CALIFORNIA STATE UNIVERSITY, NORTHRIDGE 1958

RED RAVEN 2.0

Team Members
Alex Anikstein
Sevana Avanessian
Sultan Alkeebali
Harry Tamayo
Siqin Cai
Carolina Chica
Zi Teng
Ya Li
Eric Mejia
Robert Breyer
Omar Almahmood
Amiel Hartman
Po-Jen Wang

Faculty Advisor Statement

I hereby certify that the design and engineering of RED RAVEN 2.0, as described in this report, including new path planning, GUI, color detection, and JAUS algorithms, along with physical changes, including a new suspension system and cooling system, have been significant and equivalent to completion of a senior design course.

C. T. Lin
Department of Mechanical Engineering

1. Introduction

The Intelligent Ground Vehicle team of the College of Engineering and Computer Science at California State University, Northridge (CSUN) is proud to present RED Robotic Autonomous Vehicle Engineered at Northridge (RAVEN) 2.0. RED RAVEN 2.0 is a heavily modified version of last year's IGV with significant additions both to the hardware and software. Some of the noteworthy innovations of RED RAVEN 2.0 include a powerful liquid cooling system, a new suspension system, new edge-based vision algorithms, and a more stable, powerful, and user-friendly radial path planning algorithm.

1.1 Team Organization

The IGV project is an interdepartmental senior design course at CSUN spanning two semesters. The class is officially associated with the Mechanical Engineering Department, however only half of the team members this year are Mechanical Engineering majors, with the other half being Electrical Engineering majors. The team is broken into five subgroups: Mechanical, Electrical/Power, Navigation & JAUS, Vision, and Cognition/Motion Control, each with a group leader and between one and three other students. Additionally, there is a project manager responsible for working with the group leaders to assure integration of all of the subgroups and a faculty advisor.

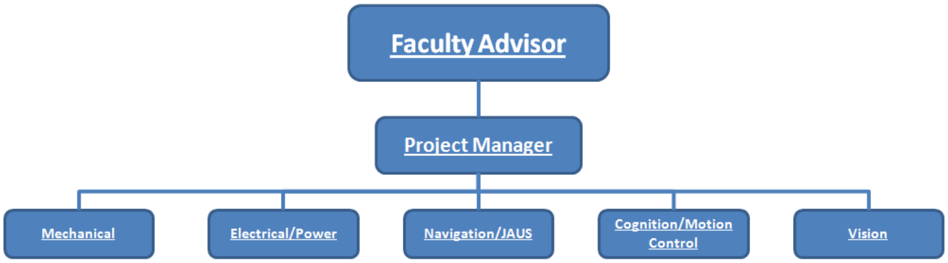

Figure 1.1 – CSUN's IGV Team Organization

1.2 Overall System Integration

RED RAVEN 2.0 combines data from a camera, Scanning Laser Rangefinder (LRF), Differential Global Positioning System (DGPS), and an Inertial Measurement Unit (IMU) to gather information about its environment and location, which it then sends to Cognition. Cognition combines all of this information on a Toshiba 16" laptop to evaluate the ideal path for the IGV. Motion Control then uses this information to calculate the proper velocity and acceleration. Using the on board battery packs and custom printed circuit board, the IGV supplies the required voltages and currents to the motors, executing the calculated path.

2. Mechanical Design

The purpose of the mechanical design of RED RAVEN 2.0 is to improve on the robot's efficient and dynamic ability to navigate through an autonomous course. Last year, RED RAVEN was introduced as a revolutionary IGV design featuring a lightweight and compact aluminum design, diamond-shaped chassis (the Linked-Bogie dynamic frame), vertically stacked top platform, and low center of gravity located directly above the differential-drive axle allowing for RED RAVEN to be extremely maneuverable. In addition, a custom-design drive axle decoupling mechanism was created, along with a form-fitting transparent polycarbonate weatherproofing shell and a battery tray with sliders

By analyzing the behavior of RED RAVEN 2.0 from last year's competition, two areas of improvements were found: top platform tipping and an overheating laptop causing all of the IGV's algorithms to crash. This year, the Linked-Bogie dynamic frame has been upgraded with newly-incorporated dual-action dampers utilized to minimize platform tilt, improve stabilization and vibration. A cooling system for the laptop has also been designed controlling the ambient temperature around the laptop's hottest areas and minimizing overheating issues.

2.1 Improved Damped Link-Bogie Frame

RED RAVEN 2.0's damped linked-bogie frame is constructed from 1-inch 6063 aluminum tubing and quarter inch sheet metal as shown in Figure 2.1, which was chosen for its high strength to weight ratio. The frame was TIG welded together to prevent corrosion.

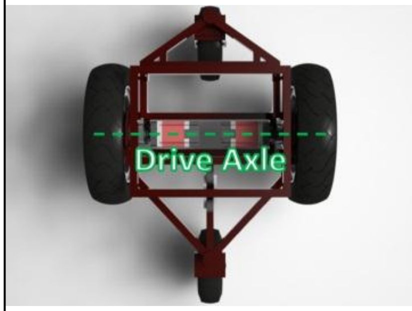

Figure 2.1: Last year's chassis

The chassis has a diamond shape with the CG of the vehicle centered and located above the drive axel, allowing for zero-radius turning on the spot. In Figure 2.2, the chassis has two floating caster wheels (front and back) that balance the weight of the vehicle, and the top platform is connected via pivot points on the back link (blue) and a slider on the front (green), giving RED RAVEN its flexibility.

This year, the Linked-Bogie dynamic frame has been upgraded with two dual-action dampers (circled in Figure 2.2). The dampers assist in minimizing vibration on all critical electrical components, guaranteeing that no connections are broken, and eliminating the danger of the top platform (yellow) tipping over during sudden stops and deceleration down inclines, thus protecting the platform, all components, and ensuring a more stable platform for the vision system.

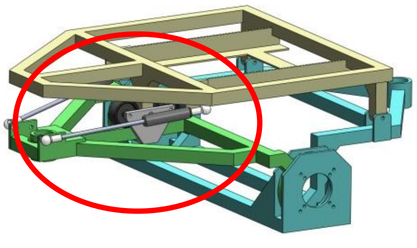

Figure 2.2: Side view showing dampers

As shown in Figure 2.3, the dampers connect the top platform to the furthest end of the front caster (green). This provides a longer stroke length to dampen the movement of the top platform and resists the back caster wheel's forward motion when the vehicle comes to a sudden deceleration, thereby forcing the back caster wheel to remain in contact with the ground and controlling the forward lurch of the top platform. Red RAVEN 2.0 keeps its original flexible chassis design allowing each wheel to move independently and remain in contact with the ground.

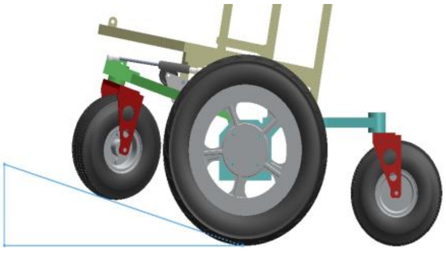

Figure 2.3: Dynamic Frame

2.2 Liquid Cold-Plate Cooling System

Previously, RED RAVEN featured 120mm computer cooling fans to cool the laptop; however they were insufficient and frequently lead to software crashes. To fix this, RED RAVEN 2.0 has now been fitted with a custom-designed liquid cold-plate cooling system. The system is made from a compact reservoir, radiator, and pump, along with copper tubing and HFE-7100, a non-conductive heat transfer fluid, as shown in Figure 2.4. The copper tubing is routed through custom machined aluminum plates, and then coated in thermal putty to eliminate any air gaps inside of the assembly. Additionally, a new vent was created and attached to the exhaust ports of the laptop, with 120mm fans mounted at the far end drawing additional hot air out of the laptop. Between the cold-plate and the new vents to amplify the effects of the fans, the internal temperature around key components, such as the hard drive or CPU, can be controlled much more effectively. This entire system is modular in design and can be easily transported from platform to platform.

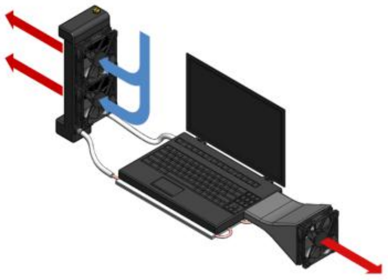

Figure 2.4 – Liquid Cold Plate Cooling System

2.3 Center of Gravity

The component layout was designed with the purpose of achieving two goals. The first was to keep the center of gravity (CG) located as close to the drive train or lower if possible, increasing maneuverability and stability against tipping over. The second goal was to concentrate the mass as close to the center vertical axis as possible, which would minimize the rotational inertia. Once the vital component layout was established and both goals were met, a vertically stacked frame was created for the upper platform as shown in Figure 2.5. The vertical stacked frame was chosen because it allowed accessibility and met the component layout criteria; the heaviest components placed at the bottom centered in the frame and the lighter components placed at the top. Although most of the components were positioned based on weight, three components were positioned based on function. The IMU is a relatively heavy component that had to be placed at a centralized location in order to achieve its function of obtaining accurate data. Another heavy component that could not be placed low was the laptop in order to allow easy user interface access. Lastly, although the PCB is one of the lightest components, it is mounted in a central location in order to minimize cable lengths. As a result of achieving the design goals of having a low and centered CG, RED RAVEN 2.0 can perform quick turns and remain stable.

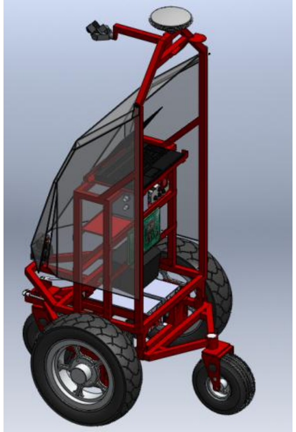

Figure 2.5: RED RAVEN 2.0

2.4 Drive Axle Decouplers and Polycarbonate Shielding

One of the most innovative and overlooked aspects of Red RAVEN 2.0 is the drive wheel decouplers. The advantage to a decoupling system is the ability to transport the robot easily and with very little effort, and without causing damage to the drive train. As shown in Figure 2.5, the drive wheels are mounted to the wheel mount. Another great innovation on RED RAVEN 2.0 is the polycarbonate shielding that allows for the components to be covered and protected from the environment. Polycarbonate was chosen for its strength, transparency, low weight, and ability to shape. With the use of thumbscrews, panels were attached to the frame allowing for both installation and removal of panels without the use of any tools.

3. Electrical/Power Systems

The power system of RED RAVEN 2.0 has proven itself to be both efficient and reliable. This is accomplished by concentrating all power distribution on a custom printed circuit board. Onboard DC/DC converters, the clamp module, and protective circuit modules (PCMs), which are built in the battery packs, provide protection against overvoltage or overcurrent, guaranteeing stable power supplies to all of the components. The Lithium Ion Polymer batteries are the main power source.

3.1 Electrical Configuration and Printed Circuit Board Design

The power system integration is accomplished using a centralized control system. A printed circuit board provides a more stable and reliable platform compared with using exposed terminals or wires only, shown in Figure 3.1. The PCB itself is a durable design consisting of electrical traces made from 2 oz/ft^2 copper paths and is able to handle currents as high as 60 A. Two trace layers were chosen in designing the PCB; the front traces all consist of various voltages while the entire back is one large ground plate. This simplifies trace routing considerably and allowed for a much more compact design.

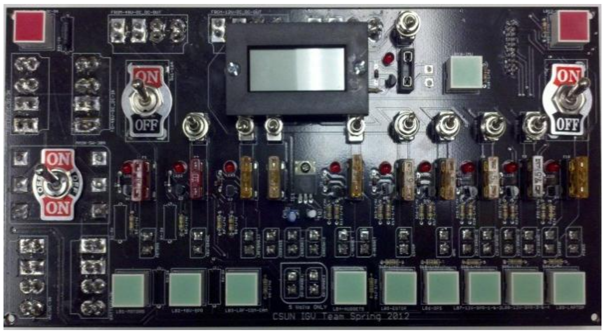

Figure 3.1: PCB Layout

Figure 3.2 shows a brief overview of RED RAVEN 2.0's power system. Main power is supplied by either the Lithium Ion Polymer batteries or an AC power supply. Battery voltage is converted and regulated by two separate DC/DC converters: 48V DC for the motors and 12V DC for all other electrical components and provides electrical isolation between the motors and sensitive electrical sensors, such as the Laser Range Finder. Physical switches are used to individually power on sensors and components, while fuses are in place to protect each component from sudden current surges. Indicator LEDs are also present to indicate whether or not power is being sent to a component.

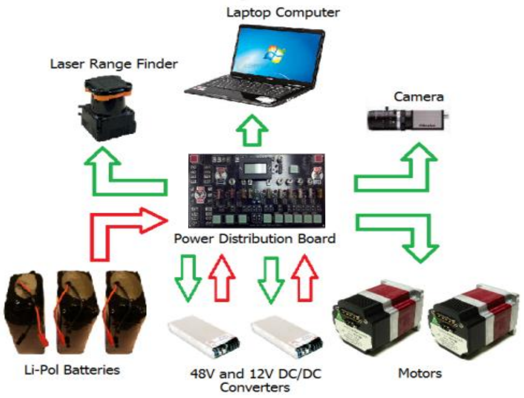

Figure 3.2: Electrical Component hierarchy

3.2 Power Analysis and Battery

The power consumption of RED RAVEN 2.0 was analyzed at both normal and extreme load conditions, shown in Table 3.1. Low load conditions consisted of the IGV traveling at approximately 1 mph, and extreme load conditions with it traveling at a top speed of 6.5 mph. The data indicated that under extreme conditions, the IGV would need approximately 1235 W.

In order to supply sufficient amount of power to RED RAVEN under these conditions, a minimum of three Lithium Ion Polymer batteries are needed. Each battery has a nominal voltage of 14.8V and is limited to 30A maximum output, which provides a nominal rated power of 444.4 W. By configuring three batteries in series, a total of 1333.2W is provided and the overall current is limited to a maximum output of 30A. In order to prevent the batteries from getting damaged by current, Protect Circuit Modules are used to monitor each battery pack during charge and discharge.

Table 3.1: Power Consumption in Watts		
Type of Load	**Low Load**	**Extreme Load**
Total Base Load	177	
Transient Motor Load	154	910
Total Load	331	1087
Total with 88% DC/DC Efficiency	**376**	**1235**

3.3 Emergency Stop

The emergency stop system on RED RAVEN 2.0 functions by making use of the motor controllers' built-in stop conditions to bring the robot to a controlled and safe stop. When the emergency stop is activated, a logic HIGH (+5V) is sent to the motor controllers' input/output pins. When the emergency stop is not activated, a logic LOW (ground) is sent to the motor controllers. A series configuration is used for the pushbutton stop and the wireless stop, shown in Figure 3.3, to ensure that they do not conflict.

Figure 3.3: The push button and wireless emergency stop.

3.4 Motor Control

The motor and nugget used on RED RAVEN 2.0 are the QCI-A34HC-2 and QCI-N3-E3-EE-04, shown in Figure 3.4. Quickcontrol software is used to initialize the motors and clear any leftover commands. The control panel in the Quickcontrol can be used to operate the motors directly for testing using the “jog” command. For manual control, the motors are simply enabled in the software. The only issue is that the motors must be powered before the nugget to prevent software errors, so it is important that the proper start-up procedure is followed.

Figure 3.4: Motors and Nugget

3.5 Audio-Controlled Relay

In order to comply with the IGVC rules, RED RAVEN 2.0 must indicate when it is powered on through the use of a solidly lit light, shown in Figure 3.5.The light be solid whenever the IGV is on and must flash whenever the IGV operates autonomously. When RED RAVEN 2.0 is set into autonomous mode, a 500Hz square wave sound file is triggered in LabVIEW. This sound file creates an output voltage waveform with a peak to peak voltage of 1.4V. It is sent through an amplifier, filter, and rectifier in order to trigger a relay, shown in Figure 3.6, to control the light and cause it to flash. Overall, this process provided a simple and easy solution ensure safety for those around the IGV.

Figure 3.5 – Indicator light

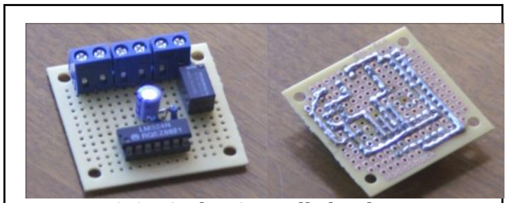

Figure 3.6 - Audio-Controlled Relay

3.6 Arduino Light Sensors

Due to conflicts during last year's competition, the IGV now has a light sensor, which it can then use to adjust the vision data and prevent the robot from being blinded. There are several ways to measure the light intensity of an area, but the options were eventually narrowed down to a TSL2561 digital luminosity sensor from Adafruit industries. This light sensor would be paired via I2C with an Arduino UNO microcontroller, as shown in Figure 3.7. The microcontroller would feed ambient light values to the laptop for light balancing. The specific microcontroller and Arduino were chosen because of their lower power consumption, simple interface, and straightforward mounting capability. A simple USB connection and some minor programming was all that was needed to integrate Arduino into RED RAVEN 2.0's current system.

Figure 3.7: Arduino UNO Microcontroller

4. Navigation/JAUS

The navigation subsystem leads the robot between waypoiunts until the course is completed. To do this, it must use a Differential Global Positioning System along with an Inertial Measurement unit to accurately locate the robot and the coordinates to which it must travel. This data must eventually be provided to cognition so that it may identify the most direct path. This group is also in charge of programming the JAUS architecture in LabVIEW for the IGV.

4.1 Navigation Hardware

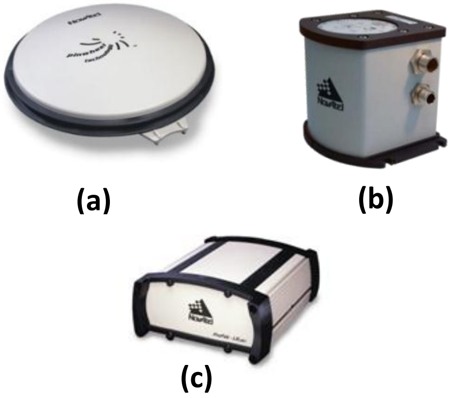

Figure 4.1 – Navigation hardware used on RED RAVEN includes an (a) antenna, (b) IMU, and (c) receiver.

A NovAtel Synchronized Position Attitude & Navigation system is used on RED RAVEN for navigational purposes. The system uses a GPS-702L antenna and a LN200 Inertial measurement unit (IMU) to send signals to the ProPak V3 receiver. The hardware previously mentioned is shown in Figure 4.1. A high performance Differential GPS service is provided by OmniSTAR, which increases the positioning accuracy to 10 centimeters. When GPS data is not available and the robot is in motion, the positioning data is provided by the IMU, which also delivers a faster refresh rate of up to 200 Hz. The IMU last year was not used to its fullest potential, which has been more thoroughly tapped for RED RAVEN 2.0. The ProPak receiver synchronizes data from the GPS antenna and the IMU to provide latitude, longitude, and velocity data to the computer. The receiver interfaces with the computer via an RS232 serial connection at a baud rate of 460.8 kBd while updating at 40 Hz.

4.2 Navigation Software Strategy

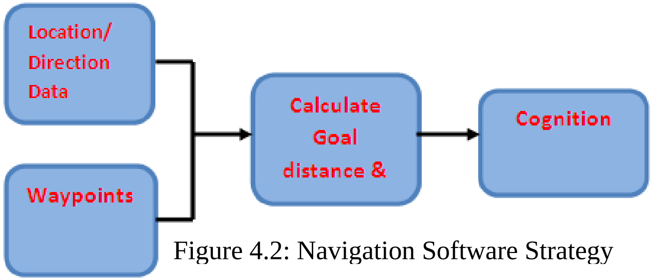

Figure 4.2: Navigation Software Strategy

Figure 4.2 shows the overall software strategy for the Navigation challenge. It begins by accepting a list of waypoints along with the location and direction data. The small number of waypoints can be quickly solved using brute force algorithm. As soon as the ordered list is created the vehicle uses the first point as its goal. The Navigation program continuously calculates the bearing and distance between the vehicle's current position and its current goal waypoint. This is done using the Haversine and Great Circle formulas. The goal distance is constantly monitored, and when it becomes one meter, the IGV is determined to have reached the waypoint and it can be checked off. This angle and position are provided to Cognition, where they can be used for path planning.

4.3 Joint Architecture for Unmanned Systems (JAUS)

The JAUS protocol is designed by the Department of Defense to facilitate the interaction between autonomous systems. The purpose of the JAUS Challenge is to program a platform for remote communication between the users and the robot. In order to ease management of the design process the set of used commands are split into two categories: Non-Navigational and Navigational. The Non-Navigational commands cover Transport Discovery, Capabilities Discovery, and System Management (i.e. commands not related to the navigation process). Since these commands are one-time linear processes, they are written within the main JAUS program and put under an array of cases to perform upon receiving the commands. On the other hand, Navigational commands cover Velocity State Report, Position and Orientation Report, and Waypoint Navigation (i.e. commands that relate to navigation and may need to recur upon request). Considering the complexity and recurring nature, these commands are modularized and then combined into the main program in both case structures and sequential structures, depending on the application. This strategy eases the management and debugging of recurring, non-linear processes.

Figure 4.3 shows the received signal is sent to two separate areas of the program: one to non-navigation queries to report the received information, and to local position and waypoint settings. The position and waypoint

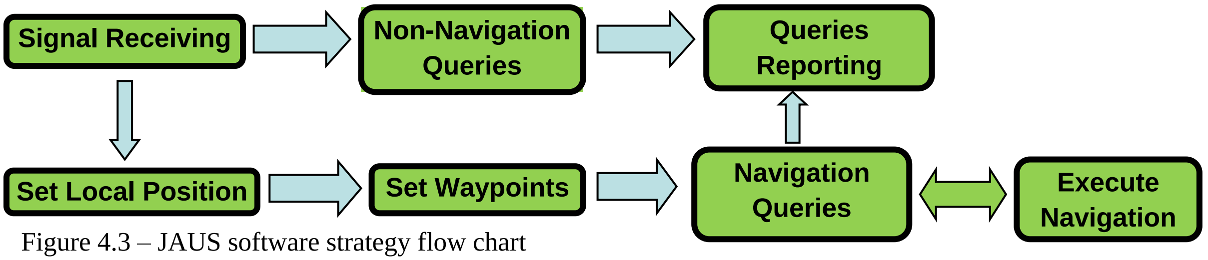

Figure 4.3 – JAUS software strategy flow chart

information is relayed to navigation queries, where it simultaneously executes the navigation data and reports information.

5. Vision System

The Vision system on RED RAVEN 2.0 utilizes a camera to detect and provide useful environmental information for Cognition to complete the autonomous task. The camera is mounted at the top of the robot and is oriented at an angle that provides a large field of view to maximize the detection area. The real time video feed that captures the front view of the robot is processed through a series of color detection and filtering algorithms to extract the white line data, identify obstacles, and generate a color map for Cognition to reconstruct the course and calculate the desired path. Vision is also in charge of finding and reporting the positions of the blue and the red flags placed at the end of the course for Cognition to generate imaginary ghost lines.

5.1 Hardware

The hardware used in the Vision system is a FOculus FO124/TC color IEEE1394 camera with a wide angle Computar T2Z1816CS varifocal lens and a polarized filter. The camera, shown in Figure 5.1, provides a high resolution and frame rates that surpass the current operating requirements. When accompanied with the wide angle lens, the camera is capable of achieving view angles up to 144^{o} horizontal and 109^{o} vertical. The lens provides manual control for precise aperture and view angle adjustments. IEEE1394, or FireWire, was chosen for power supply and data transfer because of its high speed and widespread compatibility with multiple computer interface setups. To further enhance the image quality of the camera, a polarized filter is attached to reduce glare under sunny conditions. A new piece of hardware that added to RED RAVEN 2.0 this year is the TSL2561 digital luminosity light sensors and microcontroller in Figure 5.2. This sensor is mounted in parallel with the camera to measure the light intensity of the environment local to the camera.

Figure 5.1 – The FOculus Camera (top) and Computar lens (bottom)

Figure 5.2 –The TSL2561 Light Sensor

5.2 Vision Software Strategy

The previous Vision system uses a Dynamic Color Detection approach to detect lines and obstacles. This approach analyzes the grass' average color values from a selected template area and identifies the pixels that are not associated with the grass color as either white boundary lines or obstacles. Though this approach is proved adequate to detect white lines and obstacles, it cannot differentiate between the two data types. In some scenarios, the obstacle data is preferred to be ignored because at the camera's angle, the height of the obstacle has the potential to block openings and traps the robot.

To properly detect white line boundaries while recognizing each individual obstacle and its color under a dynamically changing light condition, a new innovative Vision algorithm, the BGL Line Detection, accompanied by a digital light sensor, was created this year to improve the overall quality and accuracy of the Vision data. The first process of this algorithm adjusts the brightness of the source image using the TSL2561 digital luminosity sensor. One major problem faced with using the FOculus F0124TC camera is that it wasn't built with its own automatic light adjustment system. With the digital light sensor in place, the source image is automatically dimmed or brightened based on the light intensity value read from the sensor. This allows a better quality image to be processed through the rest of the Vision program. The light intensity value is also used to determine if the camera is blinded by the sun; if so, a signal is sent to Cognition to temporally ignore the incoming line data.

The flowchart for the Vision algorithms can be seen in Figure 5.3. The first filtering process after adjusting the source image is the grass filter. Grass filter subtracts half of the source image's green channel from the blue channel and output a grayscale image that greatly enhances the contrast of the white pixels from the grass pixels. Then, an obstacle filter is used to detect and remove pixels that are identified as obstacles from the grayscale image. The obstacle detection is completed by detecting the known obstacle colors in the image. Each obstacle color is individually detected using a unique mixture of the RGB and HSL color channels that highlights the pixels with the specified color from the rest of the pixels. For the orange barrel used in the competition, the two white lines on it are identified as part of the obstacle and are removed from the image. For flag detection, the red and blue flags are detected using a similar color detection method. The isolated flag pixels are calibrated for their height to determine their actual positions on the map.

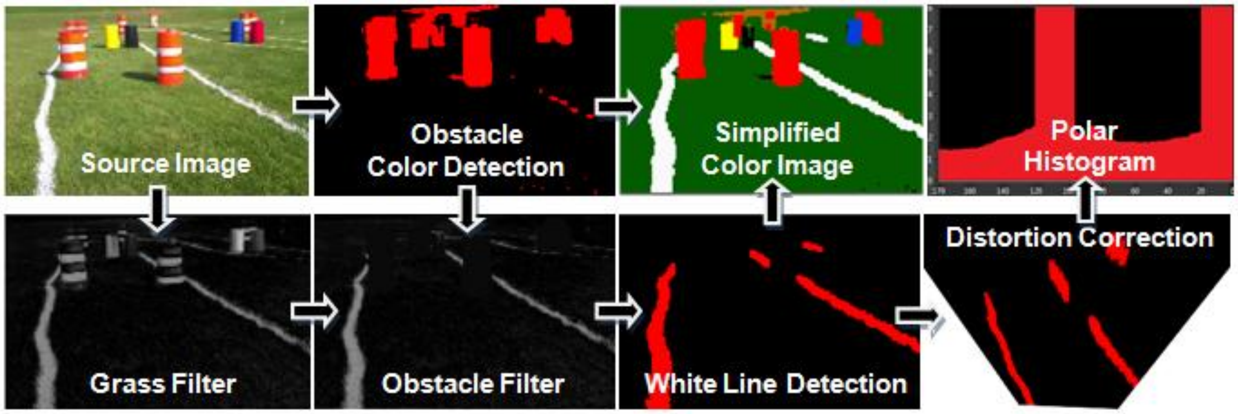

Figure 5.3: The BGL Line Detection Algorithm uses multiple filters and color detection techniques to detect white lines, obstacles, and colors of obstacles. Note: The black/red image is the default display format for binary images in LabVIEW's Vision Module.

After getting a clean grayscale image that contains only line data, a threshold is applied to find the white pixels in the image. Canny Edge Detection is also used to obtain edge information of the white lines. Edge detection is added for its excellent capability of seeing the blurred and darkened white lines under low light conditions. Next, the identified white pixels and the edge pixels are combined and a convex hull is performed to fill in gaps and to generate solid lines. The lines are then processed through a particle filter to eliminate small particles from noise and large particles from white barrels to define the white lines. Finally, the distortion of the line image is corrected and is converted into a polar histogram that contains the line boundaries' position information. The white lines and obstacles identified are also remapped into a simplified color image. These line and obstacle color data, along with the flag position data, are sent to Cognition for further analysis.

6. Cognition/Motion Control

The basic algorithms used by RED RAVEN 2.0 are similar to those used last year; however they have improved in three key areas. While they are still based on a radial approach, rather than a vector-based approach, the overall algorithm is much more robust and efficient than it was previously, with large sections of it being completely rebuilt and new features, including new methods of handling complex obstacle orientations and a new GUI.

6.1 Cognition System Integration

The Cognition and Motion Control systems on RED RAVEN 2.0 are responsible for generating a local map and utilizing the waypoint information provided by the Navigation system to determine an optimal path and speed for the IGV to autonomously navigate through the obstacle course and reach navigational waypoints. The various systems integrated into Cognition can be seen in Figure 6.1. The local map generated is a combined polar histogram that consists of the boundary data from Vision, obstacle data from a Hokuyo UTM-30LX Scanning Laser Rangefinder (LRF), and ghost line data that contains all the imaginary boundaries, such as those generated by flags. The Hokuyo LRF provides a wide scanning angle up to 270°, a fast scanning rate of 25 milliseconds, and a long detection range up to 30 meters away, with an accuracy of 1cm. The path planning and motion control algorithms used by RED RAVEN 2.0 are a heavily modified and improved version of last year's Radial Polar Histogram, or RPH approach. The major improvements and new features of this year's RPH include a new switchback detection algorithm, an improved ghost line generation algorithm for flags, a much more efficient RPH, and a new, real-time 3D rendered GUI.

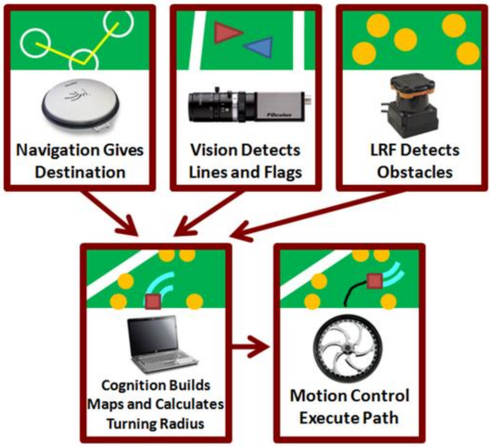

Figure 6.1: System integration flow chart.

6.2 Cognition/Motion Control Software Strategy

One of the features and strengths that differentiate RPH from other path planning algorithm is that, rather than calculating a linear heading to guide the IGV, it determines an instantaneous turning radius from the local map. This desired turning radius closely represents the vehicle's true motion that follows a curvilinear path, allowing the IGV to perform highly accurate motion executions.

The process of RPH begins with integrating the boundary

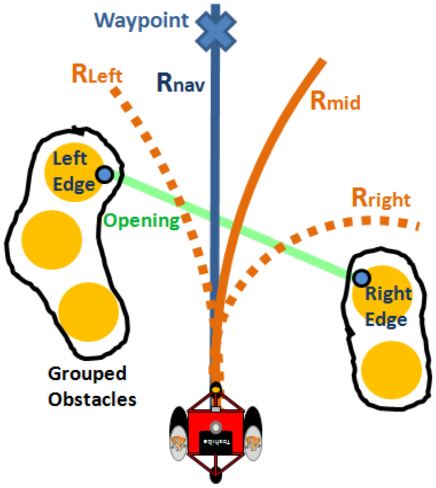

Figure 6.2: RPH uses group edges and the waypoint location to calculate four radii.

histogram from Vision, the obstacle histogram from the LRF, and the generated imaginary ghost line histogram into a local map. This local map is passed through a data filter to use a controlled range value to define the data of interest that will be further analyzed. Then, a grouping function uses the width of the IGV to group the remaining data points that are close to each other, which defines the obstacle blocks and the openings through which the IGV can travel. To allow proper waypoint navigation, the opening closest to the navigational heading is chosen to proceed and the positions of the obstacle edges on either side of the opening shown in Figure 6.2 are used to calculate the left-most radial path, Rleft, the right-most radial path, Rright, and the middle radial path, Rmid, that the IGV may travel. In addition, a navigation radius, Rnav, is calculated that guides the IGV to reach the navigational waypoint. Finally, an optimization function considers the size of the left and right radial boundaries and the waypoint location to select one optimal radius out of the four possible radii for motion control to determine the speed and execute the path.

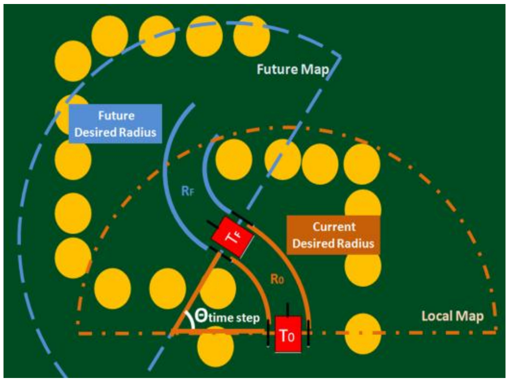

Figure 6.3: A future map is generated to determine the acceleration for each wheel.

To further improve the accuracy and smoothness of the IGV's motion, RPH's unique motion control algorithm calculates a real time-angular acceleration for each individual motor that reduces acceleration delays. The motor accelerations are determined by generating a predicted future map, as seen in Figure 6.3, and repeating RPH's path planning algorithms at the predicted future location. Lastly, the current and future motor speeds of the IGV are compared and current required motor acceleration is calculated.

6.3 Complex Obstacle Arrangement Modifications

While the obstacle avoidance provided by RPH is rather robust, special obstacle orientation detection algorithms and case structures were added to assist the IGV in navigating through complex obstacle arrangements. These complex obstacle arrangements include switchbacks, dead ends, and flags.

When encountering a switchback, an innovative, new switchback detection algorithm, demonstrated in Figure 6.4, will alert Motion Control to properly slow down the IGV to a much safer speed to prepare for a sharp turn. The switchback detection algorithm detects switchbacks by monitoring the orientation of obstacle edges that define the chosen opening.

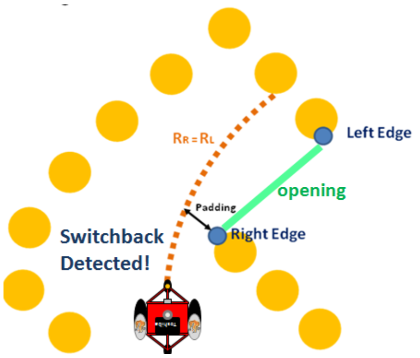

Figure 6.4: Switchback is detected using the orientation of opening's obstacle edges.

A dead end detection algorithm is also implemented to alert when no opening is available to travel. When a dead end is detected, RPH reduces the size of the range filter and checks if a new opening is available. When a small range is used and a dead end is still

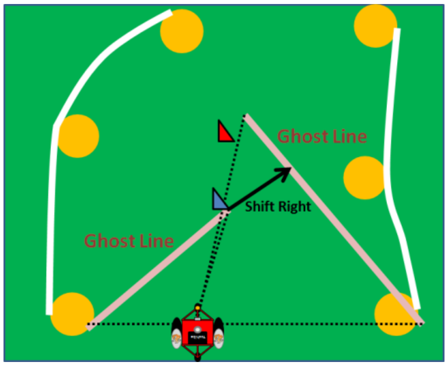

Figure 6.5: When generating the ghost lines for flags, additional padding is added to prevent creating a dead end.

detected, the IGV will perform an immediate stop followed by zero radius turn to search for an opening.

Lastly, when the Vision system detects flags, RPH uses the flag's color to generate imaginary ghost lines, as demonstrated in Figure 6.5, to block off one side of the opening to guide the IGV to the correct side of the flag. The Ghost line generation algorithm is improved this year to detect and add additional padding when necessary to prevent a dead end.

6.4 High Speed Operations

To further perfect RPH in terms of its computational efficiency, one important goal this year is to cut down the overall processing time of RPH to reduce RED RAVEN 2.0's reaction time and save the valuable computing power for other crucial systems. After detailed inspections, it was determined that the frequently used grouping algorithm alone performed over 16,000 calculations each time it was called when often the vast majority of the calculations could be proven mathematically unnecessary or redundant. The grouping algorithm has been rewritten and optimized this year to perform far fewer calculations while producing the same accurate results. Moreover, special attention has also been paid to adding different delays to RPH's sub-programs, allowing the computing power to be properly distributed to the different computational loads. These optimizations, along with many others, reduced the overall processing time of the RPH algorithms by an astonishing 80%, from approximately 60ms down to 10-12ms. Even after new, computationally and memory intensive features, including real-time 3D rendering of the environment, were added, the overall cognition processing time was 10% faster than it had been during last year's competition, on average approximating 45-55ms. Rewriting code with improved algorithms was a daunting challenge; however it was well worth it based on the results.

6.5 Real Time 3D Rendering GUI

With the efficient and speedy RPH algorithms, new advances have been made with regards to mapping and the GUI on RED RAVEN 2.0. A new, real-time 3D rendering GUI that uses LabVIEW's 3D graph to draw the identified 3D objects with their actual dimensions and color, along with displays of system information, is incorporated into the RPH front panel displayed in Figure 6.6. This is a massive feat which makes the GUI extremely user-friendly. The 3D rendering algorithm is currently capable of recognizing the cylindrical barrels used on the IGVC obstacle course. The barrels are detected by first grouping each individual obstacle's data points and applying a curve fit to the group to see if they represent a circular surface. To improve the accuracy of barrel detection, the grouped obstacle data points have to pass certain checks to be considered a barrel. These checks include the curve fit radius size, obstacle data size, and the variance of the data points from the center of each data group. The colors of the barrels are identified by overlapping and matching the grouped LRF obstacle data to the simplified color image provided by Vision.

Figure 6. 6: RPH's new Real Time 3D Rending GUI. RPH performs curve fit on the obstacle data to identify and draw individual barrels on the course.

6.6 Obstacle Tracking and Avoidance

Besides the user-friendly GUI, there are two major advantages of performing circular analysis. The first benefit is that the majority of obstacle data on the course can be simply represented and saved by three numbers. Only the central positions Xm and Ym, along with the radius R, need to be stored to represent the obstacles, saving computer memory during global map building for course memorization. Second, circular analysis of the obstacles allows obstacle tracking and avoidance for moving cylindrical objects. With the real time positions known, the perspective movement and the true movement of the moving cylindrical objects can be estimated and their predicted paths are used to generate imaginary ghost lines for the IGV to properly avoid them. In addition, when the cylindrical objects are known to be stationary, their perspective movement to the IGV is also used as a secondary source to track the movement of the IGV when GPS service is not available.

7. System Performance

Although RED RAVEN 2.0 owes much to its predecessor, it is now its own distinct IGV. The platform is much more stable, greatly reducing the risk of it toppling over and increasing its overall mobility. The algorithms that power it have received the most significant work, fixing many pre-existing issues, such as the potential to be blinded by too much light, issues with flags or dead ends, and general inefficiency in the code. The fact that the IMU is now fully functional on the IGV additionally increases the positioning accuracy and, coupled with the improvements to Cognition, almost assures that the IGV will be able to easily avoid any obstacle.

Due to the lightweight design and powerful motors, the IGV can reach speeds up to 6.5mph on hard, flat surfaces such as concrete, or up to approximately 5.5mph on rougher terrain, such as grass, and climb inclines of up to 30°. The IGV, when running all systems in parallel, has a reaction time of approximately 100ms, but this can be reduced to 40ms if the 3D map generation is disabled. Between the LRF and camera, it can detect obstacles at a distance of up to 8m with accuracy of up to 1cm. Additionally, between the DGPS and the IMU, it can pinpoint its own position with respect to waypoints to within 10cm. Between the powerful battery packs wired in series and the efficient power distribution method, the IGV has a battery life of over ten hours at rest and over four hours when in use. A complete breakdown of these figures is shown in Table 7.1.

Table 7.1 - RED RAVEN 2.0's Performance

Parameter	**Value**
Top Speed , Concrete	6.5mph
Top Speed, Grass	5.5mph
Maximum Scalable Incline	30°
Reaction Time (without 3D GUI)	40ms
Reaction Time (with 3D GUI)	100ms
Obstacle Distance Detection	8m
Obstacle Distance Accuracy	10cm
Waypoint Accuracy	10cm
Battery Life, Idle	10 hours
Battery Life, Active	4 hours

8. Appendix

Each student has spent approximately 16 hours per week on this project, both in and out of class. Over the forty weeks spent working on this project, each student contributed approximately 640 hours working on the IGV, for a total of over 8,000 hours spent working on this IGV this past year alone, as shown in Table 8.1. This does not include contributions from previous years.

To build RED RAVEN 2.0 from scratch, the financial cost of it would be approximately $92,000, due in large part to the expensive sensors, such as the Inertial Measurement Unit, the Differential Global Positioning System, and the Scanning Laser Rangefinder. Thankfully, due to donations from various companies, particularly Northrop Grumman's incredibly generous donation of allowing us to use their IMU, the cost to CSUN's IGV team has been approximately $27,300 over the past two years, with only $2000 in associated costs this previous year, as shown in Table 8.2. Without donations from the many generous companies and individuals who have helped, the costs would have been far too much for CSUN's IGV team to be able to afford to build such a powerful machine.

Table 8.1 - Time Spent on RED RAVEN 2.0

Hours in class, per week:	8
Hours out of class, per week:	8
Duration of Project, in weeks:	40
Total hours spent, per person:	**640**

Table 8.2 - Total Cost Estimate of RED RAVEN 2.0

Components	Retail Cost	Cost at Time of Purchase	Cost to Team This Year
Hokuyo LRF	$5,000.00	$5,000.00	$900.00
Nuggets	$1,600.00	$1,600.00	$0.00
Motors/Motor Cables	$2,437.00	$2,437.00	$0.00
Clamp	$200.00	$200.00	$0.00
Black/White Pack Batteries	$4,150.00	$4,150.00	$0.00
Toshiba 16" Laptop	$1,500.00	$1,500.00	$0.00
48V/12V DC/DC Converters	$300.00	$300.00	$0.00
Printed Circuit Board	$36.00	$36.00	$36.00
Misc. Electrical Items	$250.00	$250.00	$250.00
IMU	$44,000.00	$0.00	$0.00
GPS Receiver/Antenna	$28,079.00	$8,500.00	$205.00
FOculus Camera	$1,000.00	$430.00	$0.00
Computar Lens/Filter/Mount	$520.00	$255.00	$0.00
Arduino Light Sensor Board	$80.00	$50.00	$50.00
Gearboxes	$1,204.50	$1,204.50	$0.00
Driving Wheel Rims	$215.46	$215.46	$0.00
Driving/Caster Wheels	$135.98	$135.98	$0.00
Metal Materials	$133.16	$133.16	$0.00
Misc. Mechanical Materials	$159.64	$159.64	$0.00
Cooling System	$522.62	$522.62	$243.48
Dampers	$240.56	$240.46	$240.56
Total	**$91,763.92**	**$27,319.82**	**$1,925.04**