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

For entry in the 2012 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 Wunsch

Dr. Donald Wunsch
Senior Advisor

TABLE OF CONTENTS

0.0 INTRODUCTION

The Missouri University of Science and Technology (Missouri S&T) Robotics Competition Team is proud to present " JΩtron " as its entry in the 2012 Intelligent Ground Vehicle Competition (IGVC). Joe-mega-tron is named after the Missouri S&T mascot Joe Miner. The robot will be making its second appearance at the IGVC as the eighth consecutive entry from the Missouri S&T Robotics Competition Team. The team has designed Jomegatron using lessons it has learned from the robots of previous years. The simple design of Jomegatron makes it ideal for the challenges presented by the IGVC. Jomegatron's design has been greatly improved since its performance in last year's competition. The robot has seen improvements in reliability, mobility, and intelligence. Jomegatron has become a very dependable platform which should be able to navigate almost any course encountered during the 2012 IGVC.

1.0 DESIGN PROCESS

1.1 TEAM STRUCTURE

The team operates through the Missouri S&T Student Design and Experiential Learning Center (SDELC), which provides logistical support to all ten of the school's student-run design teams. The team is comprised of roughly 20 members from a variety of disciplines. A full member list may be found in *Appendix A*. 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 secretary 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 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.

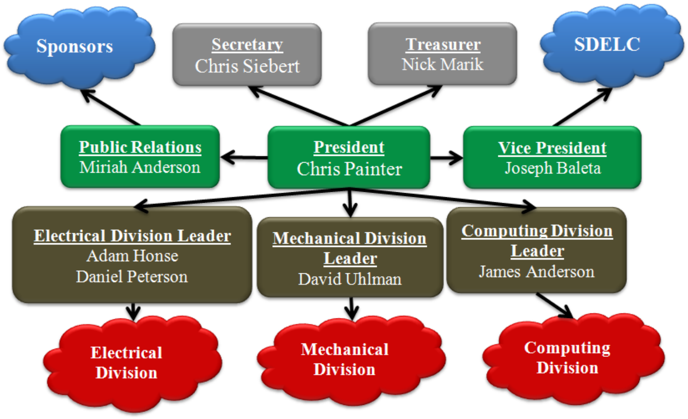


Figure 1: Team Structure

1.2 PLANNING

During the 2011 IGVC Jomegaton performed admirably and was able to successfully navigate large sections of the course. The primary goal set forth by the team at the beginning of the 2011-12 school year was to maintain and build on this success. To accomplish this goal the team looked for the rough spots in Jomegatron's performance and designed ways to improve performance. During the competition Jomegatron had several problems with the reliability of its mechanical and electrical systems. The team decided to redesign the drive platform and embedded circuits of the robot. Jomegatron's software also had several problems during competition and lacked some functionality. The team decided to implement several new methods in the software that could be used in conjunction with old software. The new additions to Jomegaton were designed to add capabilities and overall robustness to the platform without destroying any progress that was made the previous year. A full schedule was laid out, including ample time for the design and implementation of changes as well as several testing dates to ensure the desired operation. The original Schedule may be found in *Appendix B*.

1.3 EXECUTION

During the 2011-12 school year the team was able to stick to its schedule fairly tightly. The mechanical division was able to perform needed maintenance and implement various upgrades to Jomegatron. The robot's mobility and structure were greatly improved. The electrical division stayed on schedule and was able to improve the charging and control circuits of the robot. The computing division was able to implement new mapping and Joint Architecture for Unmanned Systems (JAUS) capability as well as improving the performance of previous software. The team was also able to add several new software strategies to its code base that were not originally planned. Several tests were performed throughout the year and improvements were seen in all areas. Overall, the team stayed on track and made many improvements to Jomegatron. With the new designs applied to Jomegatron it is a much better robot than it was last year and the team believes that it will make an impressive run in the 2012 IGVC.

2.0 VEHICLE HIGHLIGHTS/INNOVATIONS

By building on the lessons of previous years the team has created a robot fine-tuned to the needs of this competition. The problems of the 2011 model have been resolved, new capabilities have been added and the overall reliability of the platform has been increased. Underneath the Lexan shell of the 2012 model there have been major innovations in both hardware and software that will give Jomegatron the edge in this year's competition.

2.1 MECHANICAL UPGRADES

While most of the frame has remained the same there were major renovations made to the drive train (*Figure 2*). Jomegatron experienced some problems turning with the 4-wheel skid-steer drivetrain of the previous year. To alleviate this, a caster system was implemented. The front wheels remain independently powered, while the back wheels were replaced with casters. Casters were placed in the both rear corners of Jomegatron to provide a high level of stability. A hinge was also added to the camera mast on top of Jomegatron, allowing it to fold down for transport. This allows the team to fit Jomegatron in the back of a pickup truck for easy transportation.

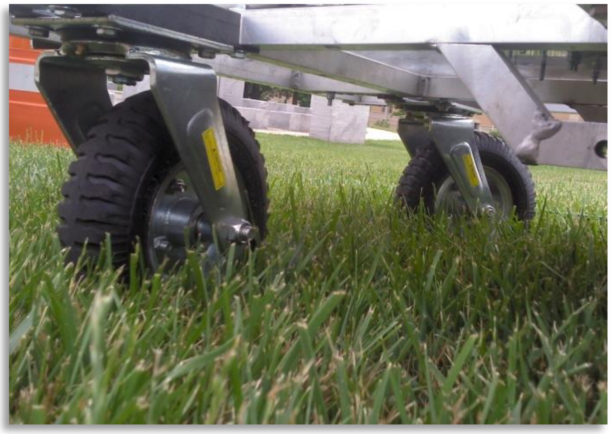


Figure 2: Caster System

2.2 ELECTRICAL UPGRADES

This year Jomegatron was upgraded with a new Emergency Stop (E-stop) control board. During the board's replacement, some upgrades were made. The new E-stop board is triple redundant, as was its predecessor. The computer, remote control, and physical buttons on the platform can all be used to E-stop the robot. There is now a switch on the E-stop board so that software control can by bypassed in the event that a board failure occurs. Even if software control is bypassed, the hardware buttons still function properly. The E-stop board also serves as a power regulator, with the ability to provide 24V for the E-stop switches and 5V for the onboard micro-controller. The 5V supply can also be used to drive other small boards that the team may want to add in the future.

The E-stop board has two methods of communication: serial and USB. The E-stop board includes a USB to serial converter, allowing the board to be plugged directly into the computer. The serial port also allows the board to communicate to a remote E-stop switch. The board is able to wirelessly send data back to the remote E-stop with battery measurements or other information the team wishes to display. With sixteen general purpose IO pins, the E-stop board can be easily upgraded to provide additional functionality as the team deems it necessary.

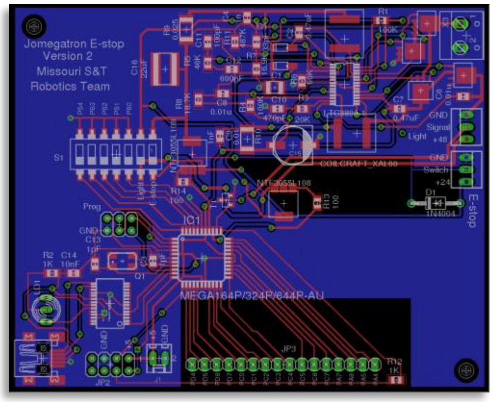


Figure 3: Custom E-Stop Board

2.3 SOFTWARE UPGRADES

The primary goals of the software team were to add mapping and JAUS capabilities as well as improving the vision and navigation software. The operation of the previous year's software has been left entirely intact by adding additional modules to the software stack. The team now has the capability to choose from several different strategies for vision, mapping, and navigation. All strategies may be chosen at runtime, allowing the team to simultaneously develop several different strategies and easily test them in the field. In addition to new capability the software now interfaces with the RViz display program to present map and sensor data, and allow for the input of waypoints. The new software has made Jomegatron better able to handle complex situations and gives users a much greater amount of feedback.

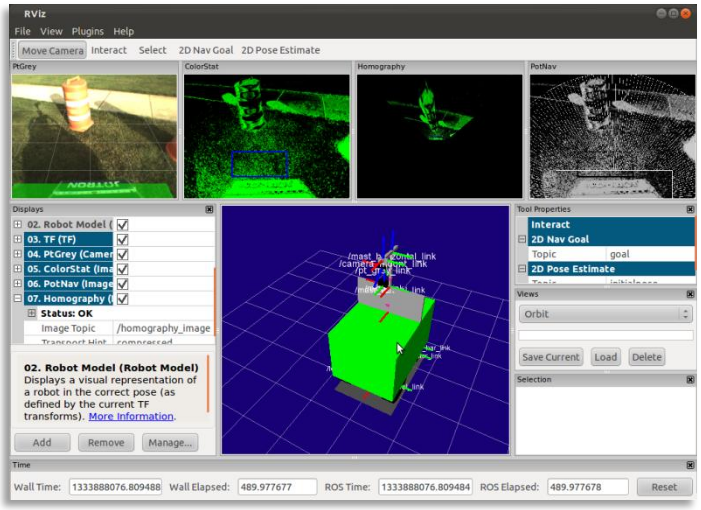


Image Topic: /homography_image

Transport Hint: compressed

Topic: goal

02. Robot Model (Robot Model)
Displays a visual representation of a robot in the correct pose (as defined by the current TF transforms). More Information.

Wall Time: 1333888076.809488 Wall Elapsed: 489.977677 ROS Time: 1333888076.809484 ROS Elapsed: 489.977678

Figure 4: RViz User Interface

3.0 MECHANICAL DESIGN

Jomegatron's mechanical design was built around simplicity and flexibility. The robot frame was built to be simple and allow easy access to internal components. The drive train gives the robot a zero point turn radius for maximum mobility, and the robot's profile allows it to fit through standard doorframes. The robot includes several locations that provide flexible mounting solutions for various sensors. The overall mechanical design provides a robust platform for all of the team's development needs.

3.1 FRAME

Jomegatron's frame is made out of 1 inch by 1 inch aluminium square tubing and is 28 inches wide by 40 inches 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 and two hydrolic 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 equipment out, or put new components in. The frame itself is welded together in order to ensure its endurance through many years of use. Jomegatron's frame is also relatively large in order to ensure that there is enough room for components. Extra room was designed to allow for easy upgrades.

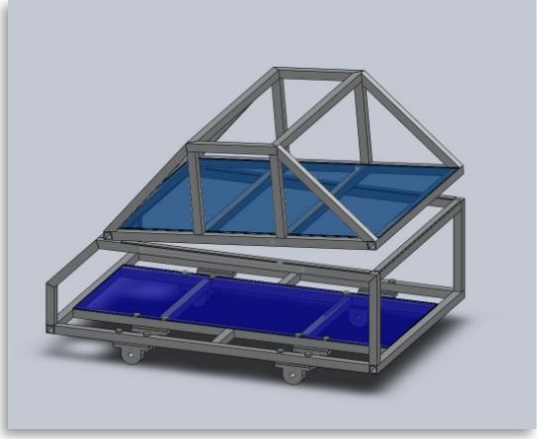


Figure 4: Solidworks Frame Design

The shell of the robot is attached using small, permanent magnets. The magnets attach to steel contact points on the frame, allowing for easy attachment and removal. This method ensures that there will always be an easy way to remove the shell for access to internal components.

3.2 DRIVE TRAIN

For Jomegatron's drive train, the team utilized a differential steering configuration with two powered wheels in the front and two caster wheels in the back. The gearboxes were custom designed and fabricated by the team to fit motors that were donated by the team's sponsor, Elmo. The two caster wheels are mounted on steel angle iron for strength and placement purposes. The wheels are mounted just inward of the rear corners to provide maximum stability, but do not extend far enough past the frame to interfere with obstacles. These components are bolted on so the team can easily make adjustments or perform maintenance on the drive train.

3.3 CAMERA MOUNT

The bolted-down camera mount, which sits rigidly on top of the robot, is made of the same 1 inch by 1 inch aluminum square tubing that comprises the frame. The mast reaches up to a height of 5 feet in order to give the robot's camera a bird's-eye view. The camera mount consists of two parts: the upper and lower half. The lower half holds the display, while the upper half holds the wide angle camera. This upper part was designed to be very adjustable. There are holes every inch on the mount to guarantee the ideal camera angle. 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 Jomegatron is moving. The camera mount was designed to satisfy the needs of this year's software and to accomodate any changes that may be made in the future.

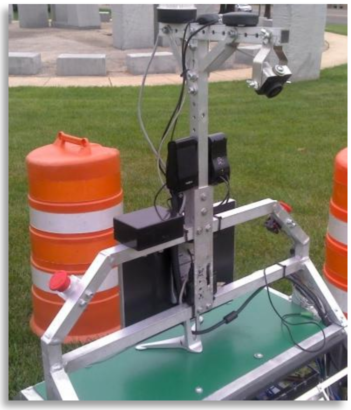


Figure 5: Camera Mount

4.0 ELECTRICAL DESIGN

Jomegatron 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 any foreseeable need for service or significant changes.

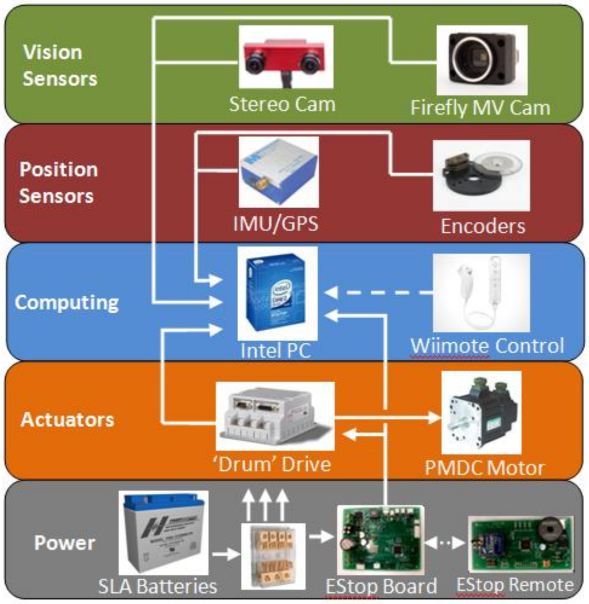


Figure 6: Simplified Sysem Diagram

4.1 SENSORS

- ### VISION SENSORS

Jomegatron employs 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

Jomegatron 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.

4.2 COMPUTING

Jomegatron carries a full desktop computer to handle all of the vision, mapping, and navigation tasks the team requires for competition. The computer has an 2.2 GHz Intel quad core processor, and a GT 430 Nvidia graphics card 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.

The computer runs Ubuntu Linux and may be controlled through the onboard wireless router or via the monitor, keyboard, and mouse mounted to the robot. This allows the team to easily make and test changes to the software. The robot automatically connects to external wireless networks, allowing software changes to be pushed to the team’s software repository. Debugging information can be displayed on the monitor, along with graphical interfaces for changing code parameters. A second monitor mounted to the front of the robot is used to display information to third parties during testing as well as public relations events.

4.3 ACTUATORS

Jomegatron uses two permanent magnet synchronous motors to drive the main wheels. The motors are controlled using “Drum” motor controllers from Elmo Motion Control. Each motor is capable of supplying 900W of power, which gives Jomegatron high maneuverability even on rough terrain. Permanent magnet machines are known for their high torque output, meaning that Jomegatron won't stall under load. The motors are rated for 100 volts each, as are the motor controllers. This leaves room to upgrade Jomegatron's power system in the future if the team feels that such an upgrade is necessary.

4.4 POWER

Jomegatron has a custom power supply to run all of the computer components. The power supply runs directly off the 48 volts supplied by the batteries with no converters necessary. It can power monitors, the computer, the camera, speakers, and a router. The power supply is capable of supplying up to 500 watts, which leaves the team a lot of room to add more computer components if necessary in the future.

5.0 SOFTWARE STRATEGY

Jomegatron’s software was programmed in C++ and designed around Robot Operating System (ROS). 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 7*. 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 Graphical User Interface (GUI) to display debugging data and allows users to provide inputs 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/Jomegatron_IGVC.

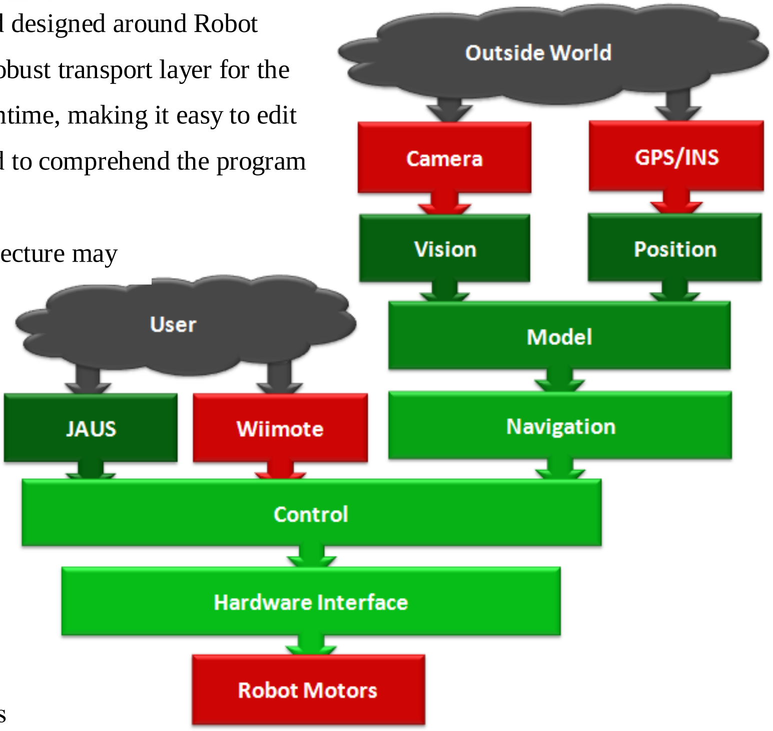


Figure 7: Software Architecture

5.1 VISION

The primary sensor of Jomegatron is a 640x480 resolution wide angle camera. The first step of the vision pipeline is to identify obstacles. As described below, the team has developed two primary methods for finding obstacles. 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 distribution curves based on the input colors' chromaticity and brightness 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, 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.

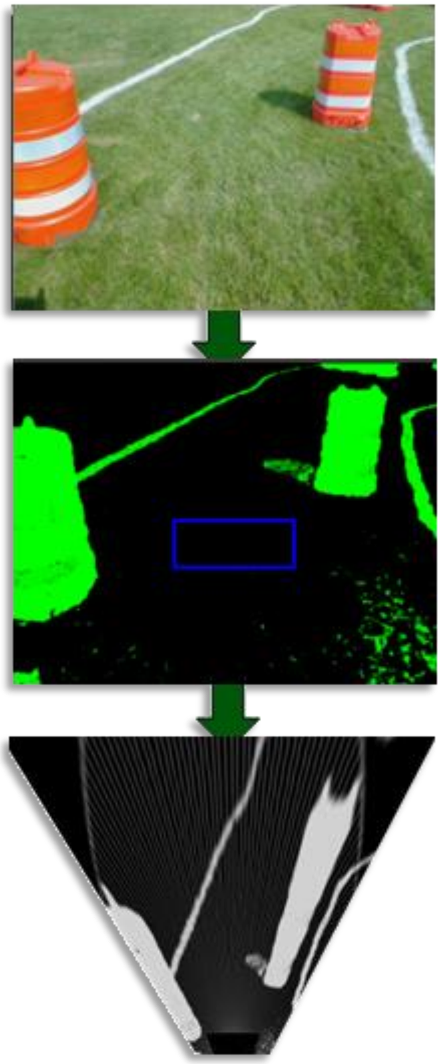


Figure 8: Image Pipeline

The modules may be launched separately or may be used together with their outputs combined. The slalom flags of the competition are handled by creating virtual walls to the right of the red flags and left of the green flags, after segmentation is preformed. Once an image has been found with all of the obstacles marked, a homography transform is performed on the image. The homography 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 homography image and ray-cast are published. The various stages of the image pipeline may be seen in *Figure 8*.

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 odometry. 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 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 chose the closest. The user may set time limits on each priority to be sure the robot has enough time to finish the course. Position outputs the combined position of the robot as well as the position of the robot's next target.

5.2 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 Simulations Location and Mapping (SLAM) 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 into the global map by using the combined position, and by tracking features from frame to frame. The software uses the tracked 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. A map created of the team's lab using gmapping may be seen in *Figure 9*.

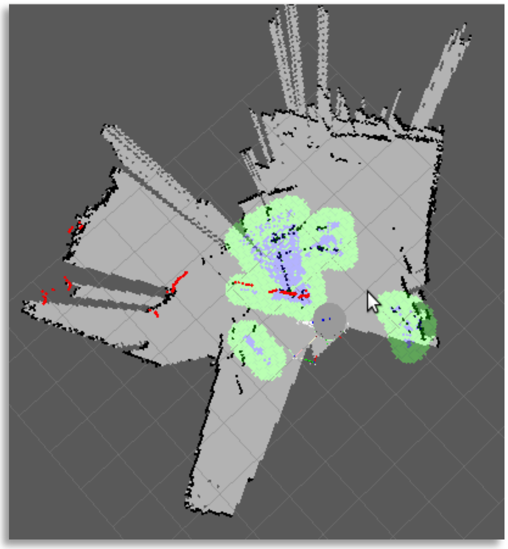


Figure 9: Map of the Team's Lab

5.3 NAVIGATION

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

- ***ROS NAVIGATION STACK***

Jomegatron'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.

- ### POTENTIAL NAVIGATION

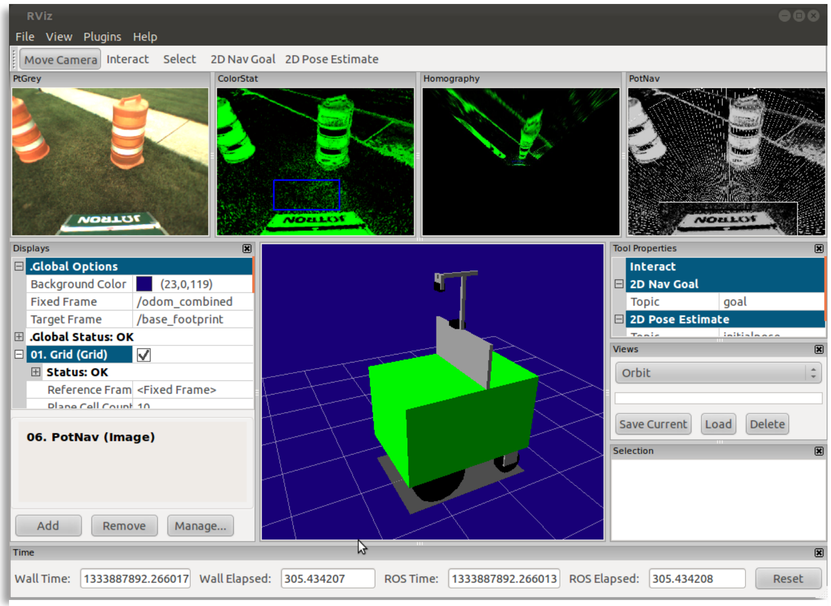
The potential navigation method was designed to mimic a gravity model with obstacles repelling the robot and waypoints attracting it. The software uses the local model to sum all of the objects repelling the robot. Obstacles that are closer to the robot repel it exponentially more than more distant obstacles. The software then gives the robot a constant forward attraction and an attraction to the target waypoint. The module has a large number of parameters that can be configured during operation, allowing the user to tune the performance.

5.4 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.

5.5 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 connected or for the JAUS node to take control. Once the user connects he/she has 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.



.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

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

Figure 10: RViz Robot Dashboard

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 10*. The display is customizable at runtime 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.

5.6 HARDWARE INTERFACE

The hardware interface node coverts the desired robot velocities output by control 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 wheel odometry. The software also interfaces with the E-stop board giving the state of the robot and control over the safety light.

6.0 SYSTEM INTEGRATION PLAN

Jomegatron's control software was developed by several student members of the team all working on different modules and levels. By using ROS the software was able to be developed in parallel. After the interfaces were defined the internals of individual modules were coded by small teams.

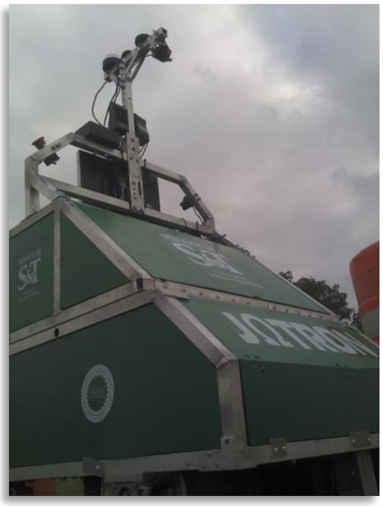


Figure 11: Field Testing

Testing was performed at both the module level and at the system level. 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.

Team members that participated in past IGVC events were 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, slalom flags, 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.

7.0 PERFORMANCE EXPECTATIONS

The whole team feels very confident in Jomegatron's ability to compete in the 2012 IGVC. The robustness of its algorithms have been proven time and time again in simulations. The additions to the software have made the robot much more reliable. The team's predictions, along with the design's demonstrated values, may be found bellow in *Table 1*. The team has taken steps to ensure that spare parts for each of the robot's components will be available at the competition. The Missouri S&T Robotics Team expects Jomegatron to achieve the university's best showing ever in the IGVC and to finish among the top five 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	• Web Cameras : ○ 4.5 M ○ 3.2 M ○ 5.52 M
Accuracy of arrival at way points	2 M	1.5 M

Table 1: Performance Comparison Table

7.1 COMPLEX OBSTACLE HANDLING

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

- ### SWITCHBACKS

When a switchback situation is encountered, Jomegatron will seek the path of least resistance. When no such path is obvious the robot will drift along the closest edge until a clear path is found.

- ### CENTER ISLANDS

Jomegatron 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

Jomegatron retains a global map of obstacles it has encountered on the field. If Jomegatron encounters a dead end, it will return to the last fork it encountered and attempt a new path.

- ### TRAPS

To negotiate traps, Jomegatron employs a method similar to that used to detect and navigate out of dead ends.

- ### POTHOLES

Jomegatron will avoid all potholes provided they are a sufficiently different color than the grass.

- ### DASHED LANE LINES

Jomegatron's navigation will tend to take the path of least resistance and the robot tends to want to move forward. The spaces between the dashed lines of the course generally have a higher resistance than the correct path. As a result the robot tends to veer away from dashed lines.

- ### SLALOM FLAGS

Slalom flags are detected as special obstacles by the vision software. Virtual walls are created to the right of red flags and to the left of green flags. The walls prevent the robot from traversing the wrong side of flags. Once the virtual walls are added the flags are treated as any other obstacle.

8.0 SAFETY

With all of the power Jomegatron can supply, there needs to be some way to stop the robot in case the operator loses control. Jomegatron is equipped with a triple redundant E-stop. 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 robot moving. If this watch-dog requirement is not met, the robot will stop moving. This prevents the robot from running off in the event of a computer glitch or total computer failure. There is also a remote control so that the robot may be stopped from afar if necessary. In the event of a power failure the robot is automatically stopped.

By controlling Jomegatron with the Wii gaming console's Wiimote controller, the team has a large amount of control over the robot. 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. Jomegatron'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. If the Motors module has not received a request in the last second, it will safely stop and turn off the motors. This prevents a single module crash from causing a runaway situation.

9.0 COST IN DOLLARS/HOURS

Jomegatron has been in development for 2 years. The total accumulated financial cost of Jomegatron to the team for those years comes to just above $7,700. A breakdown of costs may be found below in *Table2.*

Component	Cost to Team	Retail Value
Frame	$500	$500
Motors	$0	$1,300
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
Computer	$800	$800
Bluetooth Controler	$0	$65
Microbotics INS/GPS	$0	$5,710
Totals	***$7,730***	***$14,805***

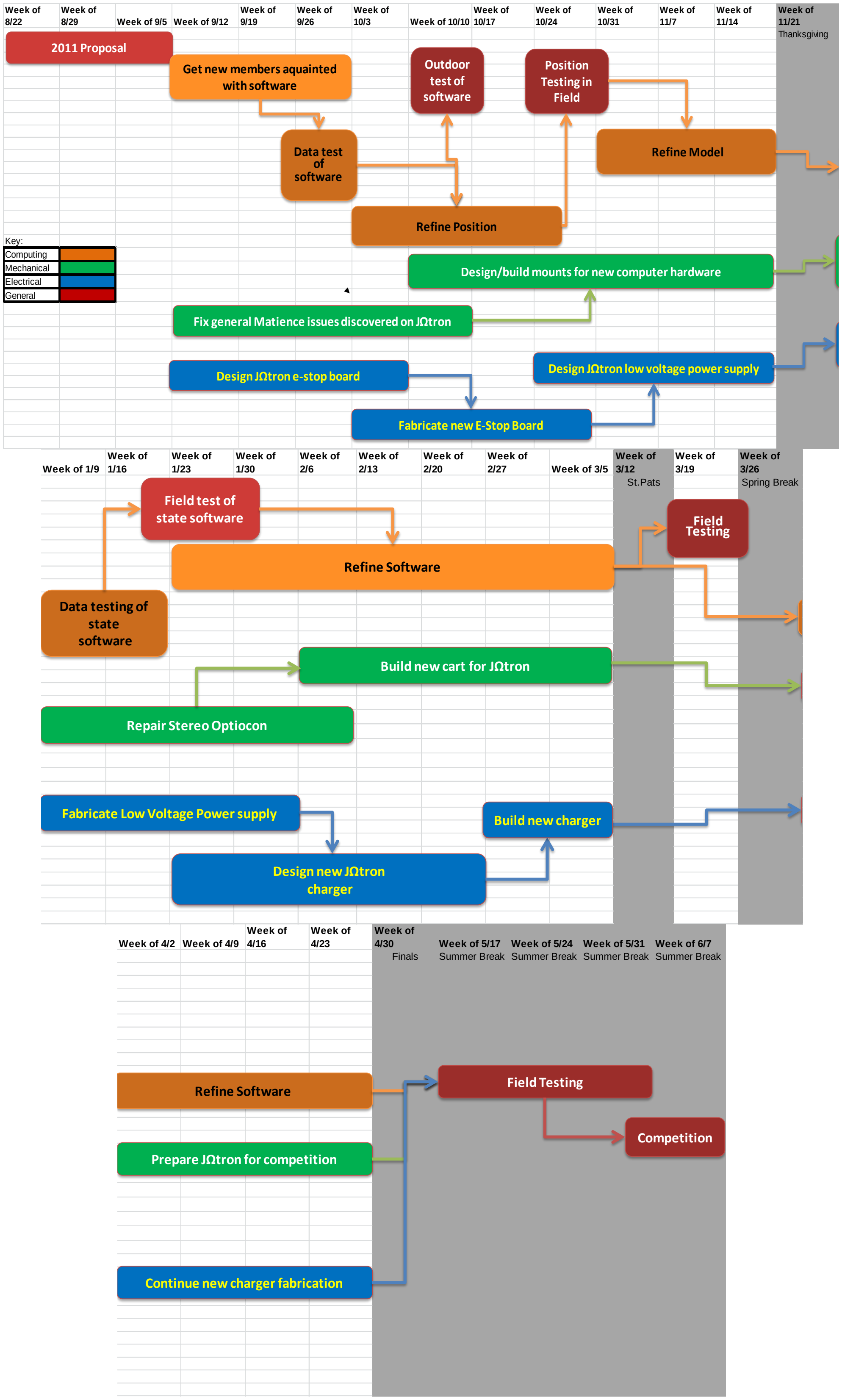
Table 2: Cost Analysis

Jomegatron'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 Jomegatron for this competition. Over the past two and a half years of design and build, this amounts to a conservative estimate of 2,000 man hours. The team is very proud of the robot it has created and expects to have a very successful run at competition.

APPENDIX A: MEMBER LIST

Name	Level	Discipline
Anderson, James (Computing Division Lead)-Appointed	Senior	Computer Engineering
Anderson, Miriah (PR Officer)-Elected	Senior	Mechanical Engineering
Baleta, Joseph (Vice President)-Elected	Sophomore	Mechanical Engineering
Bertels, Jacob	Sophomore	Computer Engineering
Bessent, Chris	Senior	Computer Engineering
Boyko, Ken	Graduate	Computer Science
Chrisco, Michael	Senior	Electrical Engineering
Jason Gassel	Sophomore	Computer Science
Honse, Adam(Electrical Division Lead)-Appointed	Senior	Computer Engineering
Marik, Nick (Treasurer)-Elected	Sophomore	Mechanical Engineering
Painter, Chris (Team President)-Elected	Junior	Mechanical Engineering
Peterson, Daniel (Electrical Division Lead)-Appointed	Junior	Electrical Engineering
Reidmeyer, Andrew	Sophomore	Computer Engineering
Siebert, Christopher (Secretary)-Elected	Sophomore	Computer Science
Uhlman, David(Mechanical Division Lead)-Appointed	Senior	Mechanical Engineering

APPENDIX B: 2012 SCHEDULE


Week of 8/22
Week of 8/29
Week of 9/5
Week of 9/12
Week of 9/19
Week of 9/26
Week of 10/3
Week of 10/10
Week of 10/17
Week of 10/24
Week of 10/31
Week of 11/7
Week of 11/14
Week of 11/21
Thanksgiving
2011 Proposal
Get new members aquainted with software
Outdoor test of software
Position Testing in Field
Data test of software
Refine Model
Refine Position
Key:
Computing 🟧
Mechanical 🟩
Electrical 🟦
General 🟥
Design/build mounts for new computer hardware
Fix general Matience issues discovered on JΩtron
Design JΩtron e-stop board
Design JΩtron low voltage power supply
Fabricate new E-Stop Board
Week of 1/9
Week of 1/16
Week of 1/23
Week of 1/30
Week of 2/6
Week of 2/13
Week of 2/20
Week of 2/27
Week of 3/5
Week of 3/12
St.Pats
Week of 3/19
Week of 3/26
Spring Break
Field test of state software
Field Testing
Refine Software
Data testing of state software
Build new cart for JΩtron
Repair Stereo Optiocon
Fabricate Low Voltage Power supply
Build new charger
Design new JΩtron charger
Week of 4/2
Week of 4/9
Week of 4/16
Week of 4/23
Week of 4/30
Finals
Week of 5/17
Summer Break
Week of 5/24
Summer Break
Week of 5/31
Summer Break
Week of 6/7
Summer Break
Refine Software
Field Testing
Competition
Prepare JΩtron for competition
Continue new charger fabrication