazaar Revision Control Server.

6. SOFTWARE STRATEGY

The high-level functions are separated into ROS packages that are described below. These separate packages help to organize the code, making it easier to edit or even replace whole packages without the user needing to comprehend the program as a whole.

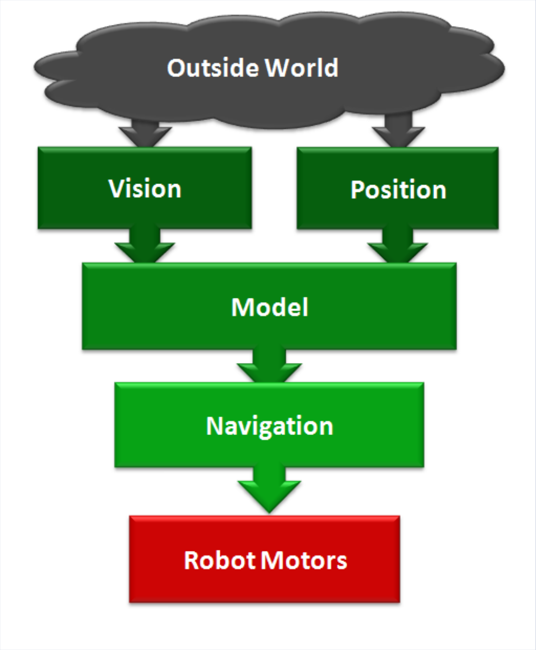


Figure 7: Software Architecture

- ***VISION***

JΩtron detects lane line and obstacle features in the outside world using optical imaging cameras. Basic obstacles are classified using stereo perception, color segmentation, and spatial filtering. Obstacle feature recognition is performed using advanced image processing techniques, such as Hough filtering and curvilinear trend analysis.

- ***POSITION***

A multitude of sensors, including a GPS, 3-axis accelerometer, 3-axis gyro, 3-axis magnetometer, and wheel encoders, determine the relative and absolute positions of the robot.

- ***MODEL***

The model integrates vision and position information to build a map of the local environment. It also generates synthetic ray casts for use with some of the guidance algorithms based on LIDAR-like range finder sensors.

- **NAVIGATION**

The navigation system determines the optimal path through the local environment and issues commands to the Motors module to move the robot.

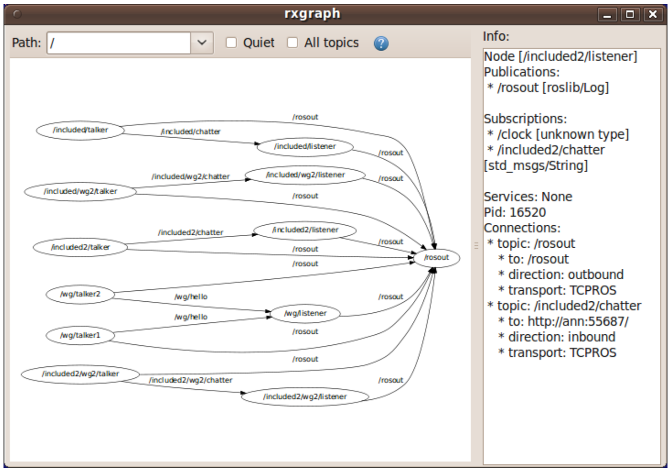
- **MOTORS**

The robot's motors convert high-level movement commands generated in the Navigation module to low-level commands used by the motor controller hardware.

All of the team's software is written in C++ using an object-oriented design. OpenCV is used extensively to support vision functions. The classes supporting a module are located in its module package. A separate package is utilized for common information (configuration and calibration), diagnostics (images / video sequences taken during testing), and advanced techniques under development. The entire software suite is managed by Bazaar revision control software. All team members are given access to and trained how to utilize the Bazaar system to facilitate coordinated parallel development. The flexibility of this software and repository system has allowed the team to test numerous strategies, and, over time, to come to the software solution that is being used today.

6.1 ROS (ROBOT OPERATING SYSTEM) PLATFORM

A major change in our software philosophy has been to incorporate ROS into our robotic architecture. ROS (http://www.ros.org) is an open-source, community-based project to standardize, facilitate, and share software for robotic platforms. The underlying platform, mostly created and updated by Willow Garage and Stanford, provides a great base for developing software. The main features of the ROS platform and the benefits the team derives from it are described below.



```
Node [/included2/listener]
Publications:
 * /rosout [roslib/Log]

Subscriptions:
 * /clock [unknown type]
 * /included2/chatter
[std_msgs/String]

Services: None
Pid: 16520
Connections:
 * topic: /rosout
    * to: /rosout
    * direction: outbound
    * transport: TCPROS
 * topic: /included2/chatter
    * to: http://ann:55687/
    * direction: inbound
    * transport: TCPROS
```

Figure 8: Display of Running ROS Nodes

The first notable feature ROS provides is a higher level of code development. ROS simplifies and abstracts many of the tedious or complicated tasks associated with using an operating system. Interprocess communication is easily achieved through a publisher-subscriber scheme using defined messages that correctly and efficiently control data flow between modules. Tasks such as module synchronization are easily achieved through the use of built-in libraries. These features allow team members to focus on developing techniques and algorithms rather than worrying about menial tasks.

A large, active, open-source codebase is another great feature of the ROS platform. If you are looking to achieve a specific purpose with ROS, there is likely already a package that can support that purpose. Drivers for specific hardware such as GPS receivers and various cameras are readily available, allowing the team to concentrate on other areas.

The migration to an ROS-based platform was moderately difficult as team members had to adjust to a new environment; however, it was, without a doubt, a positive move. The platform provides tremendous benefits for development and will also provide a stable base for development in the years to come.

6.2 SIGNAL PROCESSING

COLOR SEGMENTATION

Our system detects the boundaries of the lane and obstacles using a particle-based physics simulation. Safe zones, where no obstacles should exist, are defined as injection sites for the particles. The sensor image is then processed to create a gradient relative to these injection sites. The gradient is based on relative brightness and hue, with larger differences creating more energy.

Independent particles are then injected with a random velocity. Each particle keeps track of its starting information and current velocity. As the particle progresses through the simulation, it is affected by the sensor image gradient as a way to detect changes. When it reaches an area of high energy, the particle's velocity will decrease; if the energy, or area of energy, is sufficiently large, the particle will rebound in the opposite direction.

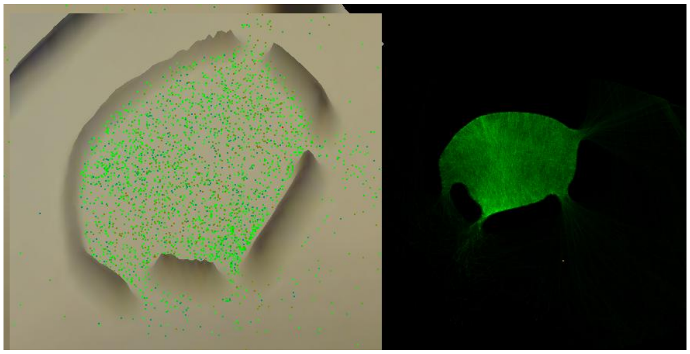


Figure 9: Left shows the individual particles and the right is the map generated images from the Firefly MV Camera

After each iteration of the simulation, the position of each particle is noted and recorded. The interplay between the energy gradient and thousands of particles produces a probabilistic determination of the continuous safe areas the robot can reach. The final determination is then put through a homography transform to create an accurate representation of the field that is used by one of our guidance modules.

This approach offers several benefits over traditional image processing techniques. First, no training or calibration is required for specific testing fields or obstacles. Second, because each particle is independent, the power of JΩtron's GPGPU (general purpose graphics processing unit) can be harnessed using CUDA to make many of these calculations simultaneously. Third, the probabilistic nature this approach automatically provides a filter for visual noise, whether it comes from the sensors, the field, or errant processing.

STEREO IMAGE PROCESSING

The stereo camera generates a point-cloud of (x,y,z) points (Figure 10) for all image points passing through a correlation reliability filter. To separate the real-object voxels from mis-correlated noise, the point-cloud is rotated and rectified to fit the same ground plane used by the web camera array. A statistical analysis of this overlay provides additional information about which

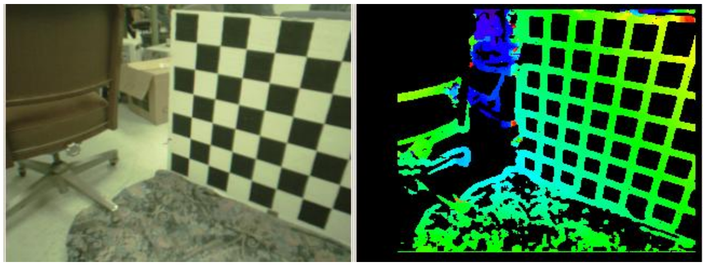


Figure 10: Stereo Camera Disparity Map

stereo voxels represent real-world objects.

In addition, stereo cameras map obstacles, such as suspended plank saw horses, to a more accurate positions than do web cams. This is due to the projective nature of the perspective rectification performed for the web cams. Web cams will correctly map the positions of objects touching the ground plane; however, the positions of obstacles that float above the ground plane (like planks on a saw-horse) will appear to be further away than they actually are. This situation is corrected by using the stereo-camera version of the same obstacle to correctly reposition it in the model space.

GPS / POSITION SIGNAL CONDITIONING

The on-board MIDG-II microcontoller is capable of resolving differences in position as reported individually by its internal GPS and accelerometers. This requires careful calibration to work properly. An internal Kalman filter determines the most probable position by taking into account the inherent time scale and the accuracy characteristics of both the GPS and the accelerometers.

6.3 GUIDANCE

JΩtron employs two different guidance algorithms, which can be selected depending on the type of situation.

RAY-CAST METHOD

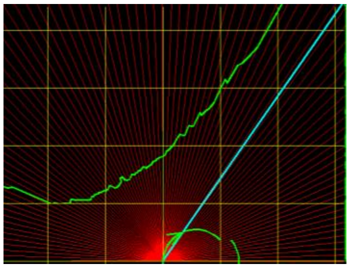


Figure 11: Ray-Cast Debugging Display

A ray-cast-based algorithm is utilized in situations where guidance methods developed under the player-stage simulation environment closely mimic the real world.

The image (*Figure 11*) to the left illustrates the ray-cast algorithm. The red lines (every degree) are the ray casts. The green line represents the obstacle detected (a white lane line on grass). The blue line represents the chosen direction and is the longest path available that provides sufficient clearance for the width of the robot. The curved green arrow at the bottom center represents the path of the robot if this vector were pursued for 3 seconds. The yellow lines represent a 1-meter grid overlay for reference.

POTENTIAL FIELD METHOD

A Potential Field algorithm is utilized in which more complex types of obstacles are expected. After web-cam images are rectified, combined, and color-segmented, a binary model is generated (*Figure 12*).

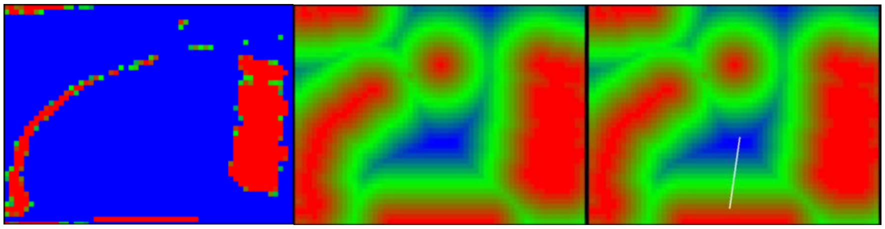


Figure 12: Potential Feild Model

When using the potential field method, obstacles are represented as hills, and way points are represented as depressions. A “robot repulser” hill is dynamically built behind the robot to encourage the robot to move in generally a forward direction. A simple gravity model is employ to propel the robot through the potential field. Local minimums are avoided by thresholding the surface, adjusting the strength of the repulser hill, and, optionally, tilting the entire potential surface to favor flow toward the target position.

A reduced-resolution field strength is determined using a statistical surface derived from the relative density of pixels, classified as "obstructions." A potential field is then built from the field strength surface, resulting in a topographic surface of the local obstruction environment.

The robot is located just above the center bottom in this potential field image. To provide JΩtron with an incentive to generally move forward (in a robot-centric space), a "repulser" bar object is added to the bottom of the potential field. Next, the immediate local area is searched to determine a target position to move to. This location will always be "downhill" from where the robot is currently positioned within this potential field.

Previously, the combination of forward speeds and rotation rates necessary to travel to every reduced resolution potential field cell was calculated and stored in a look-up table. Once the target potential field cell is identified, the forward and rotational velocities are derived from this look-up table (*Figure 13*) and passed on as motor commands to the motor controller module. Careful calibration and tuning of the curvilinear trend analysis parameters is required to achieve reliable results with this method.

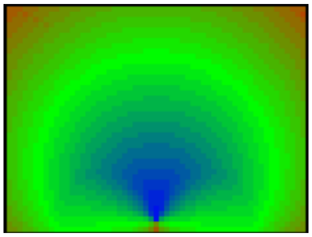


Figure 13: Motor Velocity Map

6.4 JAUS

JΩtron is partially compliant with the SAE-JAUS standard through its use of the JAUS++ library developed by the University of Central Florida - Institute for Simulation and Training – ACTIVE Laboratory. JΩtron will report all required information about its state and intentions to the JAUS controller. Currently, JΩtron will not acknowledge any commands or requests from the JAUS controller.

7. SYSTEM INTEGRATION PLAN

JΩtron'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 accomplished 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.

8. PERFORMANCE EXPECTATIONS

The whole team feels very confident in JΩtron's ability to compete in the 2011 IGVC. The robustness of its algorithms have been proven time and time again in simulations. The new design is much more reliable and powerful than the previous robots. The team's predictions, along with the design's demonstrated values, can be found in *Table 1* below. The team has taken steps to ensure that spare parts for each of the robot's components will be available at the 2011 IGVC. In short, the Missouri S&T Robotics Team expects JΩtron to achieve the university's best showing ever in the IGVC and to finish among the top ten in the field.

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

Table 1: Performance Comparison Table

8.1 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, JΩtron will seek the path of least resistance. When no such path is obvious within 190 degrees of the front view, JΩtron will turn 180 degrees and examine the rear environment for a potential exit path. The limited obstacle model memory will discourage JΩtron from repeatedly taking the same path.

• CENTER ISLANDS

JΩtron will tend to drive toward an area equidistant from all obstacles, be they lane lines or barrels. This forces the robot to choose the widest path available, thus avoiding barrels in the center of a wide lane.

- ### DEAD ENDS

JΩtron retains a short-range memory of objects visited in the past few dozen cycles. If JΩtron 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, JΩtron employs a method similar to that used to detect and navigate out of dead ends.

- ### POTHOLES

JΩtron will avoid all potholes provided they are a sufficiently different color than the grass. The stereo camera is capable of detecting slight depressions in the ground plane, even if not revealed by a unique color signature.

- ### DASHED LANE LINES

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

9. SAFETY

JΩtron is equipped with three emergency stops: one hard and two soft. The hard stop consists of two different push buttons located on opposite sides of the camera mount. These are normally closed stops, meaning that any break in the circuit will automatically trigger the stops, even if it is simply a wiring fault), and they are connected directly to JΩtron's motor controllers. JΩtron may also be stopped by a switch on our wireless E-Stop remote or through the use of the drive controls from a wireless laptop.

Power tracking and E-Stop systems can be handled by an independent AVR microcontroller, thus 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 to receive warnings via multiple interfaces when battery levels are dangerously low. JΩtron also features a two-way wireless interface that provides E-Stop functionality as well as feedback about the robot's current status. Finally, the E-Stop system features triple redundancy to help guarantee safe and controllable operation and testing. An E-Stop condition from the PC, onboard switches, or remote triggers a serial command and an electrical signal to the motor drives to shut down. This signal also engages physical brake relays to stop the robot quickly.

By controlling JΩtron with the Wii gaming console's Wiimote controller, the team significantly increased their control over the robot. The team can link the Wiimote and drive manually to the testing areas, and then the remote can be placed in standby mode to allow for autonomous testing. JΩtron's hardware limits its speed to just under 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 Wiimote 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.

10. COST IN DOLLARS/HOURS

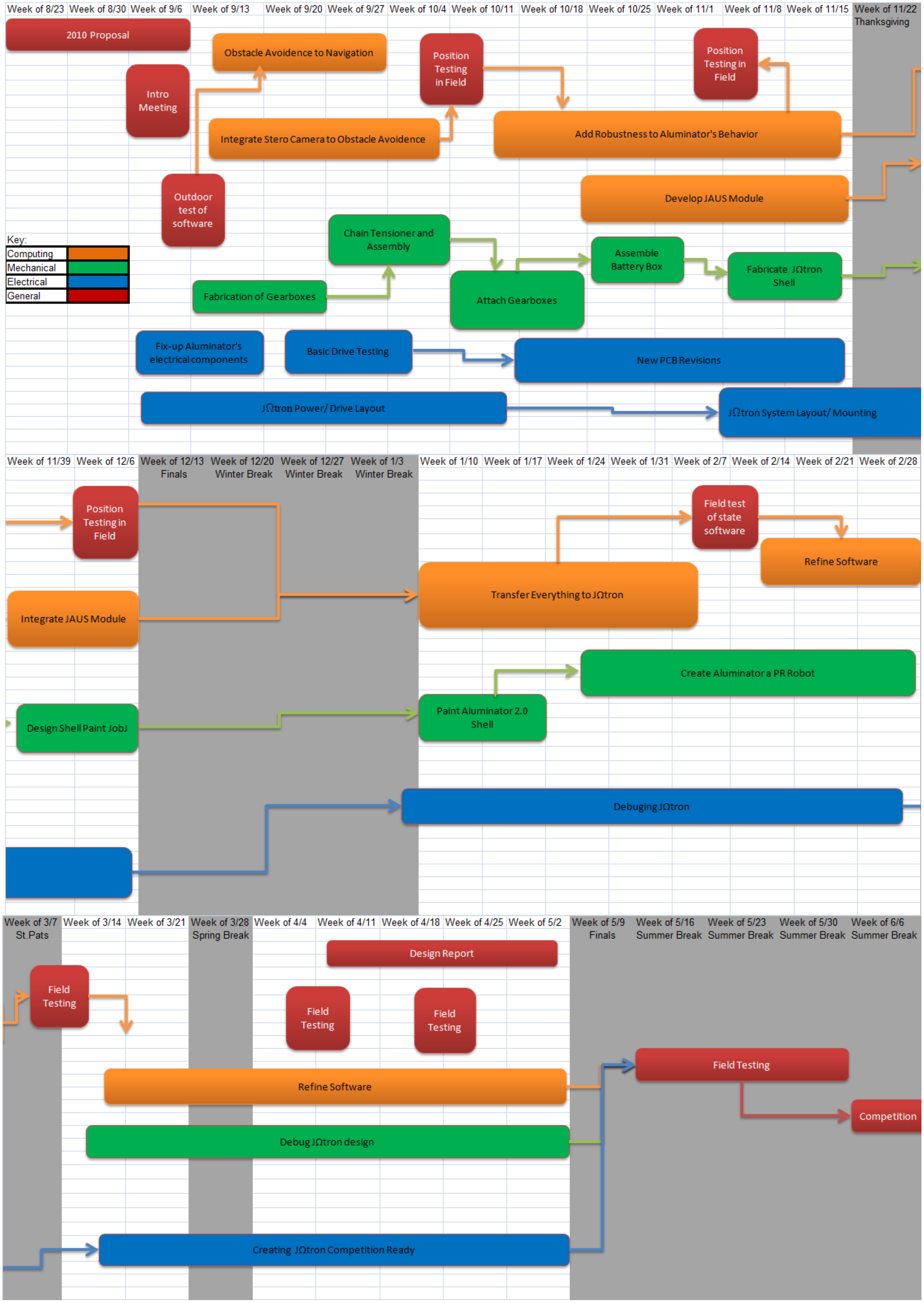
JΩtron's design and build process began during the fall semester of 2009 and was completed in spring of 2011. JΩtron's total financial cost is show below in *Table 1*.

Component	Cost to Team	Retail Value
Frame	$500	$500
Motors	$1,300	$100
Gear Boxes	$300	$300
Shell	$250	$250
Wheels	$120	$120
Misc. Hardware	$100	$100
Misc. Electrical	$200	$200
Elmo Motor Controllers	$4,000	$4,000
Batteries	$260	$260
Charger	$200	$200
Power Supply	$300	$300
Point Grey Camera	$700	$700
Videre Stereo Camera	$1,450	$1,450
Computer	$800	$800
Blue-tooth Control	$0	$65
Microbotics INS/GPS	$0	$5,710
Totals	***$10,480***	***$15,055***

Table 1: Cost Analysis

JΩtron's construction and programming required a large amount of man hours. On the whole, the team spent an average of 41 man hours per week preparing JΩtron for this competition. Over the past year and a half of the design and build process, this amounts to a conservative estimate of 2,000 man hours.

APPENDIX A: 2011 SCHEDULE



APPENDIX B: MEMBERSHIP LIST

Name	**Level**	**Discipline**
Anderson, James (Treasurer)	Senior	Computer Engineering
Anderson, Miriah (President)	Junior	Mechanical Engineering
Bessent, Chris (Computer Division Lead)	Junior	Computer Engineering
Bertel, Jacob	Freshman	Computer Engineering
Briggs, Emily	Senior	Mechanical Engineering
Calvert, Benjamin	Freshman	Mechanical Engineering
Carroll, William	Freshman	Mechanical Engineering
Chrisco, Michael (Electrical Division Lead)	Senior	Electrical Engineering
Cwach, Jackson	Freshman	Electrical Engineering
Eisenbraun, Max	Senior	Computer Science
Farris, Chris	Freshman	Mechanical Engineering
Faulkner, John	Freshman	Computer Engineering
Fitzpatrick, Frank (PR Officer)	Sophomore	Mechanical Engineering
Herrington, Shawn	Sophomore	Mechanical Engineering
Honse, Adam	Junior	Computer Science
Ho, Brian	Senior	Computer Science
James, Nathan	Graduate	Business
Johnson, Jeremy (Mechanical Division Lead)	Sophomore	Mechanical Engineering
Marik, Nicholas	Freshman	Mechanical Engineering
Mulvaney, Ethan	Freshman	Mechanical Engineering
McQuay, Sean	Junior	Electrical Engineering
Novosad, Sam	Freshman	Electrical Engineering
Painter, Chris (Vice President)	Sophomore	Mechanical Engineering
Peterson, Daniel	Sophomore	Electrical Engineering
Phan, Dan	Junior	Computer Engineering
Potter, Tyler (Secretary)	Senior	Mechanical Engineering
Santoro, Nick	Freshman	Undeclared
Siebert, Chris	Freshman	Computer/Electrical Engineer
Sites, Paul	Freshman	Computer/Electrical Engineer
Terneus, Matt	Freshman	Computer Engineering