# LINJA


**Team Members**

Alex Anikstein

Raffi Dersarkissian

Benjamin Mashian

Harout Markarian

Edder Rivera

Sevana Avanessian

Kevin Thompson

Andrew Sudono

Alfie Gil

Po-Jen Wang

**Faculty Advisor Statement**

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

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 LINJA. LINJA is a new and innovative IGV that was designed, built, and programmed during the 2011-2012 academic year at CSUN. Many of the innovations of LINJA were inspired by the success of the previous year's IGV platform, Red RAVEN, including a flexible frame, drive wheel decouplers, a custom printed circuit board, a new obstacle and white-line detection algorithm, and a radial path planning algorithm.

## 1.1 Team Organization

The IGV project at CSUN is a senior design course that is associated with the Mechanical Engineering Department and is completed over the course of two semesters. It consists of five sub groups: Mechanical, Electrical/Power, Cognition/Motion Control, Vision, and Navigation/JAUS (Figure 1.1). Each sub group has a leader that arranges the organization of the group and enforces open communication with the other groups so that proper integration is achieved. In addition, three team members are nominated to be a project manager, treasurer, and secretary of the IGV project. These three members are responsible for organizing team meetings, promotional events, registration forms, and finances.

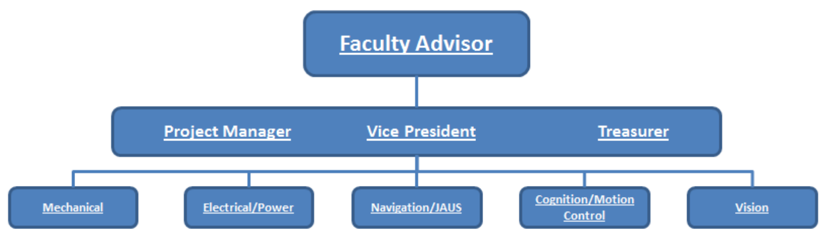


Figure 1.1 – CSUN IGV's Team Organization

## 1.2 Overall System Integration

Utilizing a Dual-Camera configuration, Laser Range Finder (LRF), GPS, and compass, LINJA scans its immediate environment and sends the data to Cognition (Figure 1.2). Cognition integrates the sensory data and evaluates an instantaneous turning radius. This information is sent Motion Control to evaluate the required angular velocity and acceleration of the IGV's motors. Using the on board Li-Po batteries and mechanical drive wheels, the IGV navigates itself to a new location in the environment.

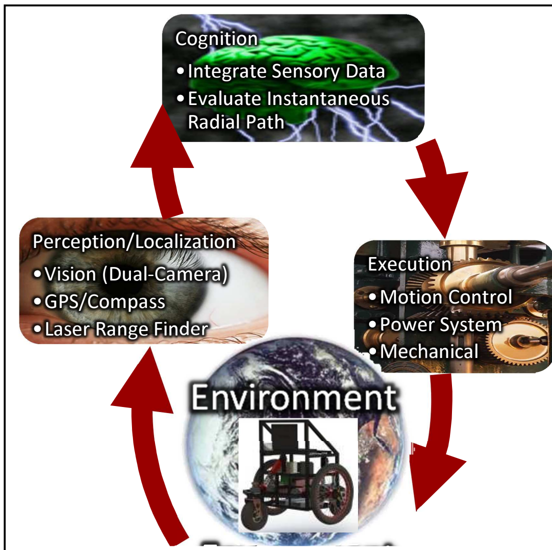


Figure 1.2 – System Integration Flow Chart

## 2. Mechanical Design

LINJA is mechanically designed to navigate through an obstacle course fast and efficiently. LINJA is designed with a flexible frame using an independent drive-wheel suspension system, stabilizing the platform and minimizing tilt by keeping all wheels in contact with uneven surfaces at all times. The platform is designed to make the robot very maneuverable, keeping the horizontal center of gravity stabilized over the center of the differential-drive axle. The vehicle also features a drive axle decoupler to uncouple the drive wheels from the gearbox for ease of travel. LINJA's platform is designed to be compact, lightweight and weatherproofed with a transparent polycarbonate shield. LINJA also features a Liquid Cooling System to control the ambient temperature on the underside of the laptop's hottest area, keeping it at an operational temperature, and has a portable design to be dismounted when an onboard cooling system is not needed.

LINJA's platform design was inspired by the success of last year's IGV platform, RED RAVEN, incorporating a flexible, diamond shaped chassis, shifting the horizontal center of gravity to be located over the driving axle, resulting in smooth turning. The component layout is lower, more compact, yet still accessible and user friendly. The Liquid Cooling System controls the ambient temperature around the laptop.

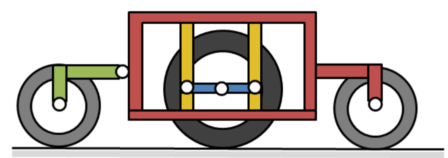

Figure 2.1 – Basic schematic of Flexible Frame

### 2.1 Flexible Frame Design

Inspired by last year's IGV platform, a flexible yet stable chassis design was desired. During the design process, an independent drive-wheel suspension system was incorporated as well as a flexible Front Pivot. LINJA's frame is propelled by two drive wheels (Center) which are able to slide up and down independently on linear bearings (Yellow). The frame balances itself on two floating wheels (Front and Rear). The rear wheel is fixed to the main platform (Red) while the Front Pivot (Green) uses a spring/damper to give the frame balance and flexibility. This configuration allows each wheel to move in the vertical direction independently of other wheels and the weight of the main platform is distributed equally on all four wheels, giving the ground vehicle better traction. The independent drive wheel suspension allows the platform to tackle uneven surfaces while keeping the center of gravity stabilized over the drive axle.

LINJA's chassis is constructed from 1-inch 6063-T6 Aluminum square tubing keeping the frame light weight yet strong. The frame was TIG welded with sealed corners, maximizing the frame's strength and minimizing oxidation.

#### ❖ 2.1.1 Independent Drive-Wheel Spider Slider Suspension

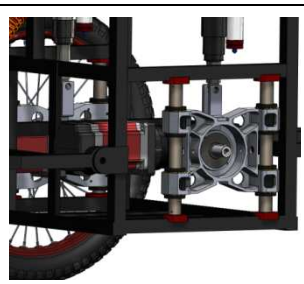

Figure 2.2 – Spider Slider Suspension

LINJA's independent drive wheel suspension uses a unique linear bearing Spider Slider design. The Spider Aluminum pillow-block slides on linear bearings over two steel tubes to keep the drive wheel stable. The drive wheels can travel on a stroke length of 3.5-inches to handle a variety of terrain. The Evolver ISX airshock by Manitou, is used as suspension allowing for an adjustable spring constant and damping ratio.

Because the drive wheels and the motors are no longer stationary, a ninety degree gearbox configuration is used to keep the motors from interfering with components and the frame itself. The gearbox is attached to the Spider Slider and moves in correspondence with the suspension system.

### ❖ 2.1.2 Front Pivot Suspension

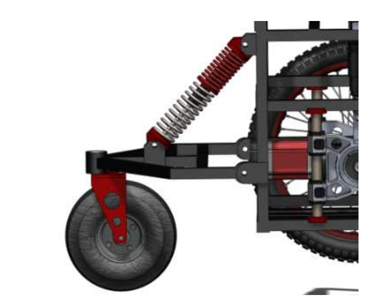

Figure 2.3 – Front Pivot Suspension

The Front Pivot was designed with a clearance to accomidate the motion rage of the moving motors. It pivots around two pivot points on the frame's front end.

The Front Pivot uses a custom built coil-shock configuration using two springs in series and an adjustable dual-action damper acting as the center axis. The Front Pivot and the Independent Spider Sliders design complement each other to provide flexibility and balance to the plaftorm.

## 2.2 Compact Component Layout

The component layout was designed with the care to keep the CG located directly over the drive train as low as possible and as close to the center verticle axis as possible. These two goals will minimize the robot's rotational inertia. The third goal was to lower the overall component layout and minimizing the danger of platform tipping.

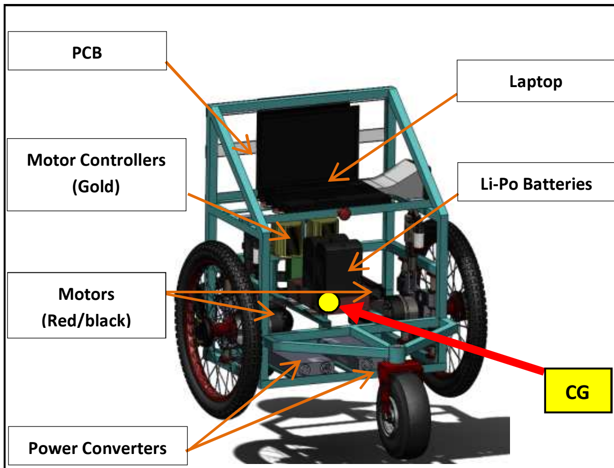


Figure 2.4 – Component layout on LINJA's frame (back view)

As shown in Figure 2.4, the components were strategically mounted to avoid interferance with moving components and to keep the CG very low, yet remain accessible. The heaviest components are placed at the bottom and the lighter are placed at the top. The motors, gearboxes, batteries and motor controllers are all located on the lower level of the robot, while the PCB, laptop, GPS receiver and antenna are located higher up. The laptop is at a user-friendly accessible level. Overall, the CG is kept low and as close to the central verticle axis of the robot, allowing LINJA to perform quick turns while remaining stable.

## 2.3 Drive Wheel Decouplers

The drive wheel decouplers are redesigned and incorporated on LINJA. The decouplers allow the robot for easy transport and to eliminate damage to the drive train. This design allows the user to decouple the trive train from the drive wheels.

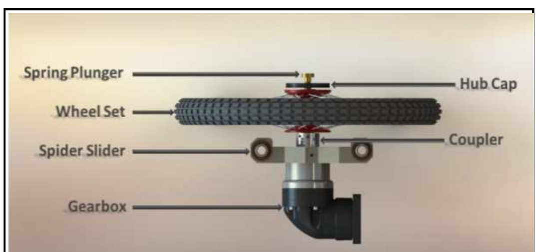


Figure 2.5 – Drive Wheel Decoupling Mechanism

The drive axle is attached to the wheel by means of a spring plunger mounted on a hub cap. A simple half turn of the spring plunger disengages the wheel from the axle and allows the wheel to spin freely. This allows the robot to roll freely wherever the user needs to go and significantly speeds up testing process and preserves the team members' strength and energy. If the wheel has to be removed for maintenance or repair, removing a series of three bolts on the coupler will allow the wheel to be removed completely. This design has proven to be a simple, compact and effective.

## 2.4 Liquid Cold-Plate Cooling System

In previous years, the IGV team has experienced multiple computer crashes due to the laptop overheating. CSUN's IGV team had previously used standard 120mm computer fans to provide cooling. A high performing cooling system was needed and is now designed that could be integrated with any standard laptop, without harming any of the electrical components that surround the cooling system.

Figure 2.6 – Liquid Cold Plate Cooling System

A liquid cold plate, using bendable copper tubing, Aluminum plates and thermal putty was fabricated. A non-conductive heat transfer fluid called HFE-7100 is the fluid used to control the temperature. The fluid is pumped through the cold plate by a compact reservoir, radiator and pump. The laptop can be placed on the cold plate and secured with industrial Velcro. Using this design, the ambient air temperature underneath the laptop's hottest area can be controlled. To further ensure a temperature controlled environment, a custom designed vent is attached to the laptop's vent area to force hot air out and away from the CPU using forced convection to control the temperature inside the laptop. This design is portable and can be easily removed from the platform should conditions not require a cooling system on board.

## 2.5 Durability and Serviceability

The last major component that is featured on LINJA is the polycarbonate weather-proofing. Polycarbonate is used because of its low weight and transparent properties. Since the material is transparent, all of LINJA's components are visible despite being covered for protective purposes. Additionally, polycarbonate can simply be bent with a sheet metal bender; there is no need to heat the material in order to get a clean bend. As a result, the polycarbonate provides LINJA protection from unwanted moisture and provides a clean, professional finish.

The ease of serviceability is a necessary feature that ensures the longevity of LINJA's design. All the screws that secure the different polycarbonate panels to the frame are medium-sized thumbscrews, allowing any panel to be removed as needed without the use of any tools. Additionally, once the shielding is removed, all components are at an accessible reach should maintenance be required. This design allows LINJA to be a swift, maneuverable and compact platform design.

# 3. Electrical/Power Systems

Increased reliability, reduced weight, and reduced power consumption were all critical goals for this year's IGV robot, LINJA. As such, many of the designs were inspired by the success of last year's robot design, RAVEN, including an improved custom printed circuit board, new Lithium Polymer batteries as the main power source with shorter charging time, an improved motion control utilizing the servo controller's initialization code and a new wiring for motors and power system.

## 3.1 Electrical Configuration

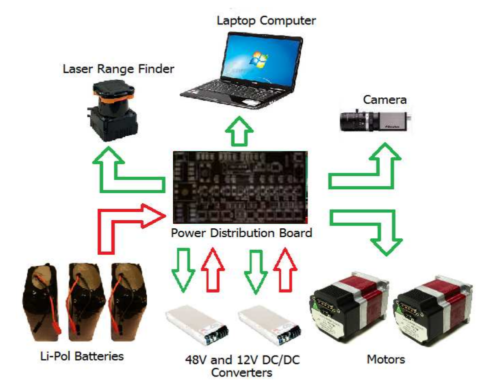

Figure 3.1 – LINJA's Power System

Seeing the success of RAVEN's electrical configuration, LINJA's power distribution is directly inspired by last year's platform. Like RAVEN, all electrical components tie into LINJA's power system through a new custom made power distribution printed circuit board (PCB) (Figure 3.1). Main power is provided by a set of three series-configured Lithium Polymer batteries. Battery voltage is converted and regulated by two separate buck/boost DC to 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. This allows the motors to function at their ideal operating voltage of 48V DC—which produces higher RPMs.

A printed circuit board provides a more stable and reliable platform compared with using exposed terminals or wires only. The new PCB for LINJA is a 2.3 mm thick durable and high reliability black design consisting of 3 oz/ft$^2$ of copper for the electrical traces, and is able to handle currents as high as 50 Amps (Figure 3.2). It features LED Light-Bar indicators emitting soft but bright colors indicating if current is flowing through an electrical device. Due to the new circuit design, the PCB copper trace density has increased dramatically which subsequently lead to a double layer design. Additionally, more details to the PCB silk layer (printed electrical symbols and legend) has been applied, thus making the mounting of electronic/electrical components easier. Finally, a fuse malfunction circuit detector is included for every fuse that protects each electronic device.

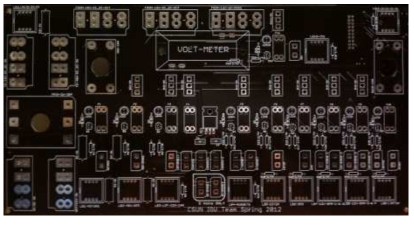
Figure 3.2 - Power Distribution Printed Circuit Board

## 3.2 Power Analysis

LINJA's power consumption analysis was conducted at nominal and extreme loads (Table 3.1). The nominal load was evaluated when LINJA traveled at 1 mph. The extreme load was evaluated at 6 mph which, based on the data obtained, requires the most power consumption (1332 watts). To prevent power outrage, new high energy density Lithium Ion Polymer (LiPo) battery packs were built (four packs in total). Each battery pack will provide a nominal and extreme voltage of 15 and 16.8 respectively. Each LiPo battery pack will yield a minimum of 450 watts and a maximum of 504 watts. Internally, each LiPo battery pack is protected against over current, over voltage, deep discharge (under

**Table 3.1: Power Consumption**

| Type of Load | Nominal Power (W) | Maximum Power (W) |
|---|---|---|
| **Total Base Load** | 169 | |
| **Transient Motor Load** | 720 | 1000 |
| **Speed Indicator** | 24 | 30 |
| **Total Load** | 913 | 1199 |
| **Total with 90% DC/DC Efficiency** | 1014 | **1332** |

voltage) and over heat—total output current is limited to 30-Amps per pack. A series configuration of three LiPo battery packs was chosen to provide the required power consumption. This assures that LINJA receives the necessary power under nominal and extreme conditions.

### 3.3 Emergency Stop

The emergency stop system on LINJA makes use of the motor controller's built-in stop conditions by bringing the robot to a controlled and safe stop. Two systems, a mechanical pushbutton stop and a wireless remote stop, are individually used. When activated, the motor controllers' input/output pins receive a logic HIGH +5V signal and when not activated, a logic LOW is received. To ensure that the wireless stop and the mechanical stop do not conflict with each other, a series configuration is used as a safety design.

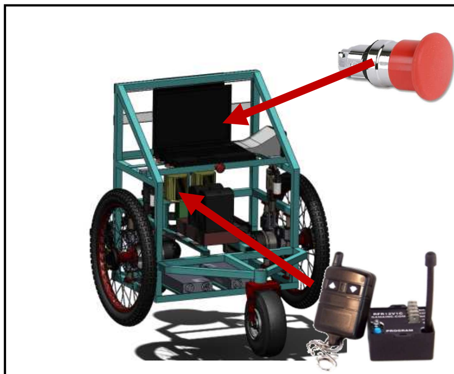

Figure 3.3 – The push button (top) and wireless (bottom) e-stops utilized on LINJA

### 3.4 Arduino Autonomous Light and Speed Meter

LINJA's electronic design includes the Arduino Micro-Controller Unit (MCU) to activate the autonomous light. In addition, the Arduino MCU is used to provide a visual analogous LED speed indicator. The speed indicator is interfaced with an LED Integrated Circuit Driver (LM3914) to adjust the LEDs based on LINJA's current speed. The LabVIEW code to perform these two tasks under the Arduino MCU is short and provides real time code execution. This facilitates LINJA to comply with the IGV 2012 rules, LINJA must indicate when it is powered on through the use of a solidly lit light. Specifically, when the IGV is autonomously running, the light must begin flashing.

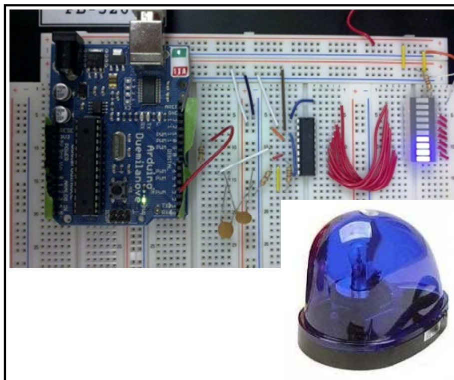

Figure 3.4 – Arduino autonomous light activator and speed meter.

## 4. Vision

LINJA features a Dual-Camera configuration allowing for a real-time video feed of wide-angle images to provide boundary line and obstacle data ahead of the IGV. The Dual-Cameras are positioned on the IGV to provide the most useful boundary line and obstacle image data for the Vision algorithms to process, and oriented to allow for the dual images to be properly combined. The camera and accompanying Vision system are necessary to isolate the environmental data that cannot be detected by the other sensors of the IGV. Additional improvements over previous Vision systems allow LINJA to not only detect obstacles from the surroundings, but also to distinguish between different types of obstacles and generate a color map showing similar obstacles of different colors.


Figure 4.1 – Sony Handycam (left) and Sony VCL-HA07A lens adapter (right)

## 4.1 Camera Hardware

The Dual-Cameras selected for LINJA are two identical DCR-HC28 miniDV Sony Handycams. These camcorders are used for their on-board light and white balance to help the Vision system dynamically adapt to changing light conditions without the need for additional light sensors. Cameras with the IEEE1394 data transfer interface were chosen for ease of use and compatibility with almost any computer setup. Two cameras are used instead of one to increase the field of view without facing the significant distortion of a single camera with a very wide angle lens. To further enhance the data acquisition, Sony VCL-HA70A wide angle lens adapters with a factor if 0.7X view angle increase are used on each camera, providing a 52 degree vertical 76 degree horizontal view angle for each camera. The lens adapters shown in Figure 4.1 provide a noticeable increase to the view angle of each camera, and because the view angle improvements are distributed between the two cameras, the overall view angle is enhanced without a significant increase in distortion.

In order to provide the right amount of separation and overlap between the two camera feeds for combining into a single image, the Dual-Cameras are mounted next to each other with each camera angled outward 60 degrees from the IGV's forward centerline field of view, as shown in Figure 4.2. Because the cameras are angled downward to avoid direct sunlight, perspective correction is applied to the projected image, along with rotation and translation, to provide an image with accurate real-world coordinates of the environment.

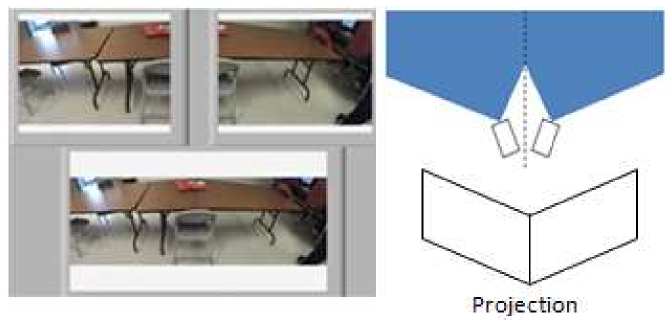


Figure 4.2 – Combined projected view for Dual-Camera configuration

## 4.2 Vision Software

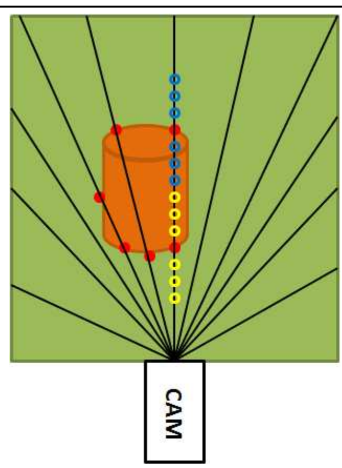


Figure 4.3 – Color Edge Detection

The main goal for improving the Vision algorithms over previous versions is to not only isolate anything that would be considered an obstacle by the IGV, but to better differentiate between the different types of obstacles in the environment. While previous versions of the Vision algorithm were focused more on isolating boundary lines, BGL Line Detection and Color Edge Detection allow for boundary lines, barrels, flags and different colored obstacles to be identified not only from the environment, but also from each other, based on color values and ranges using the RBG and HSL color scales. The Grass Filter and Canny Edge Detection provide a preliminary analysis to grayscale input images for the Dual Camera setup on LINJA. Calibrated ranges of the RBG and HSL color scales along with size analysis and noise filtering are used to isolate different colored flags. The Grass Filter is used to darken a grayscale input image to reduce noise in grassy terrain while maintaining contrast between white boundary lines and the rest of the image. Canny Edge Detection is used to enhance the boundary lines as well as detect the edges of obstacles in the image. These two steps provide the initial removal of unwanted noise and extraneous data for the rest of the vision algorithms.

## 4.3 Color Edge Detection

Color Edge Detection is a new vision algorithm developed for differentiating between different types of objects in the environment based on changing color values across a color input image. The first step of Color Edge Detection is to remove as much unwanted data as possible using the Grass Filter and then apply Canny Edge Detection to find the edges of any objects in the image. Canny Edge Detection uses contrast changes in a grayscale image to highlight all the edges, but does not provide any information as to where those edges are located. To find the edge locations, Color Edge Detection scans the image along a line for each angle increment specified. At each angle along the line that an edge is detected, a check is made along the line on both sides of the edge for a change in color. Several color samples on each side of the edge are used to determine average color values. Once enough line samples are taken, the average color values for a section of the image, instead of just along each line, can be determined. Based on the average color values on either side of the edge, that part of the image can be distinguished as a specific type of object in the environment.

The sample camera image in Figure 4.3 shows how the color image seen by the camera is checked along a line at incremental angles. For the orange barrel, the edge locations in red are found based on the average color value changes from green to orange and orange to green along each line. The vertical line shows several color samples taken, and the blue and yellow sample groups show that a separate color edge is detected for each sample group along the line.

Similarly, it can be seen in Figure 4.4 the main steps leading up to color edge detection and their results. The input image is grayscaled and preliminary filters are applied. Then Canny Edge Detection is used to find all of the edges in the image. The last step is for the Color Edge Detection algorithm to scan the color image as described in Figure 4.3, and detect color changes. Step 4 shows the edge of the grass as green and the edge of the cement path as white. Based on the color of the edges, the IGV can determine the white edge as a boundary for the grass field.

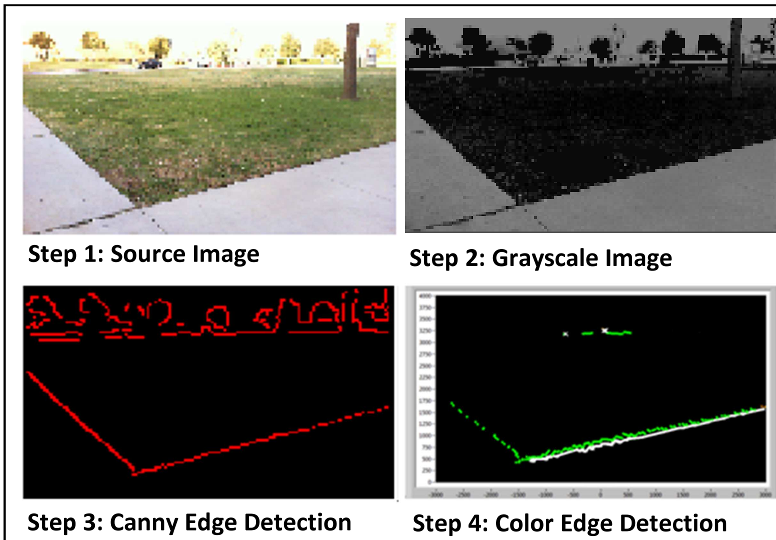


Figure 4.4 – Color Edge Detection in action

# 5. Navigation/JAUS

## 5.1 Navigation Hardware

Figure 5.1 – Navigation hardware used on LINJA includes an (a) antenna, (b) receiver, and (c) compass.

LINJA uses a NovAtel SPAN (Synchronized Position Attitude & Navigation) system for navigation. This system consists of a GPS-702L antenna (Figure 5.1a) with a ProPak-LB plus receiver (Figure 5.1b). OmniSTAR provides the IGV an HP differential GPS service, which increases the accuracy to 0.1 meters. The ProPak receiver merges the GPS and the compass data and provides latitude, longitude data to the computer. The receiver interfaces with the computer via an RS232 serial connection at a baud rate of 460.8kBd while updating at 40 Hz.

LINJA's True North Revolution 2X digital compass (Figure 5.1c) identifies heading, pitch, and roll which is used primarily to guide LINJA. This compass provides heading data at 31.25 Hz, with an accuracy of ±0.5°. It communicates with the computer via an RS232 serial connection

## 5.2 Navigation Software Strategy

The overall software strategy for the Navigation part is shown in Figure 5.2. It begins by accepting a list of waypoints along with the location and direction data. For a large number of waypoints, approximation methods are used to efficiently find the shortest path. However, due to the fairly small number of waypoints a brute force algorithm was used to quickly implement a solution. The Navigation program chooses the shortest total distance after iterating through every possible path. An ordered list is created, and the vehicle uses the first point as its goal.

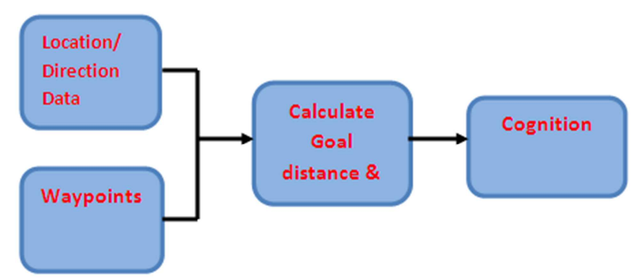

Figure 5.2 – Navigation software strategy flow chart

The program continuously calculates the bearing and distance between the vehicle's current position and its current goal waypoint using the Haversine and Great Circle formulas. The vehicle's compass heading is compared to the bearing in order to find the relative direction to the goal (i.e. the goal angle). The goal distance is constantly monitored, and when it becomes 1 meter, or half the radius given in the IGVC rules, the current waypoint can be checked off. This angle and position are provided to Cognition, for path planning purposes.

## 5.3 Joint Architecture for Unmanned Systems (JAUS)

JAUS is a protocol designed by the Department of Defense to facilitate the communication and cooperation 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-Navigation and Navigation. The Non-Navigation 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, Navigation 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.

As shown in Figure 5.3, 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 information is relayed to navigation queries, where it simultaneously executes the navigation data and reports information.

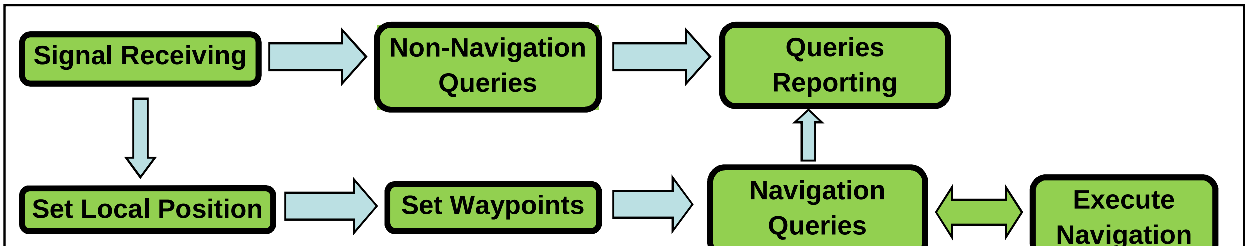

Figure 5.3 – JAUS software strategy flow chart

## 6. Cognition/Motion Control

The Cognition and Motion Control System is responsible for the algorithmic calculations to decide upon and control the optimal path for LINJA. The main three inputs to these algorithms are Navigational Waypoints, Vision Data, and a polar histogram generated by a Scanning Laser Rangefinder. For LINJA, the UTM-30LX was selected, as shown in Figure 6.1. To attain the maximum amount of useful information, the sensor is placed centered on the robot directly above the front caster wheel. Although it has a scanning range of up to 270° and an angular resolution of 0.25°, LINJA samples only 180° of data at an angular resolution of 1°. This allowed for the best compromise between informational accuracy and processing time, as benefits noticed from increasing the resolution or width of the scan range were minimal. The LRF is connected via USB 2.0 to an HP EliteBook 8560p running LabVIEW, in which the rest of the software is programmed, and is mounted at the lower front of the robot (right behind the IGV's front caster wheel).


Figure 6.1 – UTM-30LX

### 6.1 Cognition Software Strategy

The approach used by the IGV approximates the true movement of the robot by basing its paths on instantaneous turning radii, rather than vectors. This allows the robot to more precisely navigate the course, as the true path of the robot will be a single, curvilinear path, rather than a series of shorter, straight vectors. This general algorithmic approach has been named RPH with OPID, or Radial Polar Histogram with Optimal Path Interference Detection.

Once the laptop obtains data from both the LRF and the Vision system, it converts all of the data into a single, combined Cartesian map of the various obstacles. The map is then filtered to neglect the data that are not within a controlled range. The data is then grouped, allowing the robot to determine which areas are blocked off to the robot. The previous grouping algorithm used on last year's platform, RED RAVEN, is optimized, reducing the overall RPH processing time by over 20% on average and ignoring thousands of unnecessary calculations per iteration. The grouped data points are then passed through another algorithm which assumes that each group represents a curved surface, such as a barrel, as shown in Figure 6.2. This algorithm returns the radius of the group, along with the coordinates corresponding to its center. If the radius is within a calibrated threshold, it will identify the group as a barrel, allowing the program to store only its center and radius data, rather than all of the individual data points which define it.

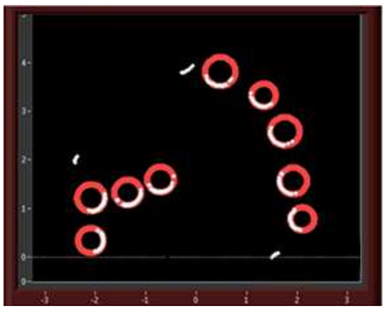

Figure 6.2 – Obstacles identified as cylinders.

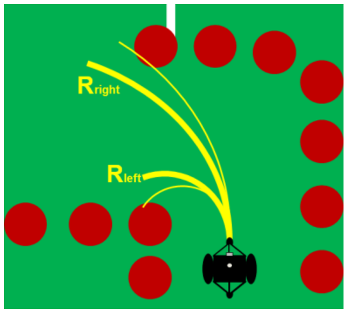


Figure 6.3 – Left and Right radial paths. Note the offset from the true exterior radii for padding.

After the groups have been clearly defined based on the dimensions of the IGV, the current navigational heading is compared to the various open paths to decide which opening will lead the robot to the navigational waypoint along the most direct path. Once the desired path is found, the IGV determines the leftmost and rightmost radii that the robot could safely travel without touching obstacles, as shown in Figure 6.3, and selects the ideal radius between the two to lead the robot towards its destination.

However, as the robot bases its path selection purely on the edges of the obstacles, there was a potential for the robot to still decide on a path that intersected an obstacle in some configurations. As such, OPID was added as a safety check before the optimal path was finalized. OPID compares the projected radii of the edges of the robot as it executes its predicted turn to the original histogram. If OPID detects any obstacle that would intersect the location of the robot at a close point along its path, it forces the data filter range to be reduced to recalculate a clear path base on the intersecting obstacle. This is demonstrated in Figure 6.4.

Finally, RPH will establish a theoretical location of the robot at a future time if it were to follow the path and then performs the entire set of calculations again, thus allowing the robot to determine its future location. This information is then passed on to Motion Control.

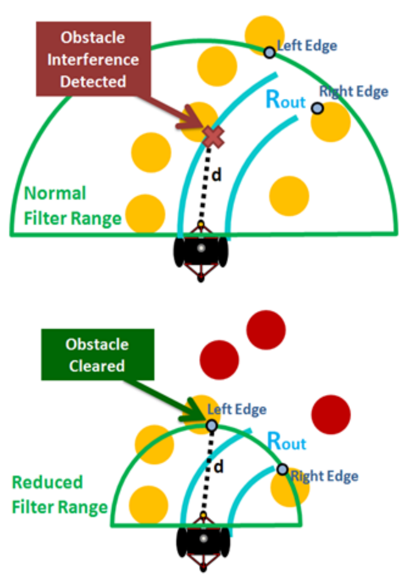


Figure 6.4 – When OPID detects obstacle interference, it reduces the data filter range to recalculate a path.

## 6.2 Motion Control Software Strategy

Motion Control receives the turning radius desired by Cognition and sends the appropriate signals to the motors to execute the path. It determines the ideal proportionality between the motors to allow the robot to turn with a given radius. Then, based on predefined minimum and maximum speed values, along with the combined field histogram, it determines the ideal overall speed for the robot. In this way, if the histogram is particularly cluttered with obstacles, the robot will instinctively move more slowly, allowing for tighter control, and if the histogram is clear, the robot is free to accelerate to a faster pace.

Additionally, by looking at the future location of the robot, Motion Control can dynamically calculate the ideal robot acceleration. This helps prevent jarring stops and allows for a much smoother transition between crowded and open areas.

## 6.3 Mapping

Due to the optimized data storage, allowing three numbers to represent a large volume of data points, the mapping algorithms allowed the IGV to track its motion as it traverses the course, while also storing the locations of any obstacles it encounters along the way. This Global Map can be used for debugging purposes, as operators can view the robot's path and the data which led it to its decision.

Additionally, as each set of data points is stored as a single circle, a real-time 3D representation of what the IGV sees can be generated, as shown in Figure 6.5. This innovative mapping technique allows users to compare what the robot perceives to the obstacles surrounding it quickly and easily on a clear and user-friendly display.


Figure 6.5 – GUI of IGV during trial run. The orange cylinders represent barrels perceived by the IGV.

## 6.4 Specific Obstacle Avoidance

Despite the realistic path planning that RPH can perform, some configurations of obstacles can be problematic. A perfect example is when the robot finds itself trapped in a dead end. Whenever a dead end is detected, the robot will immediately stop and perform a zero radius turn, allowing it to scan its surroundings for an exit. Once an opening is detected, the original RPH algorithm would begin to execute the radial path towards the opening without taking into account that the wheels were travelling in opposite directions as the robot turned. This meant that the robot would frequently miss the turn as one wheel was much farther from its ideal velocity than the other. To handle this issue, a Velocity Control Factor, or VCF was added in. This factor begins at 0 and slowly rises to 1. The desired speeds sent to the motors are multiplied by this VCF, allowing the motors a chance to accelerate simultaneously and giving more precise control.

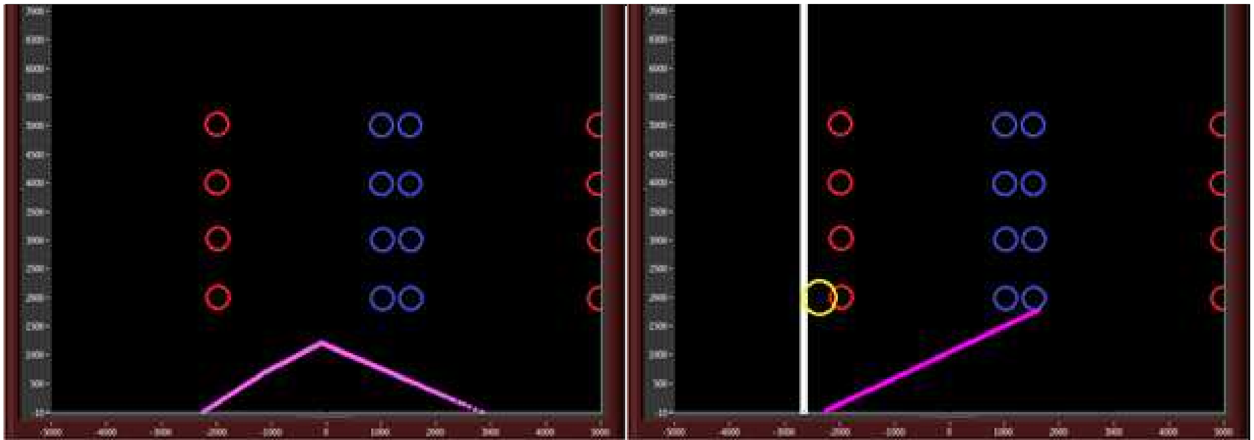

Figure 6.6 – The previous method (left) would trap the IGV. By considering the presence of the white line, the IGV only draws one ghost line, forcing it towards the correct path (right).

Another problematic configuration is the channels of flags shown in Figure 6.6. Previous IGV entries would draw an imaginary boundary line, or ghost line, from the flag to the side of the robot, forcing the robot to travel to the opposite side of the flag. However, if the robot were to approach the wrong channel, this method would create a dead end directly in front of the robot. To fix this, a specific case was added that does not create a ghost line if the specific flags are near a white line—as such, only the ghost line from the center row of flags is drawn, blocking the incorrect path and leading the robot in the correct path.

## 6.5 System Integration

Figure 6.7 shows the process that LINJA uses for the complete integration of all onboard systems. LINJA uses a LRF and a Dual-Camera configuration to scan for objects and line boundaries in the robot's environment. It also has a GPS onboard that gathers waypoint locations and the robot's heading information. These sensor data are sent to Cognition for comparison and ghost line generation. Cognition creates a local map and, by evaluating the open blocks in a selected area, a desired radius is determined and is sent to Motion Control for execution. While Motion Control moves the robot, it provides continuous feedback of the robot's predicted turning radius and velocity to Cognition for determining the ideal motor acceleration for each wheel that optimizes the smoothness of the robot's motion. The sensors then rescan the environment and the process is re-iterated again.

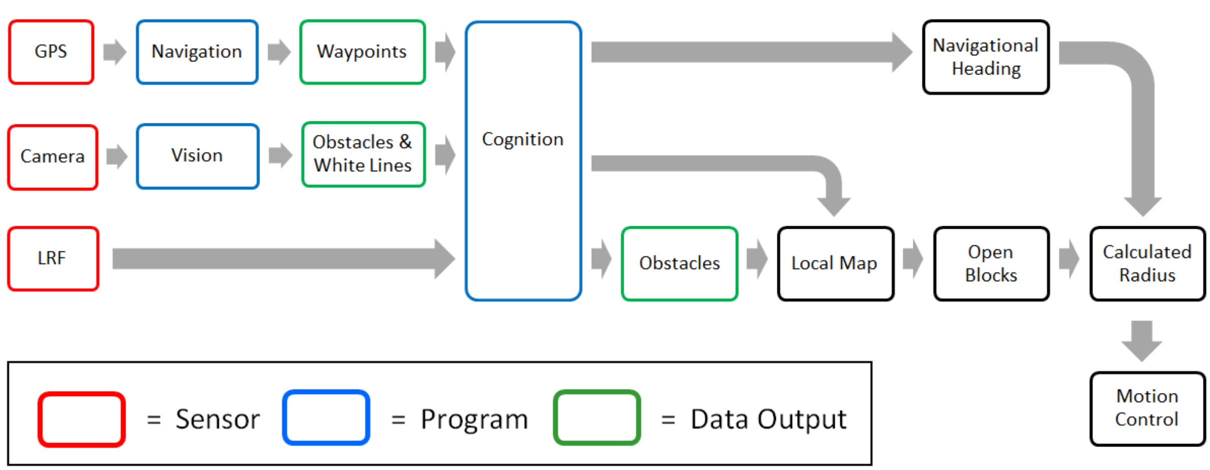


Figure 6.7 –System Integration flow chart

## 7. Overall System Performance

| Table 7.1 – LINJA's Performance | |
|---|---|
| **Parameter** | **Value** |
| Top Speed | 6.5mph |
| Reaction Time | 40-100ms |
| Ramp Climbing | 30$^{o}$ |
| Battery Life | 4-9 hrs |
| Obstacle Detection Distance | 8m |
| Waypoint Accuracy | 10cm |

LINJA is a unique IGV that has been designed, built, and programmed to achieve efficient maneuverability, high speeds, precise path planning, and near- perfect obstacle avoidance. The overall performance parameters of LINJA are shown in Table 7.1. Due to a light weight flexible chassis, the use of a 48V DC/DC converter, and three series-configured Lithium Polymer batteries, the motors are able to drive the IGV to a top speed of 6.5 mph on pavement and 5.5 mph on grass, depending on terrain conditions. With a faster processing time, the reaction rate of LINJA is clocked at around 100ms when using the 3D mapping capabilities and 40ms with them disabled. Due to the IGV's Spider Slider Suspension frame, LINJA is capable of tackling uneven surfaces and climbing 30$^{o}$ ramps. The three series-configured Lithium Polymer batteries provide enough power to run LINJA at top speed for 4 hours. Under normal operating conditions, the battery life is approximately 9 hours. The LRF detects obstacles up to 8 meters away, and the Dual-Camera configuration detects obstacles and lines up to 7 meters away. Lastly, through the use of a Differential GPS system (DGPS), the waypoint accuracy is approximately 10 cm.

## 8. Appendix

### 8.1 Total Cost Estimate and Person-Hours

The total retail cost of all the materials and components utilized on LINJA is approximately $48,144.87 (Table 8.1). The most expensive components contributing to the cost are the GPS Antenna/Receiver, batteries, Spider Slider Assembly and LRF. However, we've received donations that made this year's platform possible. Aero Seating Technologies graciously donated both Spider Slider Assemblies (which we slightly modified to fit LINJA's body) saving a significant amount of the team's budget. Most other components were reused from previous platforms and additional discounts were obtained when purchasing new vision equipment. Overall, the total purchase cost for LINJA is approximately $52,000. However, since many of the components were recycled from the previous IGV, the total cost to the team in constructing LINJA this year is approximately $13,018.13. In addition, each of the students spent approximately 26 hours a week on the IGV project. Thus, during the 2011-2012 academic year, each of the students spent about 1040 hours working on LINJA (Table 8.2).

| Table 8.1 - Total Cost Estimate of LINJA | | | |
|---|---|---|---|
| **Components** | **Retail Cost** | **Cost at Time of Purchase** | **Cost to Team This Year** |
| Hokuyo LRF | $5,400.00 | $5,400.00 | $5,400.00 |
| 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 Pack Batteries | $4,150.00 | $4,150.00 | $2620.16 |
| HP Elitebook 8560p | $2,100.00 | $2,100.00 | $2,100.00 |
| 48V/12V DC/DC Converters | $450.00 | $411.50 | $411.50 |
| Printed Circuit Board | $36.00 | $36.00 | $36.00 |
| Misc. Electrical Items | $500.00 | $500.00 | $500.00 |
| GPS Receiver/Antenna | $28,079.00 | $8,500.00 | $0.00 |
| Digital Compass | $467.00 | $397.00 | $0.00 |
| FireWire ExpressCard | $38.21 | $38.21 | $38.21 |
| Sony Handycam Camcorder | $420.00 | $324.98 | $324.98 |
| Sony Wide Conversion Lens | $100.00 | $66.60 | $66.60 |
| Gearboxes | $1,204.50 | $1,204.50 | $0.00 |
| Driving Wheels/Rims | $200.00 | $195.88 | $195.88 |
| Spider Sliders/Bearings | $2,500.00 | $2,500.00 | $0.00 |
| Caster Wheels | $80.00 | $76.24 | $76.24 |
| Metal Materials | $133.16 | $133.16 | $133.16 |
| Misc. Mechanical Materials | $159.64 | $159.64 | $159.64 |
| Airshocks/Springs/Dampers | $849.95 | $712.28 | $712.28 |
| Cooling System | $522.62 | $522.62 | $243.48 |
| **Total** | **$51,627.08** | **$31,665.61** | **$13,018.13** |

| Table 8.2 - Hours worked per student | |
|---|---|
| **Time Period** | **Hours** |
| In Class (per week) | 12 |
| Out of Class (per week) | 14 |
| Total Per Week | 26 |
| **2011-2012 Academic Year** | **1040** |