ROBOTICS COMPETITION TEAM

"Helping Robots Help Themselves"

The Missouri S&T Robotics Competition Team is proud to present its entry for consideration in the 2009 Intelligent Ground Vehicle Competition:

ALUMINATOR

Introduction

The Missouri University of Science and Technology (Missouri S&T) Robotics Competition Team is proud to submit "Aluminator" to the 2009 Intelligent Ground Vehicle Competition (IGVC). This will be the third year that Aluminator has competed in the IGVC and the sixth year the Missouri S&T has submitted an entry. While originally designed as a reliable and expandable test bed for software, Aluminator has grown into the leading robot of the team. The team has come to realize that Aluminator's sturdy and simple frame is ideal for the challenges presented by the IGVC. This realization has come as part of a philosophical shift in the team. The guiding principles of the team for the 2009 season have been flexibility and robustness. The software and electrical systems have all been rebuilt around this vision.

Vehicle Highlights/Innovations

On the surface the 2009 Aluminator may look similar to the 2008 model. However, under the hood there have been several significant changes. Each change is derived directly from our new philosophy of flexibility and robustness, and many are direct responses to past problems encountered.

The first change is the motor controllers. During the 2008 IGVC, the Roboteq motor controller that controlled all of Aluminator's movements shorted out the day before competition. This failure caused Aluminator to lose all position feedback data which was catastrophic in the team's bid to compete.

In an effort to prevent a similar failure from occurring again, the team has replaced Aluminator's Roboteq controller with a more robust Elmo Motion Control system. The Elmo system uses isolated controllers that are more durable and can easily be switched out with spares that the team has procured. While learning to utilize a new motor controller system has presented numerous challenges, the team has dubbed the change a worthwhile investment in both the 2009 and future competitions.

The motor controller failure at the 2008 IGVC revealed the importance of a robust position system. At the time the team was only using dead-reckoning to determine the position of the robot. This year a Microbotics INS system was sought out and donated to the team. The Microbotics MIDG II provides position data based upon GPS, 3 axis accelerometer, 3 axis gyro, and 3 axis magnetometer data, processed by an on-board Kalman Filter at 50 Hz. This data provides a redundancy for the wheel encoder position data and allows the software to compensate for possible failures in either system with data from the other.

The software systems of Aluminator have changed the most as a result of the team's new philosophy. A new architecture was designed to be completely modular. This change addresses inefficiencies in previous years that required the rewriting of as much as ninety percent of the software. New software was written in abstracted modules using Boost.MPI. Boost.MPI requires modules to communicate through a standard networking protocol which in turn forces modules to communicate in a well-structured way.

This means that new modules can be written and easily swapped with their older counterparts in the future without serious, if any, modification to other modules. Each module is forced to take on an extra level of abstraction that makes the overall architecture more hierarchical and organized. An added benefit is that these modules can be run on separate computers, vastly increasing Aluminator's computing power without investing in a single powerful computer. If one of Aluminator's computers were to fail, all modules could be moved to the remaining, working computers, adding to the robustness of the system.

Finally, the model for Aluminator was completely rewritten. The model stores all of the data that the sensors take in so that it can be used for later decision making. In the past, the model had been the greatest offender of the isolation scheme. Falling in line with the new philosophy of flexibility the model was isolated from other modules using the Boost.MPI networking protocol. The foundations of the model were also changed. In the past the data stored in the model was stored based on statistical confidence. The more times a location in the real world was observed as having a fixed value, the more confident the model was that it could assign it that value; unfortunately, this lead to inflexibility. After a location had been observed as being an obstacle for a large number of updates, it would require exactly that many updates as being an open space for the model to go back to being neutral. In order to give priority to the more recent information, a decay system was substituted for direct confidence. In this system the incoming data is averaged with all past data at a given ratio. The older the information in the model is, the less impact it has on the current state.

Team Structure

The team has five elected officers who in turn appoint three additional officers. The team president, vice president, treasurer, public relations officer, and secretary comprise the five elected positions. Three division leaders are elected by the team-elected officers. Each oversees and manages one of the team's three divisions; mechanical, electrical, and computing. All other team members are a part of one or more of these divisions.

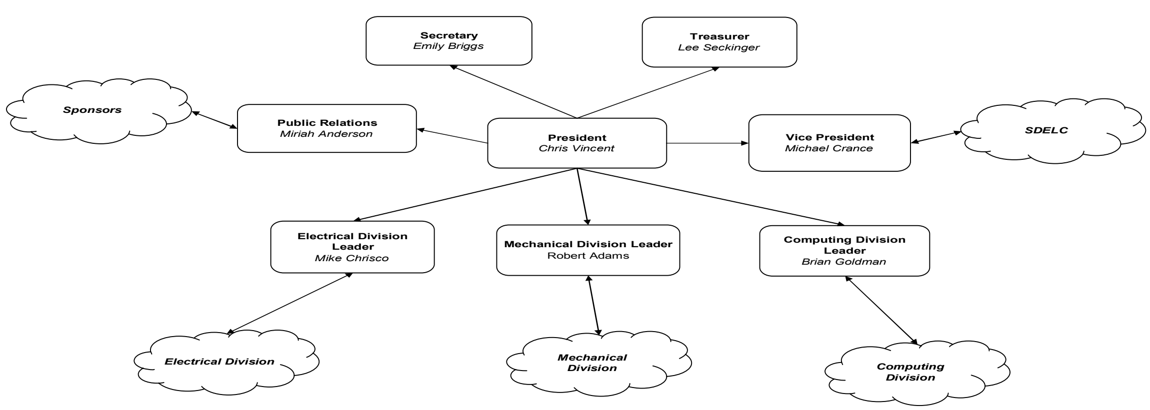


Figure A: Team Structure

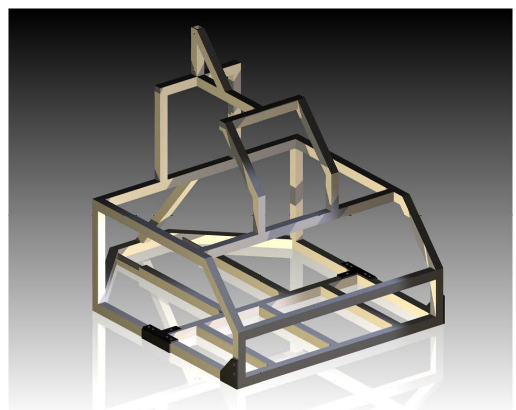
Planning

At the beginning of the 2009 IGVC construction year, the team's efforts were split between revamping Aluminator and a new robot dubbed Project X that was to be constructed in the spring of '09 and compete in the 2010 IGVC. However, as a result of a midterm review, the team decided it would be more beneficial to concentrate all remaining efforts on Aluminator and the 2009 IGVC. The construction of Project X will likely recommence in some form in the fall of 2009. The key goals for the 2009 IGVC were implementation of the new Elmo controllers, restructuring and rewriting of the software architecture, implementation of the Microbotics INS, and design of Project X.

The final revised schedule with focus on objective goals and specific aspects of the IGVC challenge can be found *Appendix A.*

The Platform

Frame



Aluminator's frame is welded one inch square aluminum tubing. Welding of the joints allows the robot to have a much more ridged and reliable frame than a bolted configuration or an 80/20 (slotted bar) system. The robot is 28 inches wide by 36 inches long, meaning that Aluminator will fit through a standard door frame. This width was chosen intentionally to facilitate indoor testing and travel to public relations events. The length was selected for maneuverability and stability. This year the team has added an array of webcams to the top of the mast as secondary sensors which has increased the overall height of the robot from 52 inches to 53 inches. The sensors are mounted on a mast situated near the rear of the robot so as to allow the main camera to have a better field of view in the immediate vicinity of the robot. All of the vision sensors are fully adjustable with respect to their vertical distance to the ground and their angle to vertical and the angle of the webcams on their vertical axis is also adjustable.

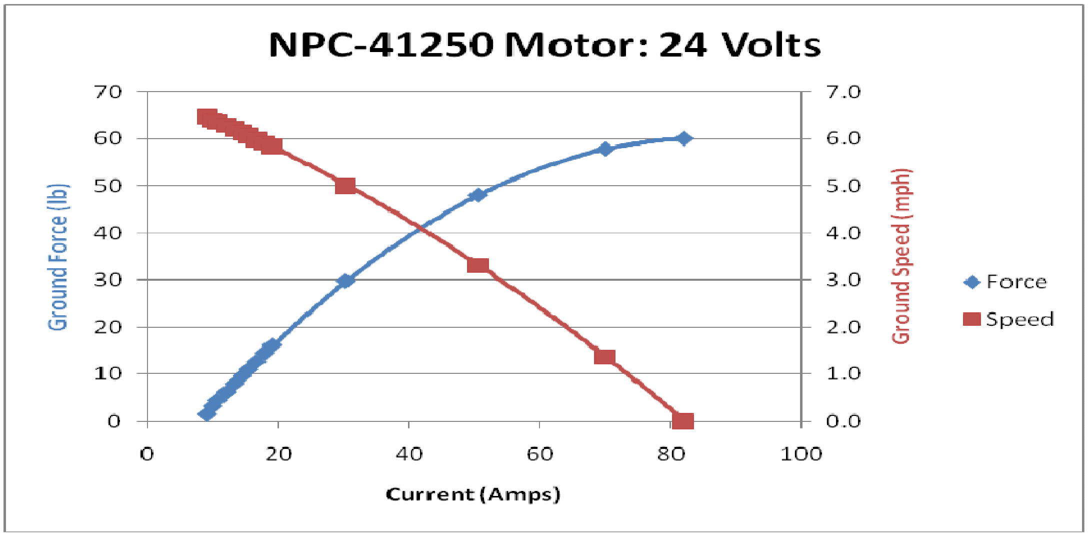
The total weight of the robot is 144 pounds with 48 pounds of that being the batteries. In the 2009 version of Aluminator two hinged doors were added to the shell. These permit access to the internal components without taking off some of the shell panels--a task that can consume several minutes.

Drive and Power Systems

The drive train of Aluminator was designed to be as simple as possible, while being able to perform well in all situations presented in the IGVC. Two fixed drive wheels are in the front of the robot with a caster in back. This allows the robot to make zero-radius turns around its drive axis. This turn radius is not only important for making the tight turns presented in the IGVC, but also for being able to turn and view the area around the robot while in one position. Another important aspect to the wheel setup is that none of the wheels slip during operation with normal motor torque. Not only does this require less power than a tracked or skid steer setup, but it also allows for dead reckoning to be used to keep track of position. The drive wheels were put in the front to allow the robot to pull the caster through terrain instead of pushing it, which is an important factor in off-road performance. Also, the drive wheels are from a snow-blower, so they will be able to maintain traction in the surfaces presented in the IGVC including the sand pit.

Different weight distributions are possible for Aluminator because the primary factor determining center of gravity is battery position. Naturally, the robot was balanced left to right, so the batteries were placed on the midline. There were two primary factors governing the battery placement: tipping and weight on the front wheels. To prevent tipping, the batteries should be placed near the center of the robot, while to increase maximum torque of the front wheels, the batteries should be placed farther forward. Both of these positions were tested, and tipping was a problem when the robot stopped while moving down a steep slope. Therefore the batteries were placed in the middle, of the robot, giving the front wheels a maximum ground force of about 2/3 the weight of the robot.

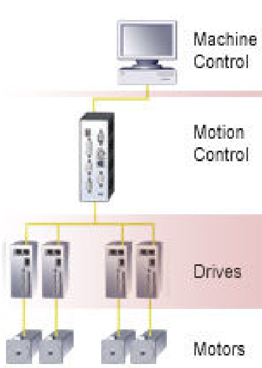
The robot is powered by two twelve volt batteries wired in series to give 24 volts to Aluminator's systems. The motors are rated for 24 volts, and they consume most of the robot's power, so having them at the same voltage prevented the need for a high amperage voltage adapter. For the drive motors on the robot, two NPC-41250 wheelchair motors were used. The performance data for the motors are shown in the graph and table below. The motors were tested by NPC with a dynamometer from ten to twenty amps, and at stall. Each motor can output 30 lbs of ground force while still maintaining a 5 mph ground speed. Also, the calculations were performed to determine the force required for the motors to climb a 6 inch vertical obstacle, which was 29 lbs per motor, requiring 30 Amps. However, in order to pull the 6" caster over an obstacle, it must either be shorter than 3 inches or have a maximum slope less than 60°. The maximum force that the motors can apply without slipping on a high friction surface is about 48 lbs per motor. This is shown below and requires 51 amps with a maximum speed of 3.3 mph. At this torque, the robot is able to climb a maximum slope grade of 90% at 42°. When an extra load is applied, and the force on the front wheels is increased, the maximum ground force per motor may exceed 58 lbs. In this case, the motor controllers can supply up to 70 amps resulting in a maximum motor ground force of 58 lbs at 1.4 mph.



Input Current (Amps)	Tourque (in-lb)	**Ground Force (lb)**	Speed (RPM)	**Ground Speed (mph)**	Notes
9	10	**2**	174	**6.5**	Dynamo
10	20	**3**	172	**6.4**	Dynamo
11	28	**4**	171	**6.4**	Dynamo
12	39	**6**	169	**6.3**	Dynamo
13	50	**8**	167	**6.2**	Dynamo
15	60	**10**	165	**6.1**	Dynamo
16	69	**11**	163	**6.1**	Dynamo
17	79	**13**	161	**6.0**	Dynamo
18	91	**14**	159	**5.9**	Dynamo
19	101	**16**	157	**5.8**	Dynamo
30	186	**30**	134	**5.0**	5 mph / 6" Rut Force
51	300	**48**	89	**3.3**	Max Slip Torque
70	361	**58**	37	**1.4**	Max Controller Amps
82	375	**60**	0	**0.0**	Stall

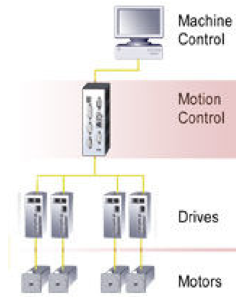
Elmo Motion Control System

The most significant problem encountered at the 2008 IGVC was the failure of the robot's Roboteq motor controller. To prevent that scenario from occurring again, Aluminator was upgraded to Elmo Motion Control motor controllers. The Elmo system is based on a modular component system. Each motor is individually controlled by a Drum controller.



The Drums can supply 70 amps RMS continuous and 140 amps peak at up to 110 volts. Because Aluminator's motors are only rated up to 24 V, this is the maximum voltage supplied to the motors. Therefore, because only a one quarter of the motor controller's potential voltage output is utilized, the high amperage is important for the robot to get enough low end torque. This controller communicates with differential encoders attached to the wheels and the Maestro.

The robot has one Maestro unit that is connected to both Drums. The Maestro is in turn connected to Aluminator's drive computer. With some configuration performed by the team, the Maestro and Drums completely abstract the motor control. The team also has spares of both units that could be quickly configured to replace any of the existing units in the event of a hardware failure.



Hardware

Stereo Vision Cameras

Aluminator uses a set of Videre Stereo Vision cameras as its primary obstacle and path detection sensors. These cameras provide the power of on-board computing along with a plethora of raw data for use in our control algorithms. The cameras have a sixty degree field of view with a distance of up to thirty feet. Data is available, with on-board computing, at a rate of sixty frames per second. This year, a lot of work was put into development of new calibration software and techniques. Although this software is not run during normal operation, its functionality is invaluable for calibrating the cameras in order to get the most accurate information possible from the cameras.

Camera calibration is performed to detect and remove systematic errors in interior orientation (camera lens/sensor geometry and lens distortion), exterior orientation (perspective errors due to angle mounting of camera), and spectral characteristics (tuning cameras to recognize several specific colors). Interior calibration is performed by imaging a calibration target from several angular orientations and computing intrinsic correction coefficients to remove both radial and tangential lens distortion. Exterior calibration is performed by imaging several 2-D targets at known 3-D ground positions. The result is an 9-parameter transformation matrix which (for the stereo data) transforms the rotated (x,y,z) point cloud to level ground space. For the 2-D web cams, a similar 9-parameter transformation is used to remove perspective errors, transforming raw image coordinates to orthogonal Cartesian ground coordinates. The spectral (color) calibration is performed by sampling test targets, and adjusting the thresholds for hue / saturation ratios. The color tuning routine can be rapidly refined to utilize real-world observations at the competition.

Videre Stereo Vision cameras were chosen over other sensors, such as laser range-finding systems, due to their lower cost and rich data. The trade-off is the workload on the computing hardware and more custom software necessary. Software had to be written to extract useful meaning from the images. Every image has to be transformed several times into different coordinate systems and run through a myriad of filters to detect colors, slopes, and screen out noise.

Webcams

Aluminator has added two Logitech QuickCam® Pro for Notebooks web cameras to its sensor arsenal. These cameras combine a 2 megapixel native sensor at thirty frames per second with autofocus lenses and sixty degrees of viewing angle. The simple USB 2.0 interface made their images easy to incorporate with the Videre data to give Aluminator a wider range of vision at very little cost.

Computers

A key advancement in the 2009 version of Aluminator is a software architecture designed around clearly defined, networked modules. This design allows Aluminator to run its necessary processes on three separate computers in parallel, drastically increasing its computing power. The modules are all run as separate processes and communicate with each other over the local area network (LAN). The computers Aluminator uses are Intel Atom 260 processors on iBase MI800F micro-ITX motherboards. These boards are industrial grade and have been fitted with flash memory and a Firewire card in order to incorporate Aluminator's stereo cameras. They communicate with each other via gigabit Ethernet ports. This approach to Aluminator's computing hardware is very redundant and flexible.

The general approach for the software this year was to keep everything in well defined modules. This decision was based upon the repeated problems the team has had using previous year's software. By switching to a modular approach, a select number of modules can be rewritten each year and the remaining modules can be retained. This eliminates the confusion caused when members who designed software are no longer available to the team. Each module was carefully designed this year to have very specific and well defined dependencies.

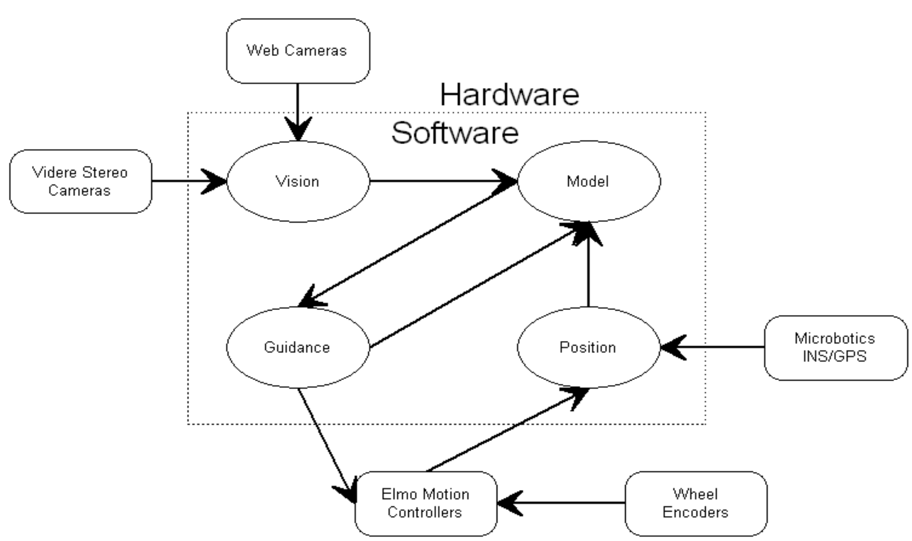


Figure B. Software Architecture

With this architecture and the distribution of processes to separate machines, a network communication protocol was required. BoostMPI was chosen for this purpose. BoostMPI is an implementation of Message Passing Interface (MPI) libraries for python and C++. These libraries allow all of Aluminator's software modules to package different data types and send them to each other.

Software

Vision

Owing to the fact that the robot's main obstacle and path detection sensor is its stereo vision cameras, vision computing entails a large part of the team's efforts. There are two main tasks that the vision module must accomplish: color filtering and slope filtering.

The on-board processor of the Videre cameras streams the left lens color image for our software's use. The team has taken two approaches towards color filtering. The more traditional filtering converts this image to hue-saturation-value (hsv) format. Colors are then separated into different basic thresholds (predefined, but easily adjustable). This allows the robot's vision to be adjusted for different lighting environments. For competition, the software keeps track of orange, white, and green. These colors are arranged into a hierarchy. Orange is above green, and green is above white. This accomplishes the same task as subtracting the colors, preventing a single pixel from being designated both green and white. Placing white at the bottom of the hierarchy helps reduce the noise produced by glare from the sun and other light sources. These colors are stored in different channels of an OpenCV image. However, the system is built in such a way as to allow easy expansion to as many different colors as are required.

To discriminate between several different colors, the ratio of hue to saturation is computed and subjected to a variable binary threshold. This method has provided the best overall color differentiation, while also being relatively invariant to varying lighting conditions. The method can be fine-tuned for lighting conditions immediately prior to the competition by adjusting sensitive sliders on the display which set the binary thresholds for various colors. A low-pass pre-filter acts to eliminate noise, and produces a high-fidelity binary image of chalk-line boundaries and barrel / fence obstacles.

The team also tried red-green-blue (rgb) images to do color filtering. This technique is rarely used in these kinds of applications, because predefined thresholds often degrade in dimmer/brighter lighting conditions. However, the team has solved this problem by defining thresholds as a ratio of the red, green, or blue values to the sum of the three values. This approach is similar to the hue value of an hsv image, but the team has found slightly different behaviors with this approach and is thus still experimenting, although the hsv approach appears to be superior.

The Videre Stereo Vision cameras provide the robot with a disparity map of the environment. The difficulty arises in trying to make sense of this data and then transform it into an image that can be used. The first transform that must be done is to transform this skewed data into a map in real world coordinates. This corrects for the tilt of the camera and the vanishing point effect. This transformation matrix was derived through calibration of the camera with known points in a controlled environment. The next transform centers this map from the camera to a point on the center of the robot's axle in a bird's eye view. This makes moving the map into global coordinates much easier, since position is tracked from the rotating center of the robot. In previous years, this was where processing stopped. However this year, the team has gone a step further to reduce error in the disparity data which arises due to the nature of the stereo cameras. The disparity data of the cameras drops off in resolution with greater distances. This causes a curtain effect where data past a certain distance ceases to have intelligible meaning. To correct for this, the map is transformed back into a perspective view, as if the disparity data were just an optical picture of the scene. These images can then be processed in the same way as color images to determine distance from the robot. The perspective view is also a much more intuitive way to store the data making a multitude of tasks easier, especially debugging. The disparity data is used to find heights at these locations, and anything over a certain height is mapped as high cost. Finally, all of these transform matrices are combined into one matrix, requiring only one pass by the processor and speeding up the process by a factor of three.

Due to noise induced in the cameras, both of these data sets (the color and slope images) are processed by a blob filter. While the basic algorithm for a blob filter is fairly simple, every image taken by the cameras must go through this filter and resource consumption becomes a major issue. To counteract this, special attention was given to processing speed when writing the blob filter. The filter was written completely in C which is slightly faster than C++. Blobs were kept track of using a mapping table that records which blob a pixel is part of. If it is later found that two blobs are actually one larger blob, the image is not reprocessed, the table is merely updated to reflect the fact that those two blobs are not equal. All loops in the code were also unrolled. This consists of implementing part or all of the loop by hand. This was done since the edges of images need to be treated differently that the interior. Instead of one large loop with a check to see where the pixel is in the image, these steps were broken up. The result is that the code reads longer, but executes faster. Finally, various steps were taken involving the way the data was processed by the hardware, such as reading the image in such a way as to utilize the full capacity of the cache. The result of these steps was extraordinary. A basic implementation of the blob filter was able to get seven to ten frames per second. After the optimization, the blob filter consistently operated at a minimum of fifty frames per second on a 640 x 480 image.

Webcams

The addition of inexpensive web cameras to Aluminator is new in 2009. Aluminator utilizes two Logitech notebook cameras with a VGA resolution of 640 x 480. Each camera, as placed on the mast of the robot, provides approximately a 60 degree field of view (comparable to Aluminator's stereo cameras). The web cameras have been positioned in such as way as to minimize the overlap of these three fields of view. Images captured by the web cameras are transformed using a lookup table generated through calibration. This simple transformation places all of the data effeciently in the ground plane. With these images added to those of the stereo camera, Aluminator is able to have a 180 degree view of the ground surrounding it out to around three meters.

Position

Another new advancement to Aluminator in 2009 is the addition of a Microbotics MIDG II INS system. This device houses a three axis gyro, a three axis magnetometer, a three axis accelerometer, and a GPS antenna. In addition, the unit does real time processing on-board, pushing all GPS and IMU reading through a Kalman filter whenever GPS data is available. The team has been able to use this data, along with dead-reckoning from the encoders, to acheive a much more accurate position module. Other modules (primarily model) are able to request this data in order to correlate the orientation of new data with that of older data.

Model

All of the images sent to the model are already in a log-polar format. This format was chosen since the vision has more accurate information about areas close to the robot than further away. In a log-polar format, the pixel resolution represents this accuracy. The model also receives information on the position of the images it is sent relative to the robot as well as the time at which the data was collected. The model then looks up the position of Aluminator at the time the data was collected, and uses all of the information it has to turn the image into a rectangular bird's-eye-view and place it in the model at the appropriate global position. This image is summed with previous images using weights in order to decay old information.

The model provides information to other modules in two different formats: a trace function and a raw local image. Modules may request a trace value by providing an angle relative to the heading of the robot. The model will return the distance to the closest obstacle in that direction. If a module requests raw data, the model will send an image in rectangular format of the space directly around the robot in a coordinate system based upon position of the robot regardless of the global coordinate system.

The model also incorporates the confidence of past data. The more an obstacle is seen, the surer the model is that it is indeed and obstacle and not just noise or a faulty reading. However, this can run into problems when the most current information is ignored in favor of older data. To compensate for this, the model also incorporates a decay function. The cost value of each pixel is determined by Equation 1.

$$\text{Cost} = B * \alpha + M * (1-\alpha) \qquad (1)$$

Where: B = new information

M = current model

α = decay rate

Guidance

The path planning algorithm is one of the few pieces of software that has been carried over from the 2008 IGVC. Due to the hardware failure in 2008, this algorithm has yet to be used in competition. Aluminator uses an algorithm called CircleAI developed by the team. The algorithm draws a circle in front of the robot's position and then the circle is expanded until it touches the edge of an obstacle. The circle is then turned away from the obstacle and continues to increase in size until it encounters obstacles on either side of its center. The robot is then aimed at the center point of this circle. The algorithm has several safeguards that prevent the circle from growing too large or small and to limit the turning angle of the robot. Testing in Player/Stage* has demonstrated that this is a very robust algorithm given fairly clean data--the subject of most of this year's improvements. The circle is able to fill in dotted lines and avoid entrapment in corners.

Another guidance algorithm has been developed just this year. This algorithm is based on ray tracing. The program uses a three hundred sixty degree ray trace to determine the closest obstacle at each heading. Then, lines tangent to each ray are drawn of a length equal to half of Aluminator's width, plus a small buffer first in the clockwise direction and then in the counterclockwise direction. Any rays where these tangents intersect obstacles are eliminated as possible paths. The remaining rays are then weighted by their length and the cosine of half of their relative headings. This causes the algorithm to try to move in a straight line and dividing the angle by two allows the heading opposite of the current motion to still carry some weight in case the robot becomes trapped in a corner. The speed is then set as a function of the turning speed and the square of the distance to the nearest obstacle. This algorithm has also passed extensive testing in Player/Stage.

**Note: Player/Stage is an open source, Linux based simulation package that the team has used for two years to test guidance algorithms. The package consists of a two dimensional environment simulator, Stage, and a robot simulator, Player.*

Navigation v. Autonomous Challenges

Because of the way that the software for Aluminator has been written, there is only one difference between the Navigation and Autonomous Challenges: the goal. For the Autonomous Challenge, the guidance algorithm begins by trying to travel straight forward. The heading is then adjusted based on the obstacles seen. For the Navigation Challenge, the guidance begins by attempting to travel directly toward the goal point (within certain turn rate parameters). The heading is again adjusted based upon obstacles. In the CircleAI the size of the circle drawn is limited to some maximum and the heading no longer becomes a bisector of the two closest obstacles. In the other guidance algorithm, the possible paths are merely weighted with a bias towards the GPS goal. *(See Guidance)*

Safety

Aluminator is equipped with three emergency stops: one hard and two soft. The hard stop actually consists of two different push buttons located on the top rear of the robot. These are open on (meaning any break in the circuit will automatically trigger the stops, even if it's siimply a wiring fault) stops, that are connected directly to Aluminator's motor controllers. Aluminator may also be stopped by a switch on our RC remote or through the use of the drive controls from a wireless laptop.

Aluminator's speed has been limited in the hardware to just under five miles per hour and the fuses installed on the motors ensure that they receive no more than eighty amps. Also, the robot weighs just under 150 lbs, and can be stopped if something were to fail.

Cost in Dollars/Hours

Aluminator has been in development for three years now. The total dollars costs for Aluminator in those years is shown below in *Table 1.*

Component	Cost to Team	Retail Value
Frame	$200	$200
Motors	$300	$300
Shell	$250	$250
Wheels	$140	$140
Misc. Hardware	$100	$100
Misc. Electrical	$200	$200
Elmo Motor Controllers	$2,958	$5,690
Encoders	$170	$170
Batteries	$0	$336
Videre Stereo Camera	$1,450	$1,450
Computers	$975	$975
Radio Control	$200	$200
Web Cameras	$200	$200
Microbotics INS/GPS	$0	$5,710
Wireless Router	$60	$60
Totals	***$7,203***	***$15,981***

Table 1: Dollar Cost of Aluminator

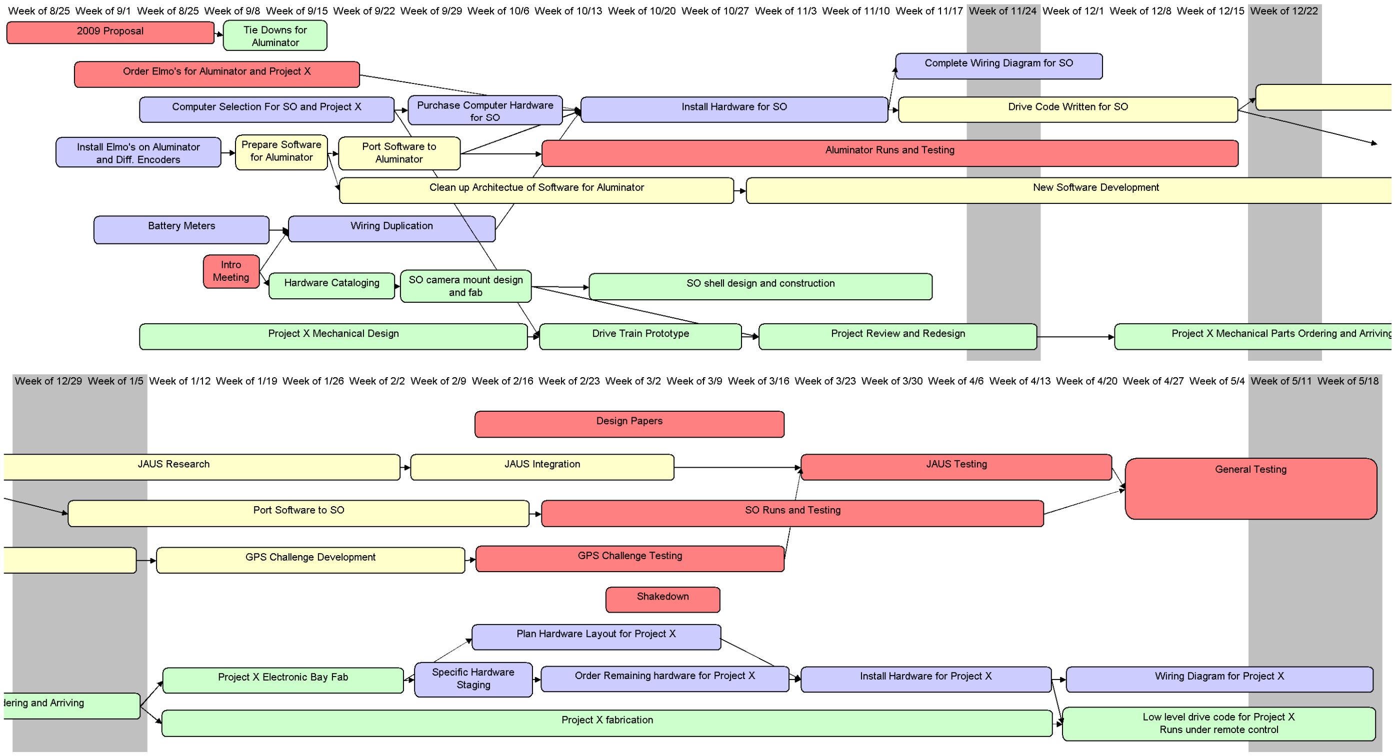
The construction and programming for Aluminator has also required a large amount of man hours. On the whole, the team has spent an average of forty one hours a week preparing Aluminator for this competition. Over the past two years, this amounts to two thousand nine hundred fifty-two hours. This is a conservative estimate.

Performance Expectations

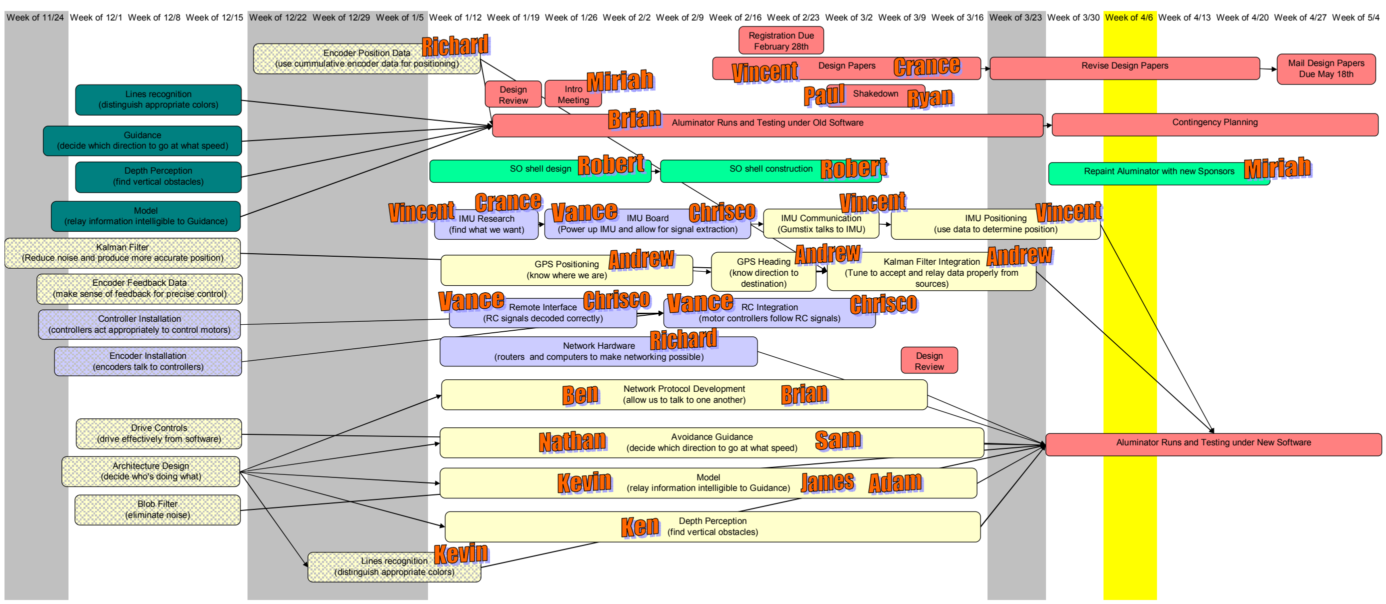
On the whole, the team feels very confident in the 2009 version of Aluminator and its superiority to previous versions. The robustness of its algorithms has been proven time and again in simulation. The new Elmo motor controllers are much more reliable and powerful than the past Roboteq model. Finally, the team has taken steps to ensure that spare parts for almost every component on the robot will be available at the 2009 IGVC, so as to prevent a repeat of the 2008 catastrophe. In short, the Missouri S&T Robotics Team is expecting Aluminator to have the university's best outing ever in the IGVC and finish among the top ten in the field.

Appendix A: 2009 Schedules

Original



Revised



Key:

Computing
Mechanical
Electrical
General

Appendix B. Membership List

Name	Standing	Major
Adam Honse	Freshman	Computer Science
Andrew Heckman	Junior	Computer Science
Benjamin Bethge	Senior	Computer Science
Brian Goldman	Junior	Computer Science
Chris Vincent	Junior	Mechanical Engineering
Cory Marchant	Junior	Mechanical Engineering
David Wehner	Junior	Electrical Engineering
DJ Madison	Junior	Electrical Engineering
Emily Briggs	Sophomore	Mechanical Engineering
Eric Callanan	Freshman	Mechanical Engineering
Erik Scroggs	Freshman	Mechanical Engineering
Jake Meiergerd	Freshman	Computer Science
James Anderson	Sophomore	Computer Science
James Harter	Freshman	Mechanical Engineering
Josh Vance	Sophomore	Electrical Engineering
Justin Priest	Senior	Mechanical Engineering
Ken Boyko	Senior	Computer Science
Kevin Howe	Junior	Computer Science
Leland Seckinger	Junior	Electrical Engineering
Michael Crance	Junior	Mechanical Engineering
Mike Chrisco	Sophomore	Electrical Engineering
Miriah Anderson	Freshman	Mechanical Engineering
Nathan Martin	Sophomore	Computer Science
Paul Robinette	Graduate	Computer Engineering
Richard Allen	Senior	Computer Science
Robert Adams	Senior	Mechanical Engineering
Ryan Meuth	Graduate	Computer Science
Sean McQuay	Freshman	Electrical Engineering