The Virginia Tech Autonomous Vehicle Team presents:

Johnny-5

Andrew Fife, Jason Fountain, Andrew Livick, and Peter King

Required Faculty Advisor Statement

I certify that the engineering design of the vehicle described in this report, Chimera, has been significant, and that each team member has earned six semester hours of senior design credit for their work on this project.

Charles F. Reinholtz, Department of Mechanical Engineering, Virginia Tech

1. Introduction

In the 2004 Intelligent Ground Vehicle Competition Johnny-5 won grand prize honors and in 2005 Johnny-5 placed third overall, with two other Virginia Tech vehicles taking top honors. In 2006 Johnny 5 placed first in the navigation challenge and third in the autonomous challenge. Built and constructed over three years ago Johnny-5 has proven to be a reliable, durable, and functional vehicle for Virginia Tech. This year Virginia Tech is proud to enter a refined version of Johnny-5 in the 15th annual Intelligent Ground Vehicle Competition to compete in the Autonomous Challenge, Navigation Challenge, and Design Challenge.

New for this year's competition, Johnny-5 is equipped with a Kalman filter in conjunction with a new cheaper GPS unit. The base vehicle of Johnny-5 has proven durable and reliable in past years; the one weakness of the design was the gear head shafts which were re-engineered to better handle the demands of Johnny-5. The electrical system was given an update from the old DIN-rail system to a modern printed circuit board.

The vehicle's name originates from the popular 1986 cinema, *Short Circuit*, which depicts a robot imbued with amazing humanistic decision making and control capabilities. Throughout the design process, the name Johnny-5 served as a constant reminder to design an autonomous system that could more closely mimic human intelligence and behavior.

Figure 1.1: The original Cinema version of Johnny-5

1.2 Innovation

Johnny-5 comes to competition this year with several new advancements. First is the new positioning solution. Previously, Johnny-5 received the current position from the expensive ($8000) but accurate Novatel ProPak LB. This year, the team is using a less expensive ($200) but less accurate U-Blox model. Fusing data with other sensors that are already onboard Johnny-5, we are able to get nearly as accurate results at 2.5% of the cost!

In addition to the reworking of the sensor suite, the power conditioning, routing, and protection on Johnny-5 are being replaced with a single printed circuit board. This will reduce the space required inside the frame of Johnny-5 to 16.7% of the original space. The team views this as an opportunity for future expansions and unforeseen uses of Johnny-5. The software is also being updated with the goal of improving performance and increasing autonomy.

2.0 Design Process

The 2007 competition will be the first IGVC for all but one of the Johnny-5 team members. Although this is the first competition for the majority of the team members, the experience from previous team members has proved to be invaluable to the overall design process and implantation of ideas on this year's team.

2.1 Team Organization

The 2007 Johnny-5 team is comprised of one graduate student, Peter King, and three undergraduate students: Andrew Fife, Jason Fountain, and Andrew Livick. Due to the small nature of the team and experience levels, Peter focused on the Kalman filter, and the undergrad students mainly focused on small improvements to fix issues discovered from previous years and in testing this year.

2.2 Target Customers

The following primary customers were identified by the design team: (1) IGVC judges and sponsors who will evaluate vehicle performance, (2) the team faculty advisor who will evaluate overall vehicle design, (3) current and future vehicle users. Secondary customers include team sponsors and the autonomous vehicle community.

2.3 Design Planning Process

A methodical design process is essential for the successful development of complex systems such as Johnny-5. The team used the Kano design method described in *Attractive Quality and Must-Be Quality Method* (Kano, Seraku, Takahashi and Tsuji, ASQC Quality Press, 1996) to guide the design process. Figure 2.3 illustrates this simple common sense approach to the design process.

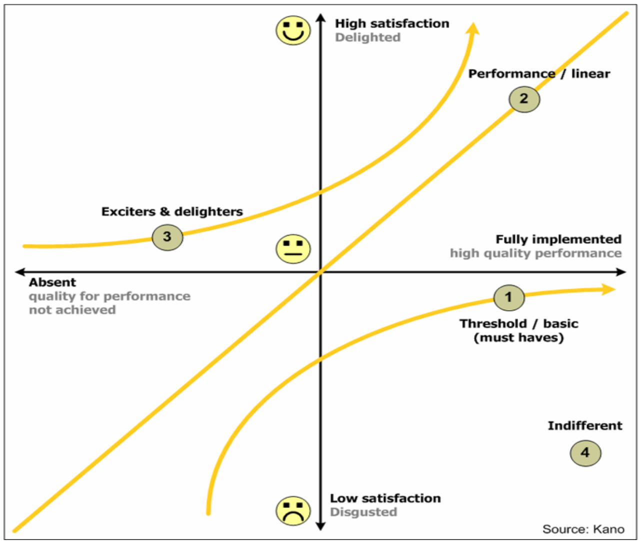


Figure 2.3: Kano design method diagram, image sourced from the Army

For example, for a customer to be fully satisfied a product must first meet the basic “must have” needs. On Johnny-5 these needs include having a sensor suite capable of providing the necessary perception, providing a reliable system for testing and evaluation of software, meeting all safety requirements, and complying with the 5 mph maximum speed limit. The Kano model predicts that customer satisfaction will increase linearly with improvements in performance parameters such as the maximum reliable navigation speed of the vehicle and the continuous run time for testing. Finally, the Kano model suggests that customer satisfaction is strongly enhanced by unexpected features that are not found in competing products, Kano refers to these features as delighters. We believe that the use of the Kano model provided a simple and efficient approach to the redesign of Johnny-5.

2.4 Establishing Target Specifications

Target specifications were established by determining vehicle performance requirements that fulfill customer needs. Performance requirements were determined after reviewing the IGVC rules and the performance of previous IGVC vehicles. Based on this review, the design team determined that implementing a less expensive GPS solution, fixing mechanical failure issues and modernizing the electronics board would be worthy improvements to Johnny-5. All of these changes were implemented while maintaining championship performance in Johnny-5.

3.0 Mechanical Design

The mechanical system of Johnny-5 has proven to be reliable and functional during competition and throughout extensive testing. Johnny-5 has served as a research platform for numerous projects at Virginia Tech. This extensive use and abuse has turned up no mechanical problems in the base chassis of Johnny-5. The drive train as previously mentioned was updated to provide a stronger shaft for the drive train and support of Johnny-5.

3.1 Vehicle Chassis

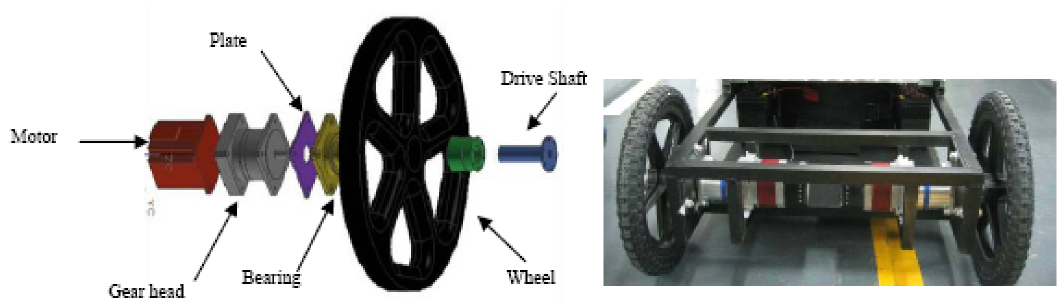
Johnny-5's chassis has proven to be durable and functional after years of constant use and abuse. The chassis is rock solid and measures 25 by 35 by 8 inches with a competition height of 65 inches (Figure 3.1). Two 16 inch rear drive wheels and a 10 inch front caster wheel provide Johnny-5 with a ground clearance of 3.75 inches. The chassis is constructed from welded 1" 6063 aluminum tubing, chosen for its lightweight and nonferrous characteristics which reduce magnetic interference with the digital compass. A 1/16 inch 6063 aluminum plate covers the bottom of the frame and plastic panels cover

Figure 3.1 – Johnny 5 chassis

the sides. Finally, an aluminum cover protects the onboard equipment from the elements. Placing the caster wheel in the front of the vehicle allowed for much greater control of where the vehicle's body would travel. The rear wheel drive design and weight distribution aids in traction as 60% of the weight is in the rear of the vehicle. Care was taken to mount heavy components such as the generator and batteries lower in the vehicle to lower the center of gravity, improving Johnny-5 maneuverability and stability.

3.2 Vehicle Drive Train

Johnny-5 is driven by two QuickSilver Control Silvermax 34HC-1 drive systems. Figure 3.2 shows an exploded view of the right drive train (a) and the assembled rear drive (b). The components of each drive system include a 16 inch composite drive wheel, a 10:1 NEMA 34 gear head, a 1/8 inch steel mounting plate, a Torrington PT Survivor bearing, and a custom drive shaft for each wheel.



(a) (b)

Figure 3.2: (a) CAD exploded view of the drive shaft assembly, (b) drive train mounted in the rear of Johnny-5

At the 2006 competition one of Johnny-5's gear head output shafts sheared off the night before competition. Because of this, the team has been considering ideas to prevent fatigue failure to the output shafts. After a demonstration in the fall of 2006 where the other output shaft failed, improving the reliability of the drive train became an issue the team had to pay attention to this year. Fatigue failure is an issue on Johnny-5 because of the direct drive method of attaching the motors to the drive wheels. The majority of the vehicle weight is placed in shear on the gear head output shaft which was not designed to take significant shear loading. The chassis of Johnny-5 made replacing the direct drive with a chain drive system not practical. Failure stress analysis was done on the output shafts assuming worst possible conditions; an output shaft supporting the entire weight of the vehicle during dynamic loading. After looking over the calculations and restrictions it was determined to replace the output shaft with hardened steel shafts held in place by two separate spring pins. This improvement will greatly improve the life of the wheel shafts.

4.0 Electronics

The electronics system serves as the central nervous system for the vehicle, making sure that all sensors and devices receive the appropriate power. The new power board provides updated safety and a greatly increased ease of maintenance.

4.1 Johnny-5 Power System

Johnny-5's power system consists of two Hawker Odyssey PC535 dry cell batteries, a Soneil 24V 8 amp battery charger, and a Yamaha EF1000iS generator. The Odyssey PC535 is a sealed dry cell battery that recycles its internal gas during operation and charging. The generator has a dry weight of 27 lbs and will produce 900 watts of power, while independently adjusting engine speed to match power demand, resulting in greater fuel efficiency and reduced noise. The generator powers the laptop and charger, which in turn continuously charges the batteries, which power the sensors and motors. This increases safety while providing a 10 hour runtime before refueling is necessary.

4.2 Power Distribution

The power distribution for the 2007 Johnny-5 is accomplished through the custom designed circuit boards shown in Figure 4.2a. This board is fixed inside a vented enclosure in the vehicle during operation, which is a safety improvement considering Johnny-5's old power distribution

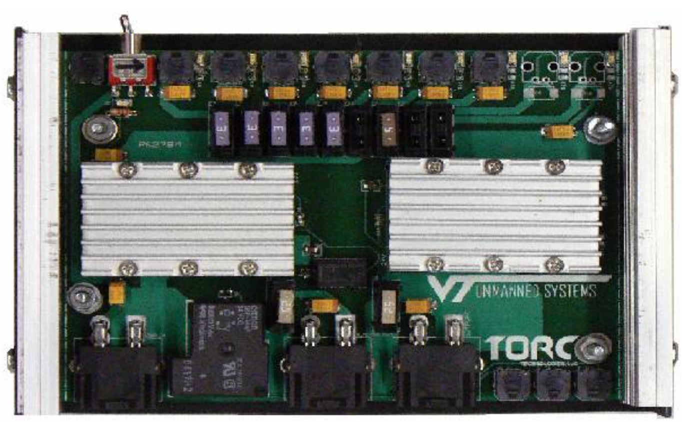


Figure 4.2a: Power distribution board

board was an open DIN-rail system. The main power input sends 24 Volts through three DC to DC Voltage regulators; one 24 Volt regulator and one 12 Volt regulator which regulate voltage to the sensors. The power to the motors is unregulated 24 Volts. This power can be interrupted by a remote emergency stop or a hard wired button. The vehicle uses one main power switch to control the entire electrical system.

The distribution section of the board receives regulated 12 and 24 Volts and is responsible for power distribution and monitoring. The regulated 24 Volts is distributed to the Laser Range Finder (LRF) and to an auxiliary connector. The regulated 12 Volts is sent to the compass, GPS, camera, and to two auxiliary 12 Volt connectors. Each of these connectors has an individual fuse to avoid damage from a power surge. The entire power distribution system is outlined in Figure 4.2b.

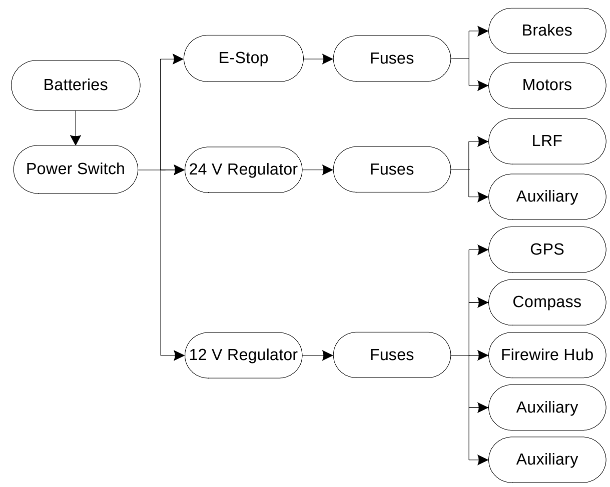


Figure 4.2b: Schematic for power distribution

5.0 Sensors and System Integration

Electronic sensors and a laptop computer are used to gather course information, process the data, and decide the vehicle's path. Four sensors are used to obtain peripheral data. Table 5-1 briefly explains the primary function of each component and how it is used in the Autonomous or Navigation Challenges.

Figure 5.1 illustrates the communication protocol used for each sensor and its path to the onboard computer. Integral to providing the computer with the ability to individually address each sensor is the PCMCIA to serial adaptor. This component is simply placed in the laptop PCMCIA slot and provides four RS-232 communications lines for each sensor. In the original design a RS-232 to USB converter was used to interface the serial devices to the computer. During testing, this converter was found to malfunction unexpectedly. Due to the difficulty in adequately resolving the issue, the design team decided to replace the converter with National Instrument's PCMCIA-232/4 serial to PCMCIA 4 port adaptor. Testing the current system has shown the serial to PCMCIA adapter to be reliable.

Table 5-1: Sensors used on Johnny-5

Sensor	Function	Characteristics
Unibrain Firewire Digital Camera 	Line detection	Resolution: 640x480 94° Diagonal Field of View Update Rate: 15 FPS
SICK LMS-221 Laser Range Finder 	Obstacle Detection	Resolution: 1° 180° Field of View Update Rate: 15 Hz Accuracy: + 5cm (range 3.2-65.6ft)
Novatel ProPack LB+ DGPS 	Vehicle Localization	Accuracy: 15 cm 99% using Omnistar correction service Update Rate: 2 Hz
PNI TCM2-20 Digital Compass 	Vehicle Heading	Resolution: 0.1° Accuracy: + 1° when tilted Update Rate: 15 Hz

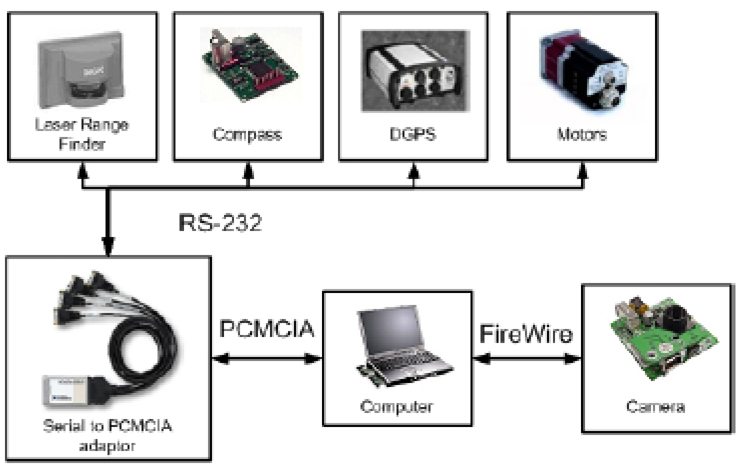


Figure 5.1: Sensor communication layout on Johnny-5

5.1 Sensor Communications and Data Integration

As described above, each sensor has an independent line of communications to the computer. However, care must be taken to analyze the data coming from each sensor correctly. For each sensor a custom driver was written to output the most recent complete data frame. This prevents a partial message from corrupting navigation algorithms and aids in error checking. Once valid data is obtained, it is combined into a common coordinate frame centered on the vehicle. Once in a common coordinate frame, the data can then be used to navigate the vehicle. Aside from converting sensor data to a common coordinate frame, there is no additional signal processing that must be done. The laser range finder, differential GPS, digital compass, and motors on Johnny-5 have signal processing built into their hardware.

6.0 Software

All software running on Johnny-5 was developed using National Instruments LabVIEW 8.2. During the past few years, the Autonomous Vehicle Team of Virginia Tech has experienced great success with LabVIEW. The use of LabVIEW simplifies coding and expedites system development.

6.1 Kalman Filter

Value engineering has played an important role in the redesign of Johnny-5. At last year's competition one of the judges mentioned that a filter could be used to improve lesser accurate GPS solutions. This prompted some research into the field and development of several filtering techniques. The solution that will be used this year is a combination of a less expensive GPS system, motor encoders that are already on the motors, and the digital compass already used by the software.

The prior solution was a Novatel ProPak which runs at a list price of $8,000. Our current solution is a U-Blox GPS receiver that is $200. The other sensors that are used to improve the GPS signal are already on the vehicle. The data from the various sensors will be combined with an Extended Kalman Filter (EKF).

The Kalman filter is an excellent way to fuse data from multiple asynchronous sources. It works on a 'predict and update cycle', where the prediction is made on a system model and the model is updated with the sensor measurements. The update is controlled by the Kalman Gain, which can dynamically control the weighting of the measurements. Resulting in an estimate of the state (position) that is better than any of the single sensors could give on their own. We use an Extended Kalman Filter because the nature of our model is nonlinear, the EKF allows for this.

In order to implement the filter, we have to verify that the data is valid. If the incoming data from a sensor is not valid, then it is not used in the update phase of the filter. If there is no valid data present, the update phase will be skipped. Despite the added computational load, we have been able to run the filter at 25 Hz, but we typically run it at 20 Hz, in order to synch it up with our decision cycle of the Navigation Challenge. In addition, we found that adding a bias to the U-Blox GPS would greatly improve the performance. Figure 6.1a shows the data taken in from the two GPS systems and the dead reckoning for an approximate 15 meter run. Figure 6.2b is the same run, but with the U-Blox GPS bias in place and the filter running in real time. Neither the dead reckoning nor the (bias corrected) U-Blox GPS is a bad estimate for this short run, but the combination of the two offers a more reliable, closer to reference, estimate.

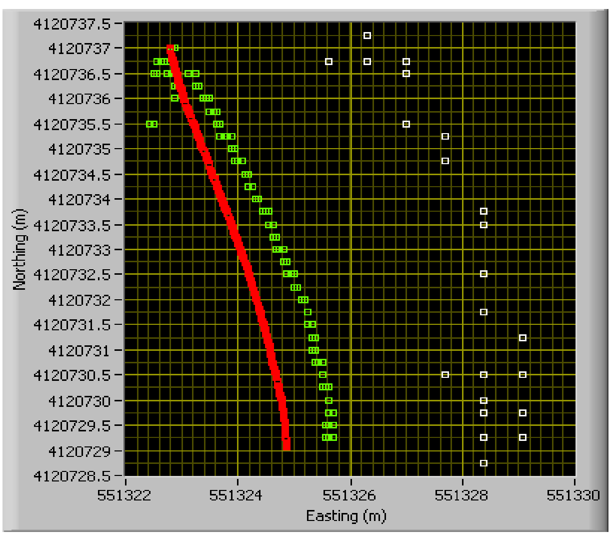


Figure 6.1a. Individual sensor group outputs. The green is the GPS data from the Novatel Propak (used as a reference), the white is the output from the U-Blox GPS, and the red is the output using only dead reckoning.

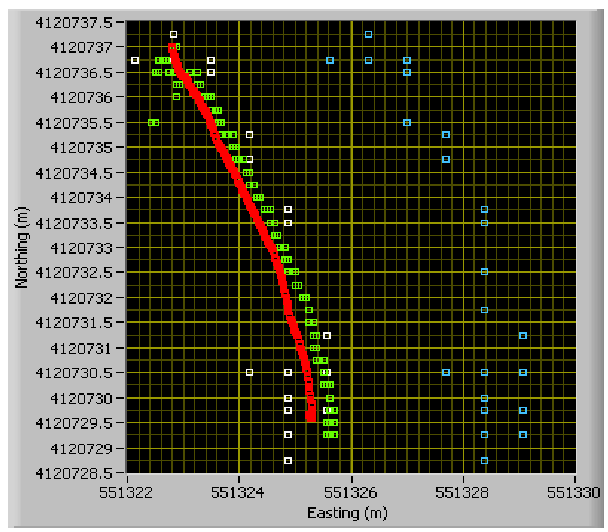


Figure 6.1b. Filtered output. The green is the GPS data from the Novatel Propak (used as a reference), the blue is the output from the U-Blox GPS, the white is the bias-corrected U-Blox data, and the red is the filtered output

The power of the Kalman filter for our application lies in its versatility. The filter can handle asynchronous data and can scale with the number of measurements we take. Our current filter takes in one measurement for every state in our filter. If we were to include other measurements (visual odometry for velocities, landmark navigation for position, etc.) we could easily assimilate those measurements into our estimate of our position and increase the accuracy at little cost to the consumer.

6.2 Software Structure

To simplify software development, the programming structure shown in Figure 6.3 was implemented in both the Autonomous and Navigation Challenges. This structure is primarily reactive and does not rely on complex global map building to navigate. Rather, a series of behaviors is used to determine the vehicle path in response to sensor data. Sensor data is collected simultaneously through individual communication channels and processed by the navigation algorithm to determine a desired vehicle heading. Once a desired heading is computed, the navigation software plans a path for the vehicle around any detected obstacles, which is commanded by the motion control.

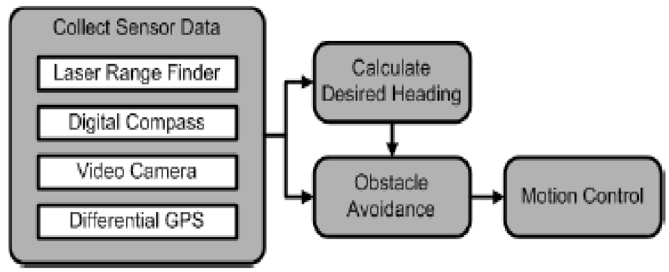


Figure 6.3: Software architecture for autonomous and navigation challenges

6.3 Autonomous Challenge Lane Following

The software programming to keep Johnny-5 within the course boundaries is a fairly simple lane following algorithm. As shown in Figure 6.4, the lane following task can be broken down into subtasks consisting of detecting the lines, analyzing the lines, and setting the desired travel direction.

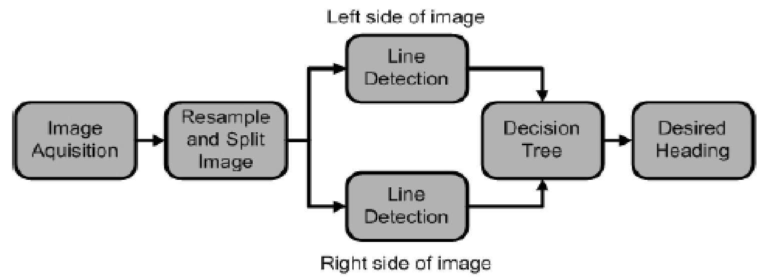


Figure 6.4: Autonomous challenge software diagram for calculating the desired heading

Once the image has been acquired it is reassembled from 640x480 to 160x120 pixels, passed through a threshold operation, and divided into left and right halves. Respectively, these three steps are intended to reduce processing time, eliminate noise, and facilitate a structure for the line detection algorithm. Lines are detected using an algorithm known as the Hough Transform. Figure 6.5 shows the result of the Hough Transform used on an image of a line on grass after a threshold operation. Notice that the Hough Transform is not affected or skewed by the noise to left of the line in Figure 6.5. The result of the Hough transform is a score indicating how many points are on the line and an equation, giving both the location, and direction of the line. This information is then passed to a decision tree which determines the best direction for the vehicle to move towards.

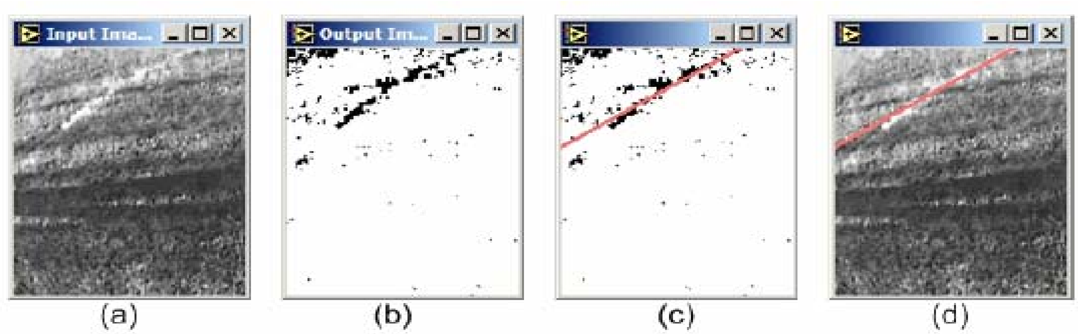


Figure 6.5: Original Image (a), After Threshold (b), Detected Line (c), and Original Image with Detected line (d) Image with detected line super-imposed

If the image contains both lines, the desired direction is set so the vehicle will head to the center of the lines. If only one line is present, the software assumes the lines are 8 feet apart and sets the direction based on the position of the known line. The desired heading, as well as the location of the lines is then passed to the obstacle avoidance software.

6.4 Autonomous Challenge Obstacle Detection and Avoidance

The obstacle avoidance process begins with mapping detected obstacles into 3x3 inch squares in an occupancy grid. During the Autonomous Challenge, the equations of lines generated by the Hough Transform are considered obstacles. Johnny-5 also considers distinct regions containing more than 80 white pixels as potholes. The lines and potholes detected by the camera are mapped into the same occupancy grid as the laser range finder data, resolving all the data into a common form.

Johnny-5's software then examines 36 potential arc shaped paths, beginning with the path closest to the direction determined by lane-following. An obstacle will lie in the vehicle path if the distance from the obstacle to the arc's center is between the turning radii of each wheel, as indicated in Figure 6.6. Each path is checked for obstacles, and the final path of the vehicle is chosen by combining the following factors for each path: distance to closest obstacle along path, deviation from desired heading, and the deviation from the last heading chosen. Once a suitable path is selected, motor velocities are calculated and commanded to the Silver Nugget controller.

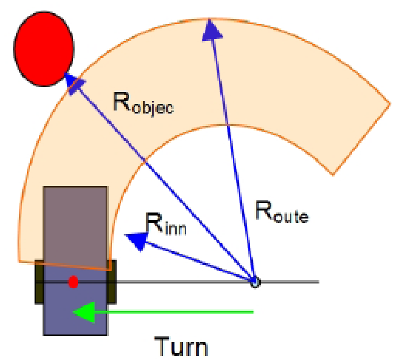


Figure 6.6: Distances used to calculate vehicle path to avoid obstacles

6.5 Navigation Challenge Software

Johnny-5 placed second in the 2004 IGVC Navigation challenge, but suffered due to unreliable software. In 2005, the navigation software was redesigned and Johnny-5 placed third, behind two other Virginia Tech vehicles. For the 2007 IGVC, this software has been expanded to allow for a dynamic selection of waypoints. In other words, instead of giving the vehicle a list of waypoints to visit in order, the team can merely provide the waypoints, and Johnny-5 will select the best order to visit them. The order of waypoints is selected using a derivative of the Dijkstra's Algorithm, using the distance and a multiplicative weighting factor. The algorithm computes the total weighted distance traveled through the course and then sets the order of waypoints to visit. One concern the team had was the time it would take for this rigorous method. Testing has shown that for a small set of points, the cycle times only increase slightly and have no visible effect on the performance. Figure 6.7 shows the interface for inputting the weights. There, starting at point 1, traveling to point 2 may be longer than point 3, but because of the weighting, the algorithm may choose point 2 as the second point in the path.

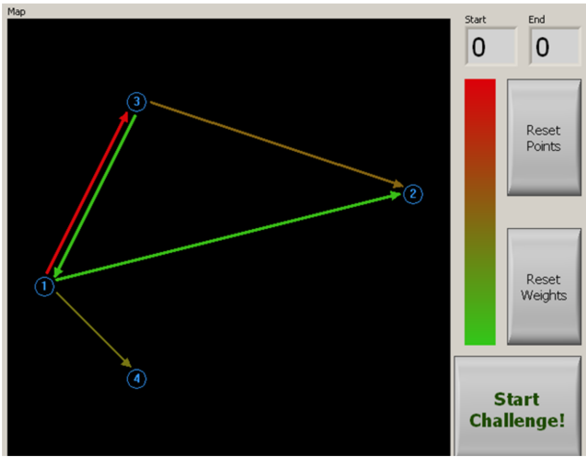


Figure 6.7: Interface for inputting weights for the Navigation Challenge. Weights are color coded from green (easily passable) to red (difficult terrain). No information is treated as unity.

At its core, Johnny-5's waypoint navigation employs subsumptive architecture of two key behaviors. In the first behavior a desired direction is computed, moving the vehicle directly towards the waypoint. If an obstacle is encountered while traveling to the waypoint the obstacle avoidance behavior then takes over motion control. Once the obstacle has been cleared, potential field's waypoint navigation is then resumed.

The forward-mounted laser range finder is used to obtain a polar plot of the obstacles in front of the vehicle. Next, analyzing the predefined regions in front of the vehicle, depicted in Figure 6.8, the vehicle determines which direction to turn based on which region contains an obstacle and the direction to the goal waypoint. The first priority is to avoid obstacles in the center (red) region, followed by the middle (blue) and side (green) regions respectively. Testing in the simulator indicated a tendency for the vehicle to oscillate away from and then back into an obstacle that was detected in the center region. To correct this the two middle regions are grown out to the length of the center region and an obstacle that was avoided in the center region would then transition into a middle region and no longer turn back towards an obstacle. This is indicated by the dashed blue lines in Figure 6.8.

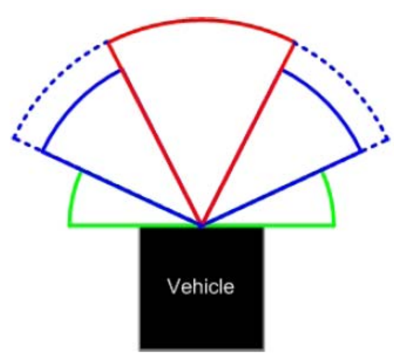


Figure 6.8: Obstacle avoidance regions for the navigation challenge. Data is from a single laser range finder scan.

Testing with both the simulator and the real vehicle has proved this algorithm both robust and simple to implement. The emergent behavior from this algorithm is a tendency for the vehicle to round obstacles or follow along a wall of obstacles until a clear path to the goal is reached. We are confident that this reactive, behavior-based cone-avoidance algorithm will once again perform well in the 2007 IGVC.

7.0 Vehicle Cost

Table 7-1 shows an itemized breakdown of Johnny-5's cost at retail prices and the cost to the team.

Table 7-1: Johnny-5 vehicle cost

Item	Retail Cost	Team Cost
6063 Al Square Tubing 1-1/4" X 1-1/4"	$120	$120
Electronics Box	$100	$50
Aluminum Cover	$300	$0
Electronics Parts	$500	$500
Laptop	$2500	$0
Serial to PCMCIA Converter	$495	$0
Wheels	$58.99	$58.99
PWR MOS UltraFET 80V/75A/0.010	$62.50	$0
12V sealed Lead-Acid Battery	$340	$0
24V/8A Battery Charger	$160	$145
16" Tuffwheels with Disc Brake Hub	$85	$0
48V DC High Output Servo Motors and gear head	$2450	$2450
Wide Angle 80.95 deg Firewire Camera	$81.75	$81.75
U-blox ANTARIS 4	$200	$200
LMS-221 Laser Range Finder	$5927.25	$5927.75
EF 1000is Generator	$700	$700
Total Price	**$13972.50**	**$10233.50**

The 2007 version of Johnny-5 uses a significantly cheaper GPS unit in the U-blox ANTARIS 4, as opposed to the Novatel ProPak. The price difference between the two units is $7795 retail. The retail cost of the vehicle with the ProPak is in excess of $20,000 while without the ProPak the price drops down to $13,000. The team feels that with the new Kalman Filter in place the extra money could no longer be justified in using a high precision, high accuracy, and expensive GPS unit.

8.0 JAUS Implementation

Virginia Tech has been exposed to JAUS messages for over two years. Previous students have developed a LabVIEW toolkit that can parse the JAUS header and handle many of the defined JAUS messages. Having worked with JAUS in the previous competition, the learning that took place was mostly reviewing the specific messages needed in the reference architecture.

Level 1 of the JAUS competition was completed last year and has remained unchanged. As dictated by the rules, the team wrote a separate JAUS component that will be running on the vehicle node to respond to the global waypoint query after validation of the message header. An OCU from last year was also modified to be able to send the query and interpret the response.

When using the toolkit developed at Virginia Tech, we discovered that one of the messages that we needed was not completed. Using the structure that was already in place, this was a minor obstacle in an overall smooth JAUS implementation.

9.0 Conclusion

Johnny-5 is an autonomous ground vehicle that was designed and fabricated by students at Virginia Tech. Johnny-5 was designed using the latest design and simulation tools, resulting in a reliable, compact, and safe system. An onboard generator can power the vehicle for up to 10 hours of continuous operation. A single powerful computer running National Instrument's LabVIEW software streamlined systems integration.
Over the years the Johnny-5 platform has proven to be reliable and dependable in both its performance and durability. The 2007 competition season marks improvements to make the vehicle more up-to-date, more versatile in the way it processes sensor inputs and structurally more reliable. With the addition of the Kalman Filter and the replacement of the GPS unit we believe that Johnny-5 can still deliver championship performance but now at a 35% reduction in cost to the end user. We believe that Johnny-5 will remain in high competitive form and will prove to be a valuable asset to future autonomous teams at Virginia Tech.