

VADER

2023-2024



ABOU-BAKER KADMIRY
AMANDEEP BRAR
CHRIS SMITH
DANIEL KIM
DANIEL VALENZUELA
DENNY FARIAS
ERIC ESPIRITU
HASSIB ABI RAFEH
JESSE CAMPOS
JESSE COMBS
JOSEPH PRINCE
MARGARET GOLDMAN
MELISSA FLORES
QUSAI QANEETAH
RICHARD GILLBERG
RIEL PALIS
RUBEN BERUMEN
TORRENCE PINEAU
WILLIAM PANGESTU
WONKYU WHAN

Faculty Advisor Statement

I certify that the engineering design of the vehicle described in this report, VADER, has been significant and equivalent to each team member earning four semester hours of senior design credit for their work on this project.

Dr. C. T. Lin
*Department of Mechanical Engineering
California State University, Northridge*

1 INTRODUCTION

The Intelligent Ground Vehicle team of the College of Engineering and Computer Science at California State University, Northridge (CSUN) is proud to introduce VADER. Innovations throughout the 2013-2014 academic year have resulted in a remarkable intelligent ground vehicle. VADER features a lightweight and low-profile design with a traditional spring and damper suspension system. Unique obstacle avoidance and “turn around” algorithm as well as several efficient video processing techniques contribute to the impressive autonomous performance of VADER. Innovations to the new platform include a 70% reduced printed circuit board, dynamic light sensor, ghost lines algorithm, as well as a newly created IGV computer simulation. Construction on VADER began during the Fall semester of 2013, and was completed early in the Spring semester of 2014. Progression of the mechanical build for the new VADER platform, from early stage development to the completed design, is shown in Figure 1.

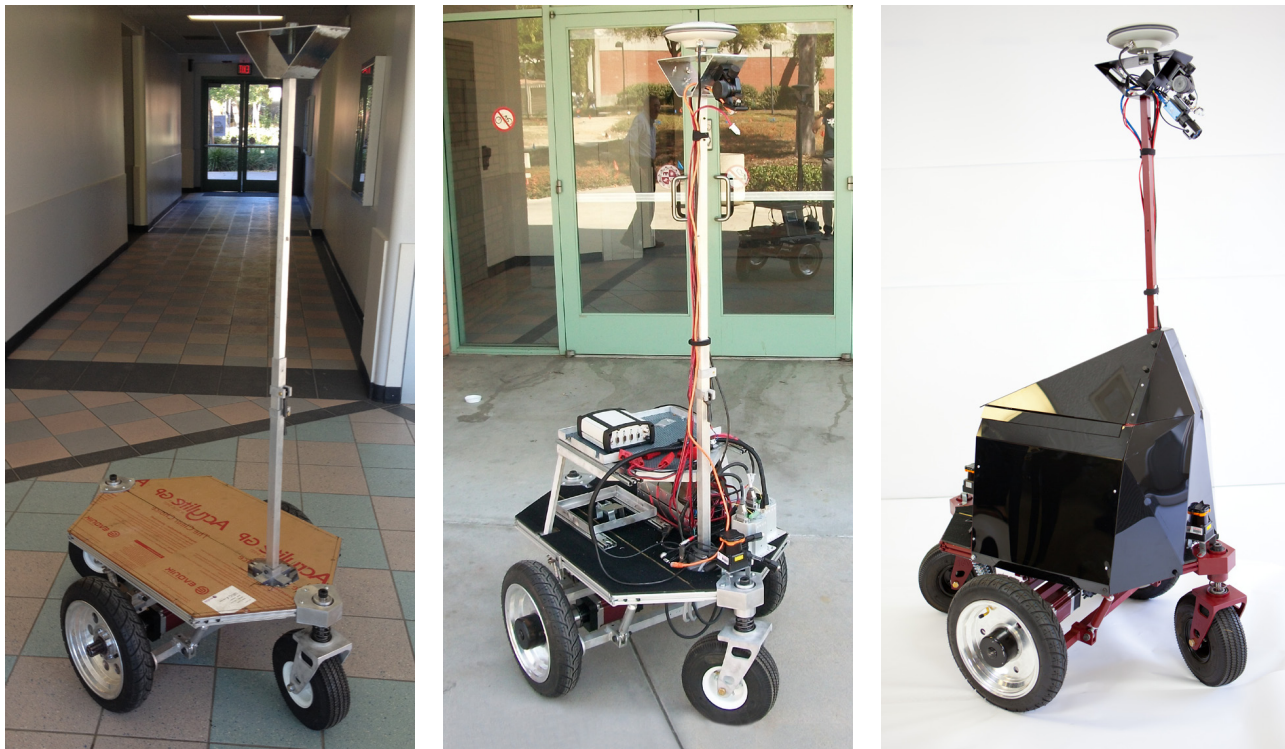


Figure 1: Progression of the mechanical build for the new VADER robot.

2 ORGANIZATION

The intelligent ground vehicle (IGV) project at CSUN is a senior design experience that progresses over the course of two semesters in the Mechanical Engineering Department. Overall group leadership is decided democratically and includes a project manager, a technical manager and a treasurer. Team leads are responsible for organizing team meetings and events, as well as keeping open communication between the six sub-system groups. The six sub-system groups are: Mechanical, Power, Cognition, Vision, Navigation, and JAUS. Within each of the subsystems is a group lead indicated in red in Figure 2. Each of the group leads is responsible for delegating tasks to their group members and enforcing open communication with other groups.

3 MECHANICAL

3.1 Overview

The mechanical design of VADER is like the bones of the platform. Design features of VADER include a low-profile, flexible layout with a modular structure over a differential drive motor system. VADER’s frame is composed of

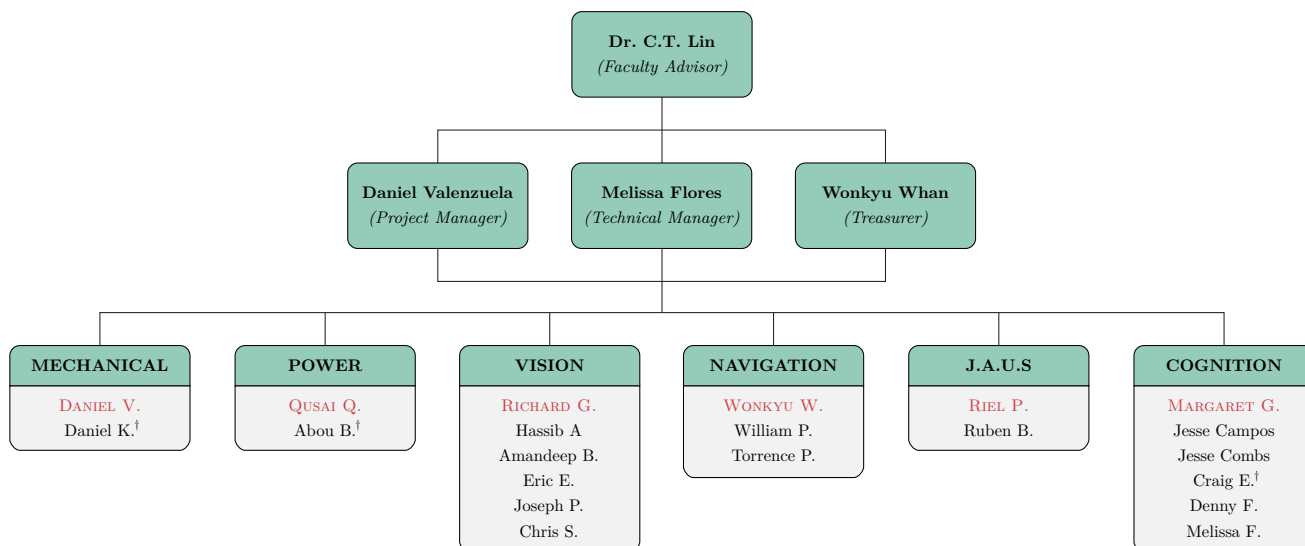


Figure 2: Intelligent Ground Vehicle (IGV) team organization chart for 2013/2014.

80/20 T-slotted aluminum framing which eliminates the need for welding. An advanced spring and damper system supports the completely ridged frame in order to allow the maximum surface area available to mount the hardware for all of VADER's subsystems. Velcro was utilized as a method for securing each component to the floorboard of the robot while allowing for a dynamic layout. The camera mast was designed with a telescopic movement to allow the robot to fit within a smaller total volume.

3.2 Frame Design ‡

The frame for the new IGV platform is perhaps the most important part of the entire vehicle. Made with extruded aluminum 80/20 framing, the frame allows for an extremely sturdy base for the lower suspension to connect to, and for all the components above it to be supported, as shown in Figure 3b. The biggest benefit to using the 80/20 framing is the fact that no welding is required in order to assemble the frame itself and anything that attaches to it. The 80/20 aluminum extrusion members have a specific cross-section which not only improves its strength over standard square aluminum tubing, but ensures that attached parts remain secure. This specialized-cross section also allows for the ability to attach parts to any face on the extruded aluminum using the t-slot.

In order to construct the frame and attach brackets to it, a series of specialized nuts and bolts were used. Attaching components to the aluminum extrusions required the use of three major parts: a 1/4 - 20 × 3/8" bolt, a bracket (or the component itself), and a t-nut. An example of how these parts work together to secure a part to the aluminum is shown in Figure 3a. Parts that were attached to the frame itself include the floor boards, laptop tray frame, fish plate, solid strut brackets, spring brackets, and the caster housing. All of the aforementioned parts were attached using the same methods described above.

3.3 Suspension ‡

The suspension is broken into two main components; the lower suspension and the caster wheel suspension. The lower suspension consists of two front ridged struts, and two rear spring and damper shocks. These members prevent unwanted x-axis (i.e., front-to-rear) sway, and dampen input forces from the drive wheels. A fifth middle rigid strut was designed to prevent unwanted y-axis (i.e., side-to-side) sway. These five members are the connection points to the upper body frame, as shown in Figure 3.

† Indicates alumni members.

‡ Indicates section(s) containing new innovations.

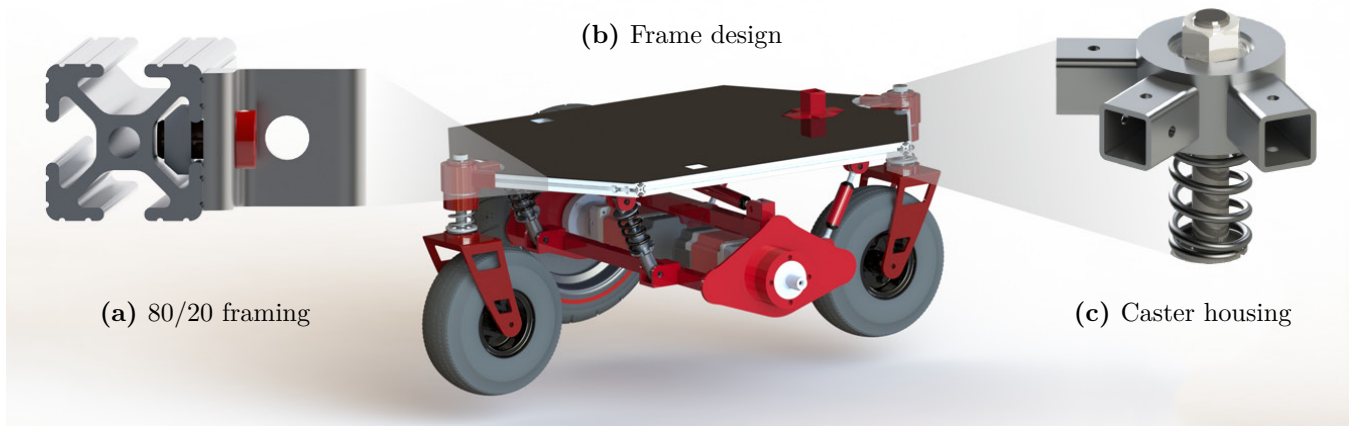


Figure 3: VADER's new platform built with 80/20 slotted framing utilizing a new caster suspension system.

The caster wheel suspension is a new innovative design. The idea to incorporate suspension into the caster wheel housing blossomed into multiple benefits throughout the rest of the overall design concept. This feature gave a lower mounting surface for the front Laser Range Finder (LRF), doubled the surface area to mount components on the body frame, and kept true to the original low-profile design. The caster housings are the connection points for the triangular front and rear 80/20 aluminum framing, as shown in Figure 3c. The caster housing allows for rotational movement of the caster wheel. The new spring-incorporated design allows for z-axis (i.e., vertical) translation.

3.4 Telescoping Mast ‡

Development of the camera mast presented many design challenges. The camera mast needed to have a telescoping design with camera mount, as the sliding portion of the mast needed to be easily removable for transportation. Features of the mast assembly include two locking levers and clevis pin, a camera/antenna mount, and a fish plate to attach the base of the mast to the frame, as shown in Figure 4.

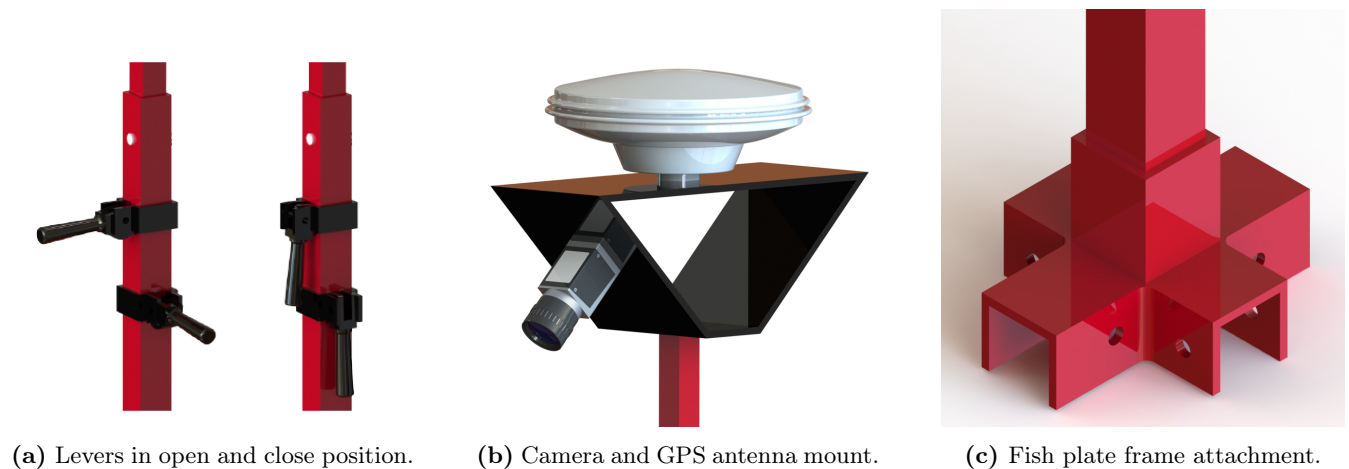


Figure 4: Mechanical components used for the telescoping mast.

Using two levers with a cam design, the movement in the mast could be adjusted. In order to clamp the inner telescoping mast to the outer stationary mast, the levers are lowered until it is parallel with the member it is mounted to. Being a cam design, the lever clamps the inner portion of the mast tighter the more it closes. The orientation of both levers ensures that there is no lateral movement in any direction. Minimizing movement in the mast provides a more stable image for the camera.

The fish plate was designed to take advantage of the 80/20 framing and be secured by a total of three different members and eight bolts. In order to minimize the deflection in the telescoping mast, the fish plate base needed to

‡ Indicates section(s) containing new innovations.

be well-secured. The stationary portion of the telescoping mast carries two important design features on it; a carbon steel clevis pin, and two cam design locking levers. The carbon steel clevis pin is designed to hold the weight of all the components that are held above the mast. It is mounted through both the stationary and telescoping portions of the mast. Although the weight of the components above the mast is held by the clevis pin, there was still a considerable amount of movement in the telescoping portion of the mast. In order to ensure that the moving portion would remain still, it needed to be secured to the stationary member of the mast through two cam-locking levers.

3.5 Gearboxes & Motors

The gearbox and motors are in line with each other and are located at the bottom of the frame. Each gearbox shaft connects to the drive wheels of robot and are held in place by the engaging hub mount. This layout is to provide a differential drive to VADER which allows the motors to react independently from each other making it a very maneuverable platform.

A torque analysis was performed in order to determine the amount of torque that is required to allow the vehicle to move without straining the motors. Calculations were performed using the assumptions provided in Table 1.

Table 1: Parameters used for torque analysis of the gearboxes and motors.

Parameter	Value
Weight of vehicle	160 lb
Coefficient of friction	0.35
Number of motors	2
Wheel diameter	14 in.

Newton's second law of motion along with a free body diagram was needed in order to calculate the forces. When calculating the torque required for VADER, a worst-case calculation was conducted; that is, the analysis of the robot on an inclined plane.

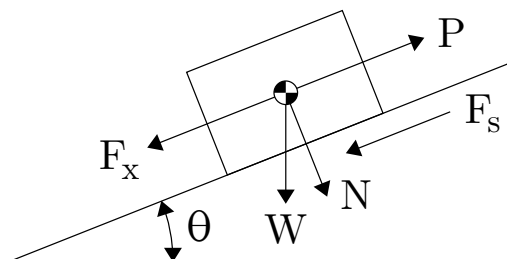


Figure 5: Free body diagram used for torque analysis of the gearboxes and motors.

The IGVC rules state that there could be a maximum angle of incline of 13.5° . Taking this into consideration, a free body diagram was used to derive the corresponding force equations, as shown in Figure 5. Using the known angle of incline, θ , weight of the robot, and the coefficient of friction, the forces and torque in question were calculated. The calculated values ensured that the selected motors would be sufficient for VADER to successfully navigate through all field conditions.

4 POWER

4.1 Overview

The purpose of the power system on VADER is to distribute the correct amount of electrical power to each component on the vehicle. Power requirements for each sub-system group within the intelligent ground vehicle project were attained by communicating with each of the groups. It was determined that the vision, cognition and motion control

groups would use a shared power source, while the navigation would be provided with an external power source separated from the others. A full power distribution map is shown in Figure 6.

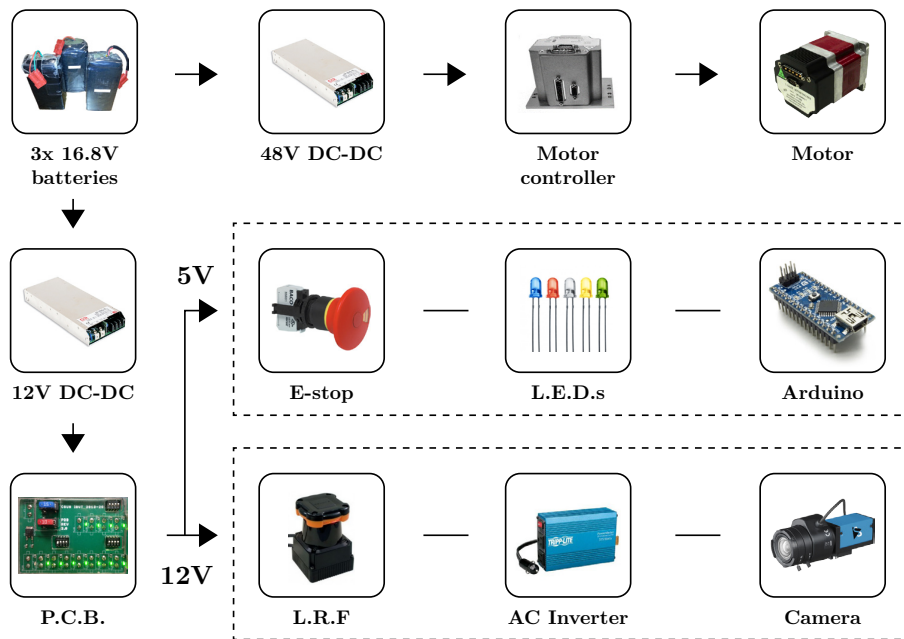


Figure 6: Power distribution map with indicated voltage rails.

4.2 Batteries

The designed battery life of VADER during operation is 3 - 5 hours depending on the carried load. In order to be able to sustain this battery life while testing, both normal and extreme loading conditions were calculated. The motors on VADER draw the largest amount of power, therefore the power draw of the motors were adjusted to attain the desired outputs. A velocity of 1 mph was used for normal conditions, while a velocity of 6.5 mph was used for extreme conditions.

Table 2: Power consumption for normal and extreme conditions in Watts.

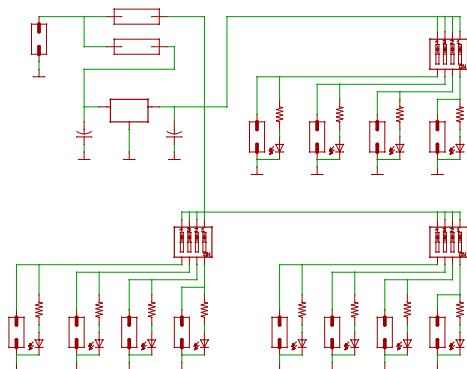
Type of Load	Normal Load	Extreme Load
Total base load	177	177
Transient motor load	154	910
Total load	331	1087
Total at 88% DC/DC efficiency	376	1235

Based on the loading conditions provided under extreme conditions, it was determined that VADER needs at least 1235 W, as provided in Table 2. The optimal battery option to supply the required power for VADER were Lithium Polymer (LiPo) batteries. Each LiPo battery has a nominal voltage of 14.8V and a 30A output. The rated power output of each battery is 444.4W. Three of these batteries in series provide the power and voltage output required while limiting the overall current that each battery can provide. Battery voltage is converted and regulated by two separate DC-to-DC converters: 48V DC output for the motors and 12V DC output for all other electrical components. This provides electrical isolation between the motors and sensitive electrical sensors, such as the laser range finder. This also allowed the motors to function at their ideal operating voltage of 48V DC.

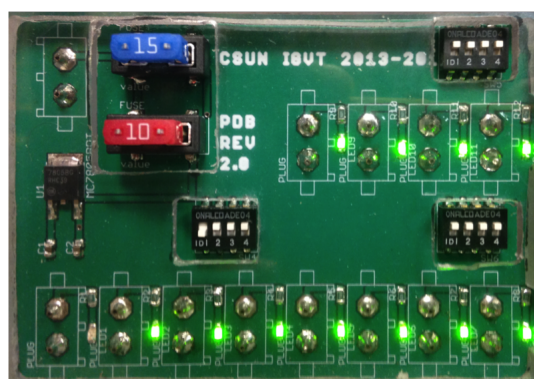
4.3 Power Distribution Board ‡

The power system integration was accomplished using a centralized control system, constructed on a printed circuit board (PCB). The top-level objective in the PCB was to come up with a board design that would occupy less area. The approach was to incorporate surface mount components and keep the 48V rail, which is needed for the motors, separate from the board. A relatively minimalistic design was implemented using EAGLE CAD to produce the required Gerber files, as shown in Figure 7a.

The innovation in the board design that saved a large amount of space was the fuse layout. Instead of running a single fuse in series with each component, a single fuse is used for the 12V and 5V rails. A switch was dedicated for each component, which was to remain on unless in the event of a short circuit. If this occurred, all switches would be opened, the fuses replaced, and each switch turned on one-by-one until the source of the short circuit was determined. Reducing the number of fuses present on the board, along with the removal of the 48V traces and a section for the voltmeter, allowed for an overall reduction in area of 70%, as shown in Figure 7b. The switches to send power to the board were designed on a separate acrylic board, along with the switch connecting the 48V to the DC motors.



(a) EAGLE CAD circuit schematic.



(b) Actual power distribution board on the robot.

Figure 7: New power distribution board with 70% reduction in area.

4.4 Emergency Stop

The emergency stop system on VADER functions by making use of the motor controllers' built-in stop conditions to bring the robot to a safe and controlled stop. When the emergency stop is not activated, a logic LOW (i.e., ground) is sent to the motor controllers. When the emergency stop is activated, a logic HIGH (+5V) is sent to the motor controllers' input/output pins. A series configuration is used for the push-button stop and the wireless remote stop to eliminate any conflict between the two.

4.5 Hot-Swap ‡

The hot-swap system is an innovation that was implemented in order to reduce down-time of the vehicle while testing and at competition. Hot-swapping allows the robot to have the ability to plug in fully-charged batteries to the system before removing the depleted ones. This method of switching batteries makes it possible to keep all of the components on the robot powered on, thus eliminating the need to re-initialize all of the hardware and software.

5 VISION

5.1 Overview

Vision is “the eyes” of the robot, and is responsible for providing a map of all the obstacles and white lines in the surrounding area. Image data is first acquired through an Imaging Source industrial camera paired with a Theia wide-angled lens. It is then filtered through various processes, such as color plane extractions, thresholding, and convolution, and then passed through a series of particle filtering techniques to remove noise and unwanted small

‡ Indicates section(s) containing new innovations.

bits of information. Finally, it is passed to cognition in the form of an obstacle polar plot at which time intelligent decisions are made about forecasted path-planning.

5.2 Convolution Edge Detection ‡

The process of detecting obstacle edges (i.e., white lines, barrels, or flags) involves a hierarchy of layered filterization techniques, in which images are overlaid on one another to expose similarities and differences. To further improve this adaptive process, a new mathematical method was implemented: convolution.

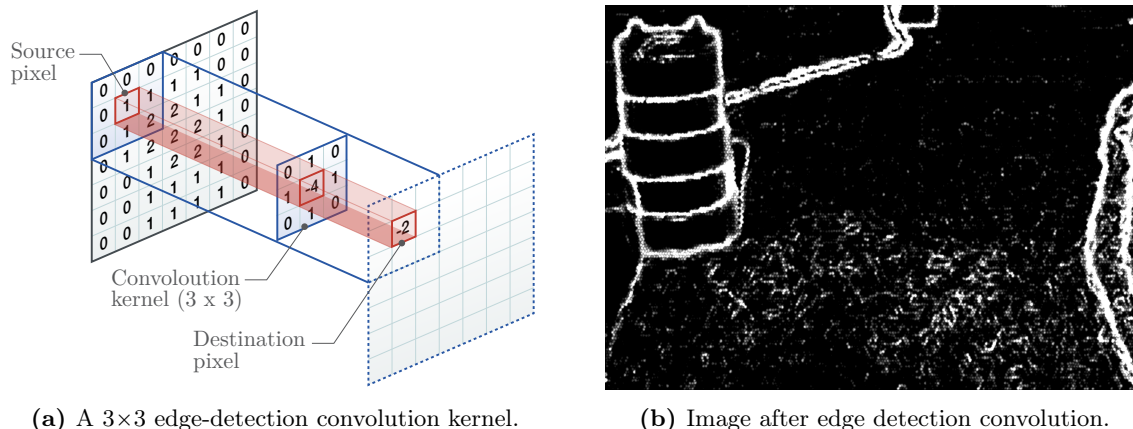


Figure 8: Example of edge detection convolution.

Simply put, convolution is a feature extracting process of multiplying a predefined, constant 3×3 matrix (see Figure 8a) across an entire image, and replacing the middle pixel (i.e., the source pixel) with the summation of the product of the image pixels and convolution kernel entries. In doing so, information about neighboring pixels in relation to the source pixel is obtained, and differences in pixel intensity (e.g., from light to dark or vice versa) are heightened. The end result of applying a convolution kernel to the source image shown in Figure 9a, where the edges of the white lines and barrels are present, is shown in Figure 8b.

5.3 Closest-Edge Algorithm ‡

In past years, the process of determining positions of filtered obstacles was extremely cumbersome, and involved scanning every pixel in the image. In order to improve on this antiquated notion and to reduce processing time, a new algorithm was developed, taking advantage of LabVIEW's IMAQ EDGE TOOL.

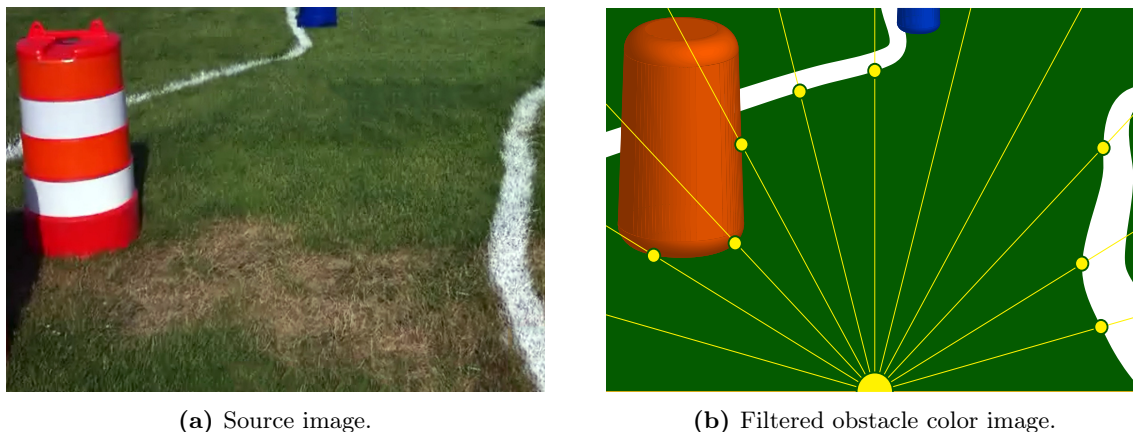


Figure 9: Example of closest edge algorithm.

‡ Indicates section(s) containing new innovations.

From the robot's position, centered horizontally on the image and at the bottom, lines are extended until they either contact an obstacle (i.e., white line, barrel, or flag) or reach the edge of the image (see Figure 9b). If an obstacle's edge is found, the algorithm halts processing at that angle, stores the coordinate location of the obstacle edge, and proceeds to the next incremental line. The process uses one-degree steps across the entire image, and produces an array of only 181 values (i.e., $0^\circ \rightarrow 180^\circ$) corresponding to the closest obstacle encountered at each angle.

5.4 Light Sensory ‡

Anticipating change in weather conditions (e.g., relative brightness) was a sensory function overlooked in past years of the intelligent ground vehicle. The importance of this sensory function is vital in the robot's ability to adapt to varying lighting conditions, as a cloud cover or extremely bright day can wash out the ability to decipher white lines from dead grass. In order to combat this issue, a light sensor was implemented into the robot, as shown in Figure 10a.

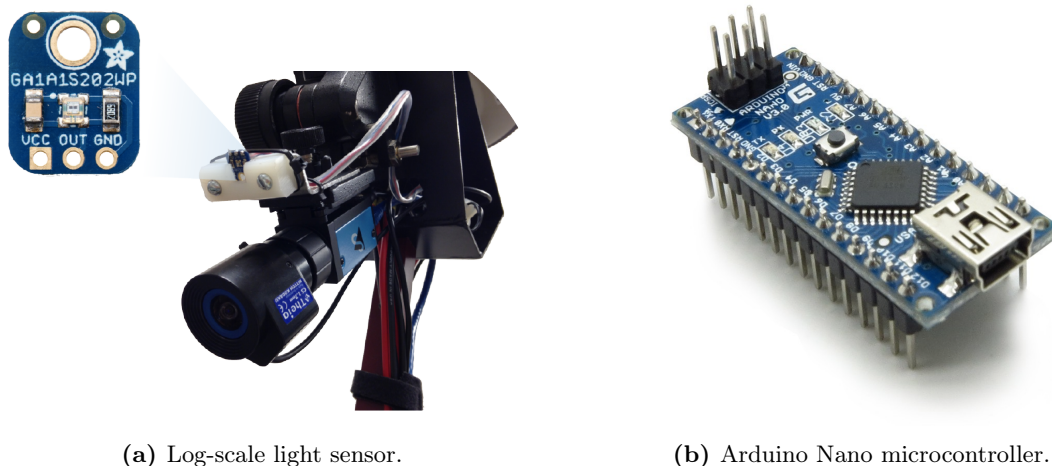


Figure 10: Hardware components used for vision acquisition.

For A/D conversion of the light sensor signal, an Arduino Nano microcontroller was used, as shown in Figure 10b. By conducting various field tests, a set of ideal gain values corresponding to relative brightness (i.e., voltage) were calibrated for perfectly harmonious image noise filtering.

6 NAVIGATION

6.1 Overview

Navigation is responsible for gathering and extracting information from the compass and the GPS antenna. The shortest distance and heading direction to the waypoints are computed, and then sent to cognition. The navigation system utilizes a NovAtel Pinwheel GPS antenna, a NovAtel Propak V3 Receiver, and a FieldForce Trax AHRS Module (Compass), as shown in Figure 11.



Figure 11: Navigation flowchart [from left to right: GPS antenna, GPS receiver, computer, compass]

‡ Indicates section(s) containing new innovations.

6.2 Intermediate Waypoints ‡

Navigation is now implementing a new program, called Intermediate Waypoints, which will allow the vehicle to map out the course after an initial run through the obstacles. The algorithm will allow longitude and latitude information to be recorded. The waypoints that have been recorded will go through a simple process to refine the information and apply it to the following attempts through the course to increase its path efficiency.

6.2.1 Turning Method

The turning method is a process that analyzes the vehicles current heading and its previous heading. The difference in these compass values allows the turning method to understand that the vehicle has made a great change in its heading due to an obstacle or a white line. This will then record the latitude and longitude values that are related to this heading change, as shown in Figure 12.

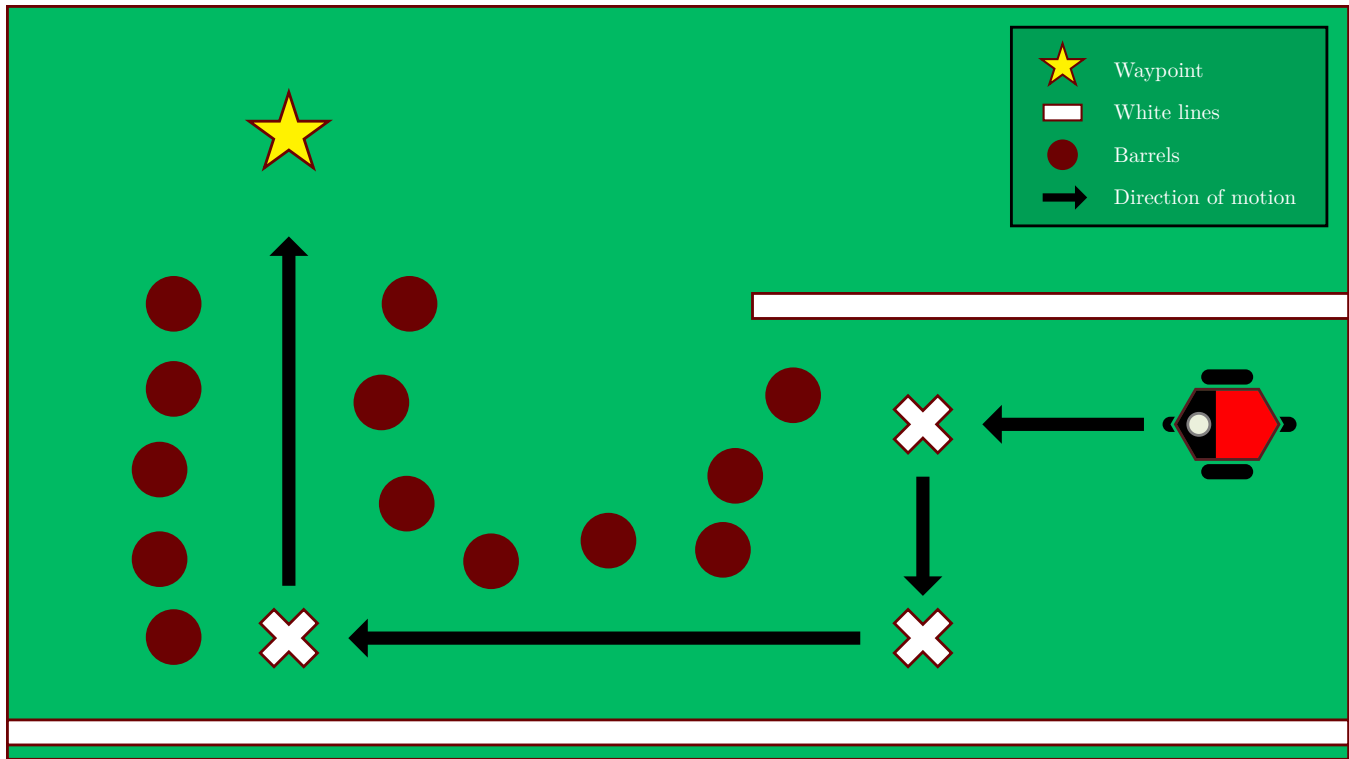


Figure 12: Illustration of turning method.

6.2.2 Point Averaging Method

The point averaging method is a process that eliminates repeated waypoints or multiple waypoints in a designated area. Due to the random nature of the obstacles of the course, the vehicle may rotate 360° in a small area. Through the logic of the turning method, multiple waypoints will be recorded due to the rotation. The point averaging method will take the waypoints in one designated area, and average the longitude and the latitude values into one waypoint, as shown in Figure 13.

6.2.3 Intermediate Waypoints Threshold

The intermediate waypoints that will be used to help guide the vehicle will also have a threshold applied to them. This threshold will create a radius around each point and create an area that will allow the robot to reach each waypoint without returning to the exact location of the waypoint. This will allow a more fluid like path for the vehicle, as shown in Figure 13.

‡ Indicates section(s) containing new innovations.

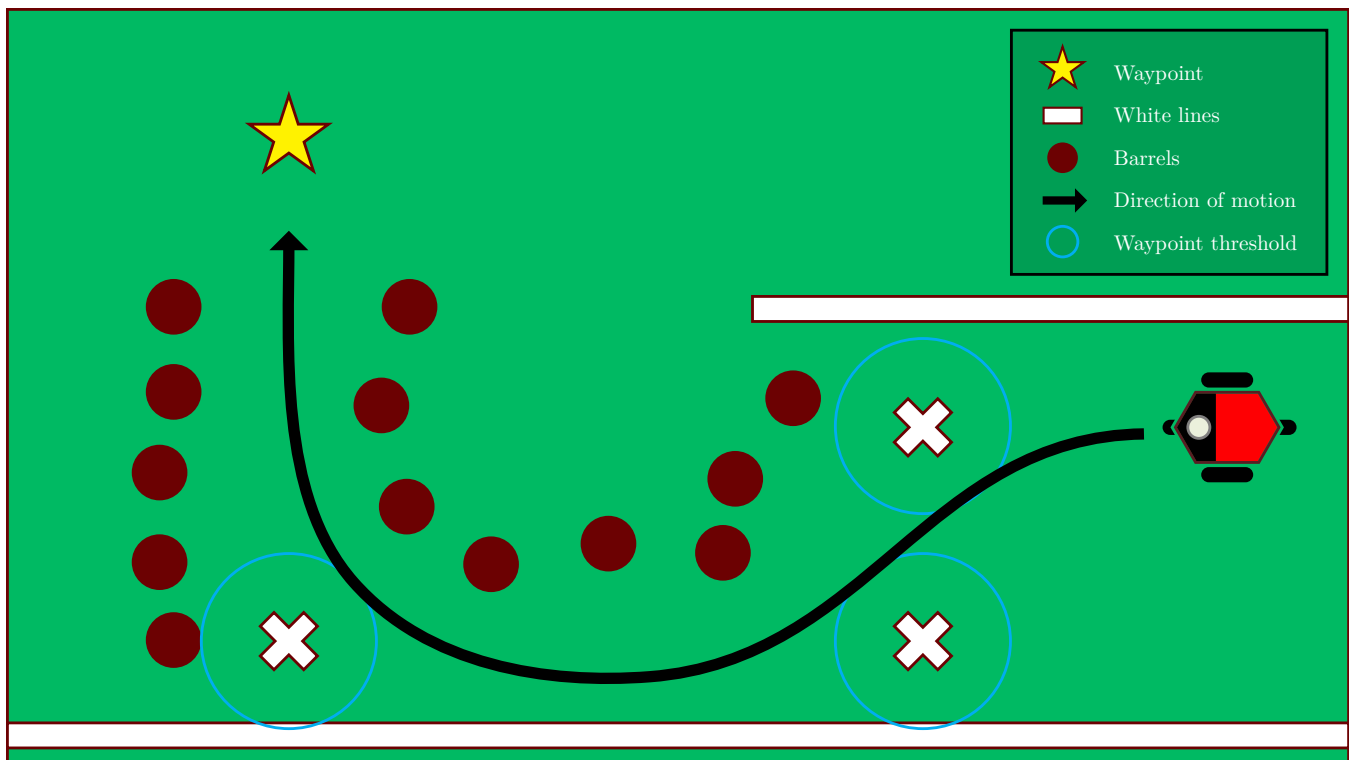


Figure 13: Illustration of intermediate waypoints with applied thresholds.

6.3 Flag Triggering ‡

The navigation system also has incorporated an algorithm that will tell the vision systems to activate their flag detection program. This system is designed to record the starting waypoint and activate the check once the vehicle has gone through all the mandatory waypoints and returned to a specific distance of the first waypoint.

7 IOP/JAUS

7.1 Overview

For the interoperability profiles (IOPs) challenge, VADER adopted the Joint Architecture for Unmanned Systems (JAUS) Toolset which is a free software designed to support the modeling, analyzing, specifying, implementing, simulating, and testing of protocols for distributed systems. JAUS Toolset (JTS) provides a graphical user interface (GUI) for developers to design JAUS compliant software. It also has the capability to generate specification documents and source code based on the service design. The generated code — available in C++, C#, or Java — allows the developers to further define and integrate their software into the design service architecture. JTS also enables validation of the user designs automatically within its GUI, to ensure the created design will be interoperable and produce the desired results. JTS also provides an abstraction of the functionality of AS5669A transport layer definition and supports the JAUS Router Middleware, a AS5669 compliant software, for management and routing of network traffic in distributed systems.

7.2 Protocol State Machine

The robotic protocols and service behavior is described as a finite state machine where the robot can be in only one particular state at a time, as shown in Figure 14. The letters A, B, and C represent functions that transition the machine from one state to another, characterized by a combination of inputs with particular outputs. The state machines that were created for the service protocols and behaviors conform to the AS5710 JAUS Core Service Set standards. JTS is able to create services using state machines, defining the triggers and actions of each state, which can be translated one-to-one to C++, C#, or Java code for further development.

‡ Indicates section(s) containing new innovations.

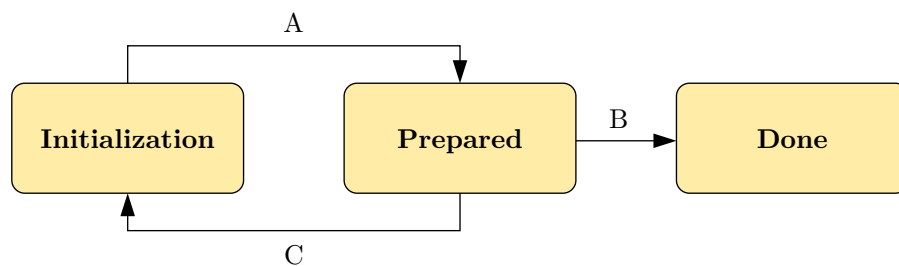


Figure 14: Protocol state machine flowchart.

In previous years, implementation of JAUS was utilized using visual programming languages, such as LabVIEW, which presented a number of disadvantages and advantages in terms of designing the JAUS software. Since the JAUS software is self-contained within LabVIEW, this negates possible resource consumption issues when compared to a more modular approach. The software can also be operated within the main program allowing for simplified debugging and modifying. However, while following this particular approach the required standards present a more challenging and tedious design process that proved to be time-consuming; evidence seen by an incomplete showing at the previous competition.

7.3 JAUS Toolset (JTS) ‡

This year, the development team took full advantage of JTS, which provides a GUI-based environment for modeling, analyzing and implementing AS5684 based systems, as shown in Figure 15. As an advantage, JTS provides compliant constructs for building our system from the ground-up. This modular design allows the development team to modify and test the JAUS software without the need to interfere with the programming of the robot.

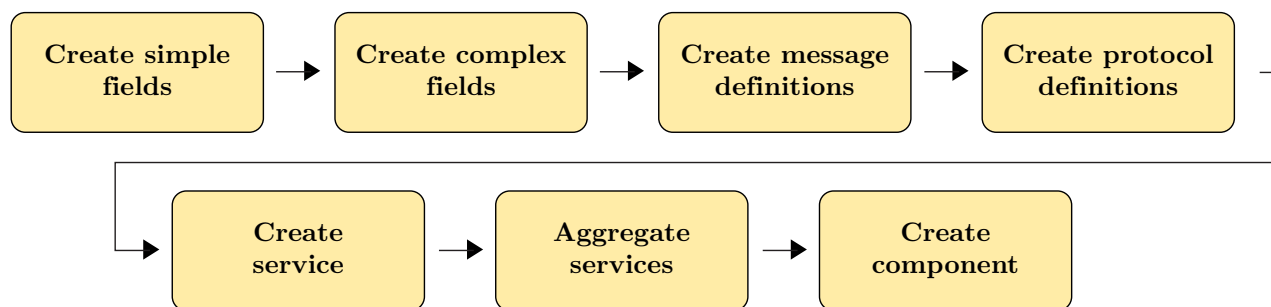


Figure 15: JAUS Toolset flowchart.

In previous years, it was chosen to contain JAUS within LabVIEW to make integration simpler by having these services communicate directly with the vehicle's programming. This is a less than ideal approach as the JTS builds these services to standard, resulting in code that is auto-generated. In order to avoid this, a few methods were considered when integrating the JTS generated code, such as shared memory and network variables. Inter-process communications can be achieved with shared memory techniques which is an efficient means of passing data between two systems sharing the same resource. To further improve on the process, National Instruments provides network variables for developers using the LabVIEW software to simplify the data exchange process.

8 COGNITION

8.1 Overview

The cognition and motion control systems of VADER are responsible for calculating the successive optimal paths to enable the IGV to navigate through the obstacle course autonomously. These algorithms utilize maps of the local

‡ Indicates section(s) containing new innovations.

environment in the form of polar plots, provided by the vision systems and a Laser Range Finder (LRF). The LRF is used to scan a 180° area at a scanning rate of 25 milliseconds and a resolution of 1° . The range of the LRF is automatically adjusted based on the proximity and configuration of obstacles at any one time. The algorithms used by VADER are based on those used in previous years, with the addition of many new improvements. These improvements include adding an LRF to the rear of the vehicle for obstacle avoidance in the reverse direction and a new algorithm for navigating through the flag section called ghost lines. Furthermore, a simulation system was developed in order to test software when the platform is unavailable.

8.2 Radial Polar Plot (RPP)

The path planning and obstacle avoidance algorithm used by VADER is referred to as the Radial Polar Plot (RPP). This algorithm's main functionality is to determine successive safe paths that lead the robot to the next navigational waypoint. The RPP process begins by receiving data in the form polar plots from the LRF and the vision systems. These plots provide the locations of any obstacles, white lines, and flags within the robot's view. RPP combines these polar plots to generate a local map which is then fed through a number of processes in order to identify the optimal course, as shown in Figure 16.

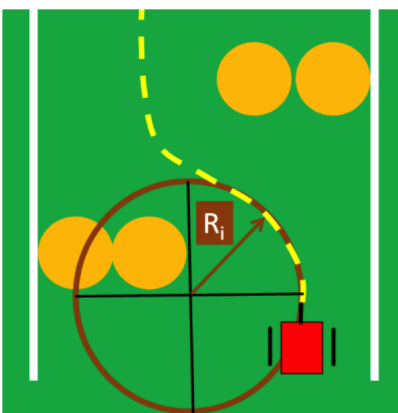


Figure 16: Illustration of radial polar plot path planning.

The data is filtered down to a certain range, determined intelligently, to preserve only the most critical information for processing. The obstacles in the data are then grouped together and open paths large enough for the IGV are identified based on many complex criteria. The program then identifies the best opening based on factors that include navigational heading, obstacle orientation, and the current location of the IGV. Taking into account the dimensions of the vehicle, the left and right most radial paths within the selected opening are calculated (see Figure 16). Finally, an optimization function selects one radius within these radial boundaries and outputs it to motion control as the desired turning radius.

In order to further optimize the motion of the IGV, a dynamic acceleration function executes in parallel with RPP. Based on the current location and speed of the vehicle and the desired future location and speed, the angular accelerations of the vehicle's wheels are calculated. As such, each full cycle of RPP determines two radial paths — one current and one future path — allowing it to transition smoothly between each successive path.

8.3 Motion Control

Motion control involves communication between the motors and the laptop in order to output the correct voltage, as shown in Figure 17. The hardware consists of two motors and two controllers, known as silver nuggets. The controllers regulate the amount of voltage and current that is allowed to the motors in order to initiate motion. Motion control receives the desired radius from radial path optimization and the velocity that is desired from the velocity function to calculate the required wheel angular velocity. The dimensions of the wheels, gear ratio, desired angular velocity, and linear velocity are used in order to calculate the left and right motor speeds and convert it to revolutions per minute (RPMs).

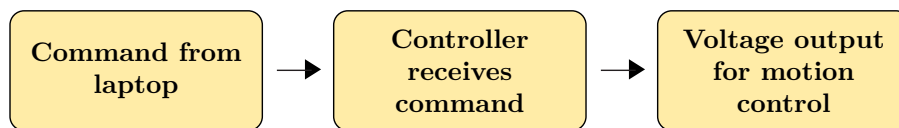


Figure 17: Motion control flowchart.

A velocity function takes into consideration the vehicle's radial path and the presence of obstacles in order to adjust its linear velocity to a desired level. The maximum and minimum velocities of the vehicle are specified according to competition rules and the physical limitations of VADER. An acceleration model evaluates the changes in the desired motor speeds over time and uses the motor's acceleration to predict each wheel's real-time motor RPM. Once the predicted motor RPM for each wheel is determined, a forecasted path is output to the local path-planning map.

8.4 Obstacle Path Interference

To ensure that the path generated by RPP is valid, a path checking algorithm known as obstacle path interference (OPI) is used to ensure that the vehicle will not collide with any obstacles. This is accomplished by analyzing the predicted path and comparing it to the current combined plot, which consists of data from the camera and the LRF. If OPI detects that there will be collision or that the vehicle has encountered a dead end, it will force the vehicle into a recovery mode and trigger the backup movement algorithm.

8.5 Backup Algorithm ‡

Due to issues backing up when encountering an obstacle, an algorithm to allow the robot to drive in reverse was avoiding obstacles was implemented. Since the obstacle avoidance algorithm RPP plans a path using data from the LRF mounted to the front of the robot, it can operate in the same manner while traveling in reverse. Thus, a second LRF was mounted to the rear of the robot and the RPP code was modified such that depending on the direction the robot was driving, the data from the appropriate LRF would be used.

While the vehicle is backing up, the motor rotation must be reversed while negated so that it backs up properly. If OPI detects an obstacle, it will force the robot into the recovery mode, which will initiate the backup algorithm for a specified duration of time. After backing up, the vehicle will turn in order to provide a change of scenery for the robot, ensuring that it will choose a different path. Once this sequence is complete, the recovery mode is turned off and the robot resumes driving forward and planning its path accordingly.

8.6 Ghost Lines ‡

The term "ghost lines" refers to a process wherein simulated blockages are intelligently inserted into the robot's sensory data to force it into the desired course through the flag sections. This system functions by first selecting the nearest red flag and the nearest blue flag within the robot's view. The polar angles at which these selected flags lie are then compared to determine which color flag is to the left or right of the other. If the nearest red flag is to the right of the nearest blue flag (the correct path based on competition rules) ghost lines are drawn to effectively "funnel" the robot into this path (see Figure 18a). However, if the nearest selected flags are in the incorrect configuration a ghost line will be drawn to connect the two flags, effectively blocking off that path (see Figure 18b). The robot will then be forced to turn away from that path and approach the other opening, the edges of which will soon become the nearest edges to the robot, and will initiate the funneling performance described previously.

‡ Indicates section(s) containing new innovations.

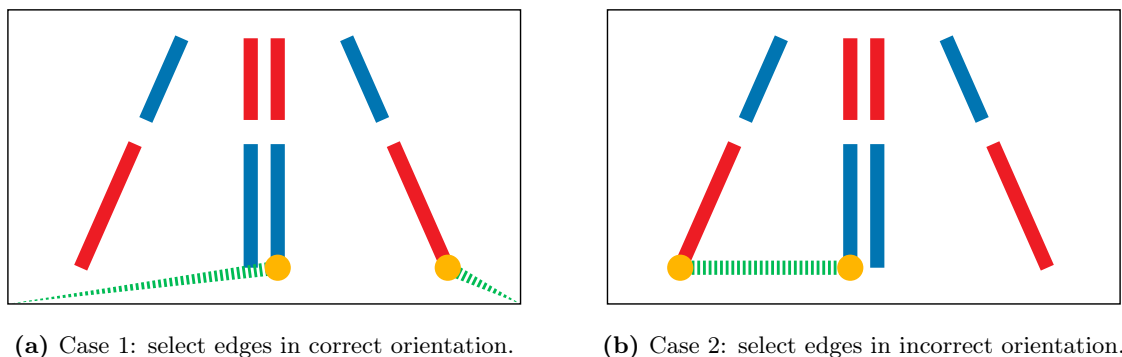


Figure 18: Illustration of ghost lines effect.

9 SYSTEMS INTEGRATION

As previously stated, VADER uses an LRF and camera to detect obstacles and white lines in the course. The vehicle also utilizes a GPS that allows the vehicle to determine its location as well as its heading relative to the waypoints. Data from all of these sensors is collected for cognition and used to determine the most effective radial path to each waypoint. Once cognition determines the best path, the data is sent to motion control for implementation. As the robot moves, all sensors provide feedback to cognition in order to allow for path correction as well as the determination of the appropriate motor acceleration and speed. This process is continuously repeated throughout the course, as shown in Figure 19.

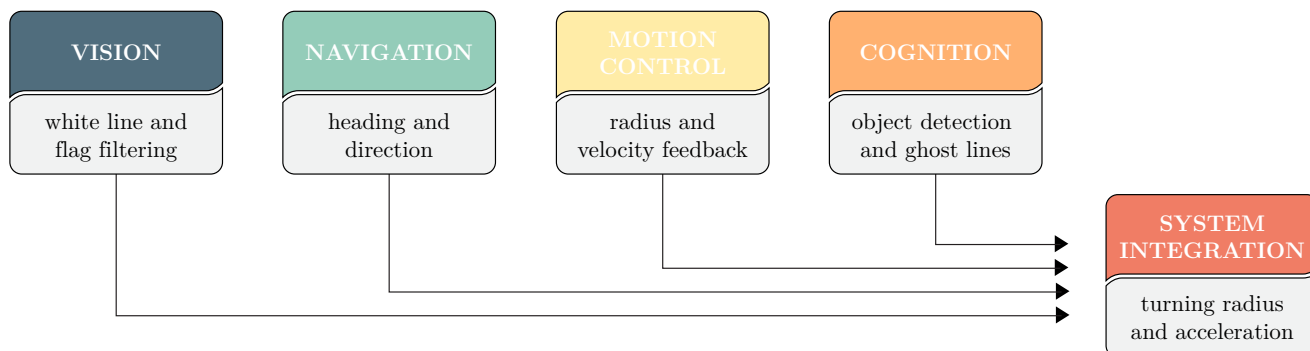


Figure 19: Systems integration flowchart.

10 SIMULATION ‡

10.1 Overview

A simulation was created from scratch using LabVIEW to test developed cognition and navigation code. The purpose of the simulation is not to replace field-testing, but rather to supplement it. The simulation creates virtual LRF and compass data that is passed into the cognition and navigation codes for processing. The simulated IGVC responds to the codes in order to run successfully in autonomous mode as it would during actual operation.

10.2 Simulation Course

The size of the simulated IGVC and the course map must be defined at the beginning of the simulation, as shown in Figure 20. The center of the red circle, which represents the center of the IGVC, is defined at the center of the differential drive. The radius of the simulated IGVC is the distance from the center of the differential drive to the farthest point on the actual IGVC. Measuring to the farthest point ensures that the simulation IGVC won't run into

‡ Indicates section(s) containing new innovations.

the obstacles. The distance of the LRF from the center of the differential drive must also be defined. This ensures that the LRF output graph produced by the simulation displays accurate obstacle distance values. Thus when the virtual LRF data is read in by RPP, they will be able to output accurate motor RPM values that are then used to calculate the simulated IGCV's velocity and rotation rate. The map is composed of lines, circles, and X marks that are read in from an input file. The graphical display, as seen in the Figure 20, is created using LabVIEW's built-in plotting VIs. The orange circles represent barrels, the white lines are like those on the competition course (which the cognition code considers as barriers in the simulation), and the white X marks are waypoints.

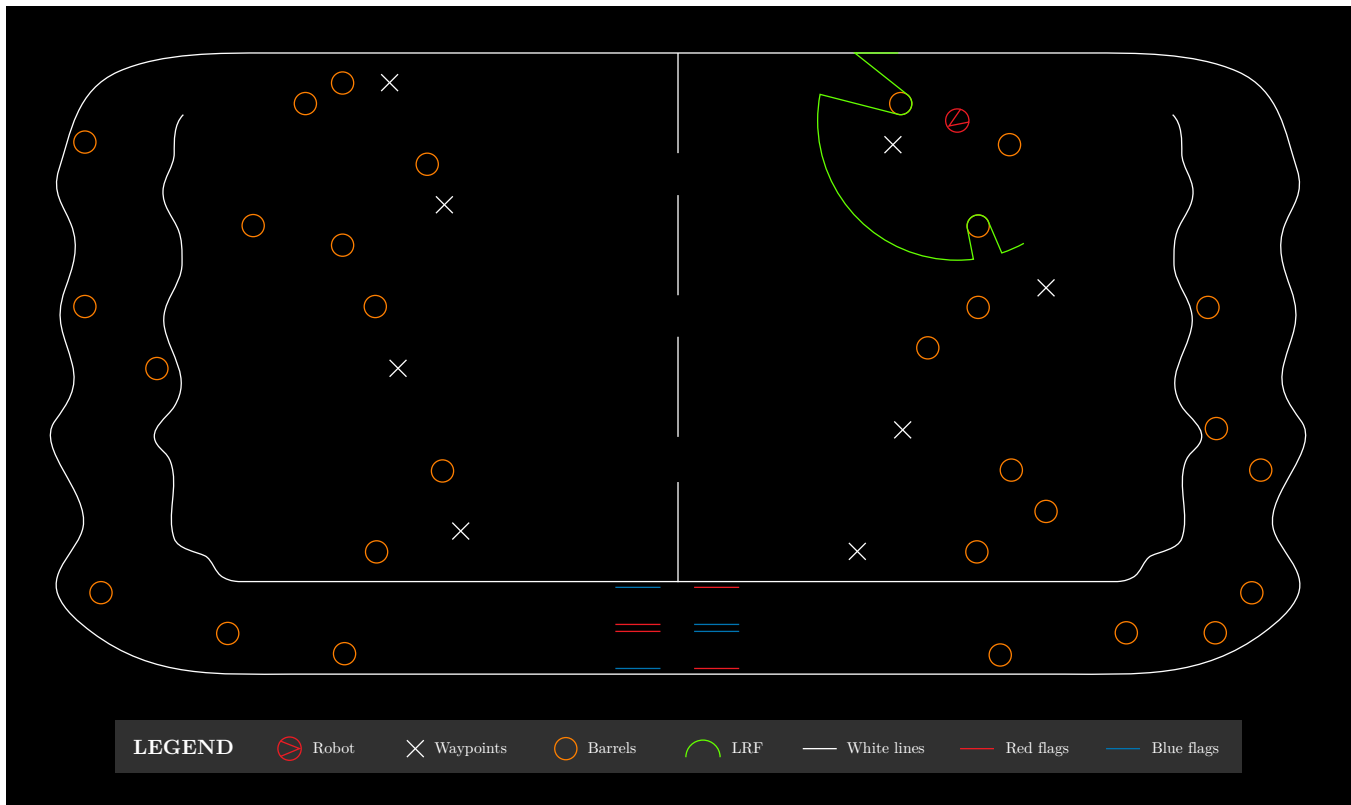


Figure 20: Computer-generated IGVC map used for simulation.

10.3 Cognition

As the simulation runs, virtual LRF data is created by the simulation based on the defined course map's obstacles. The virtual LRF's range is shown as the green line in front of the simulated IGCV, and it is distorted as it encounters obstacles, indicating their distances from the IGCV. These distance readings are inserted into an array that simulates the LRF output. This array of simulated LRF data is written to a global variable called LRF plot that is then sent to the RPP code for processing. After RPP reads in the virtual LRF data and responds to the information, it calculates appropriate rotational speeds for the left and right motors of the IGCV and then writes these values to their respective global variables. These global variables are read into the simulation, where calculations are performed on these values to determine the velocity and the overall rotation rate of the simulated IGCV.

10.4 Navigation

Simultaneously, the waypoints and the instantaneous position of the simulated IGCV on the virtual competition course are converted from x and y positions to latitude and longitude values. These values are used to compute virtual compass data, and then the latitude, longitude, and compass data are written to their respective navigation global variables and processed by the navigation code. The navigation code proceeds to behave as it would in field-testing, as it communicates with cognition on which direction the IGCV should head. In this way, the simulation provides an alternate way to test cognition and navigation codes in autonomous mode.

11 OVERALL SYSTEM PERFORMANCE

Some key performance attributes of VADER can be seen in Table 3. A lightweight, low-profile, design coupled with the use of three LiPo batteries in series allows VADER to maintain a top speed of 6.5 mph for approximately 3 hours. It takes VADER an average of 82 ms from the time it detects an object to the time when the final path-planned decision is made. Under normal operating conditions, battery life is roughly 6 hours. Both the front and rear LRF detect obstacles up to 8 meters away. Vision utilizes the camera to detect obstacles and lines up to 7 meters away. Lastly, through the use of a differential GPS system (DGPS), the waypoint resolution is approximately 30 cm.

Table 3: Overall system performance parameters.

Parameter	Value
Top speed	6.5 mph
Reaction time	82 ms
Battery life	3 - 6 hrs.
Obstacle detection distance	8 m
Waypoint resolution	30 cm

12 TOTAL COST ESTIMATE

The total retail cost of all the materials and components utilized on VADER is approximately \$26,000. The most expensive components contributing to the cost were the GPS antenna, the GPS receiver, as well as the LRFs. This year, with the exception of one of the LRFs, most of the components were purchased as new items, so the total cost to the team in constructing VADER was approximately \$20,000, as provided in Table 4. In addition to the cost of parts, each of the students spent approximately 20 hours per week on the IGV project. Thus, during the 2013-2014 academic year, each of the students spent approximately 800 total hours working on VADER.

Table 4: Total cost estimate for 2013/14.

Group	Item	Cost
Mechanical	Motors	\$2,250
	Motor controllers	\$1,030
	Gear boxes	\$2,000
	80/20 aluminum	\$60
	Lower chassis	\$250
	De-couplers	\$60
	Casters	\$100
	Wheels	\$200
Cognition	Front LRF	\$6,000
	Rear LRF ¹	\$6,000
Navigation	GPS antenna	\$2,000
	GPS receiver	\$3,500
Vision	Imaging Source camera	\$510
	Theia lens	\$250
	Arduino Nano	\$30
Power	DC-DC converters	\$510
	PCB	\$100
	Miscellaneous parts	\$50
	LiPo batteries	\$600
TOTAL		\$19,650

¹ Indicates re-used item not included in the total cost.