19th Annual Intelligent Ground Vehicle Competition

CL.E.T.I.S.

CitadeL Engineered Traversing Intelligent System

Autonomous Mobile Robot

Kendall Nowocin _____

Matthew Player _____

Nathan Lett _____

Luther McBee _____

The design and engineering of the CitadeL Engineered Traversing Intelligent System (CL.E.T.I.S) by the current student team has been significant and equivalent to what might be awarded credit in a senior design course.

_____

Advisors: Dr. Mark McKinney and Dr. Johnston W. Peeples

1. Team Overview and Organization

This is the first time that The Citadel has competed in the Intelligent Ground Vehicle Competition (IGVC). The team members are senior undergraduate electrical engineering students: Nathan Lett, Luther McBee, Kendall Nowocin, and Matthew Player. The design, fabrication, and validation of the robot were completed over a two semester senior design capstone course.

The team organization is shown in Fig. 1. The team had meetings in June and July to brainstorm on critical aspects associated with an autonomous robot. In August, the team leaders were voted, robot subtasks were assigned, and a Gantt chart and process flowchart was used to visualize the workflow. The first semester team leader would oversee progress of design and prototyping, and the second semester team leader managed the manufacturing and testing. The team members were assigned areas of concentration based on their background experience and preference. The Gantt chart and process flowchart were used to determine high priority tasks, process bottlenecks, and time tracking. Weekly meetings were held to discuss issues that arose, progress made, and deliverables for the capstone class.

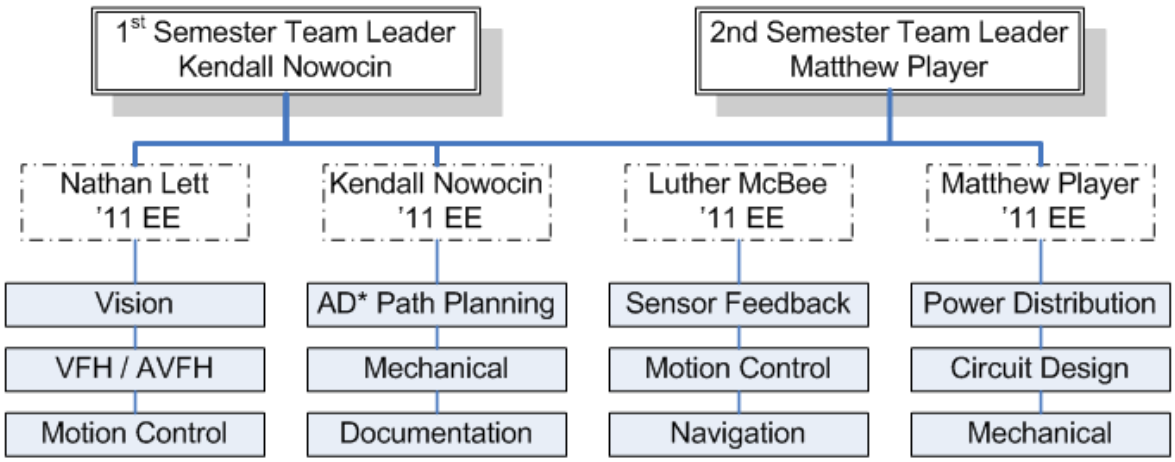


Fig. 1: Team Organization Diagram

2. Design Process

The procedure for developing CLETIS was a multistep process involving design methodology and a variety of engineering tools. The design methodology is shown in Fig. 2.

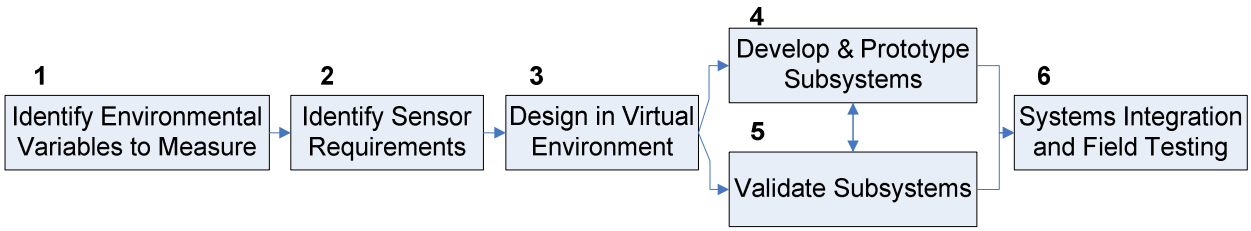


Fig. 2: Design Methodology Flowchart

Steps one through three were done in sequential order for the overall system while steps four and five were done in parallel for multiple subsystems to maximize development efficiency. In step one, flowcharts were created to determine the required sensor data that would be used to make intelligent navigation decisions. Distance to obstacles, line recognition, feedback sensors, and latitude/longitude coordinates were the best parameters to measure. Step two analyzed engineering tradeoffs in speed regulation, image processing, pulse width modulation circuits, and etc. using Excel spreadsheets. In step three, computer aided design (Autodesk Inventor 2010) and modeling software (Matlab and Simulink) were used to develop the structural frame and simulate theory based design, respectively (Fig. 3 and 4).

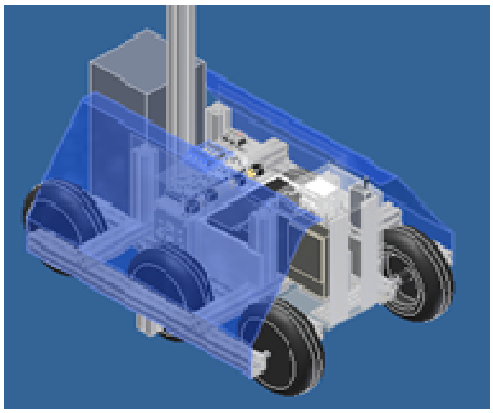


Fig. 3: CAD of robot

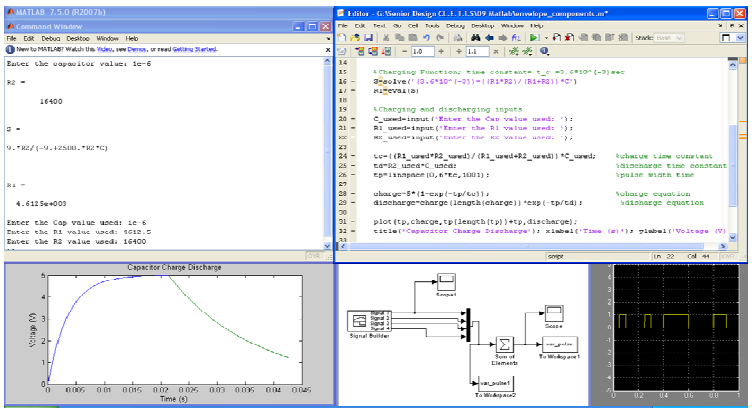


Fig. 4: Matlab and Simulink of Ultrasonic timing circuit

Steps four and five used LabVIEW and a commercial off the shelf robotics controller to develop, prototype, and validate subsystems. The ultrasonic array, Kinect camera, and RGB camera were the main tasks during this phase. A Gantt chart was created using Microsoft Project to give transparency and streamline the development process to the time constraints. Fig. 5 shows the main tasks in black, and the total 126 subtasks are suppressed inside each appropriate main task.

	Task Name	Duration	Start	Finish
41				
42				
43				
44	Robot	211 days?	Wed 8/25/10	Mon 6/6/11
45	Mechanical	53 days	Wed 9/1/10	Fri 11/12/10
53	Sensors	72.65 days	Wed 9/1/10	Thu 12/9/10
57	Navigation	74 days?	Wed 8/25/10	Fri 12/3/10
62	Power	87 days	Wed 9/1/10	Wed 12/29/10
67	Kinect Camera	81 days?	Tue 12/14/10	Thu 3/31/11
72	Software	104 days?	Wed 8/25/10	Thu 1/13/11
76	Testing	115 days?	Sat 1/1/11	Thu 6/2/11
112	Documentation	147 days?	Wed 10/27/10	Tue 5/10/11
146	Competition	3 days	Thu 6/2/11	Mon 6/6/11

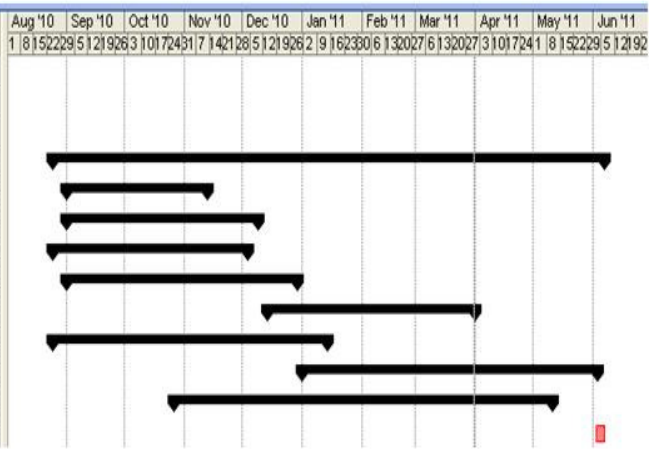


Fig. 5: Project Schedule

3. Efficient Use of Power and Materials

Green engineering practices were utilized in the manufacturing of CLETIS. The structural and electrical components were salvaged from an automated test bench that the team acquired. Efficient use of power was an important concern that was focused on at the beginning of the design phase. A detailed list of current ratings, voltage ratings, weight, and size of all components was compiled to determine the power requirements. The results of the data showed that a 75amp-hour battery would provide sufficient power for at least 10 hours of idle operation and 2 hours of run time. A DC to DC converter with greater than 95% efficiency was used to improve on the efficiency because multiple voltages were required. CLETIS's power consumption was validated during the field testing step. CLETIS exceeded all the minimum requirements and stayed operational for 12 hours in idle and 2 hours while running.

The overall profile of CLETIS was designed to minimize the area that needed to be water resistant. Three dimensional concepts were designed in Autodesk Inventor, and many component layout problems were resolved before the first piece of material was cut. CLETIS was designed to be modular, and the aluminum extrusion material used made this feasible. The electronics were grouped and located to provide short and neat wire runs for better cable management. The serviceability of the main electronics systems were improved by placing all the components in a single area that was user friendly to service. A power cube was constructed and located on the front of CLETIS. The components fit in the folded box to allow for efficient use of space and superb protection from the elements while allowing it to unfold completely within 30 seconds.

4. Safety, Reliability, and Durability

Safety devices were utilized to maintain a zero accident project. A manual emergency stop was mounted on the back of CLETIS. A wireless emergency stop allowed for non-contact stopping of the robot. In addition, changes the program can be made without the motors engaged. This was beneficial because changes to parameters and corrections to the program during troubleshooting were accomplished and tested before the system has given control. The speed of CLETIS was ensured to be less that the 10 mph speed limit by choosing the appropriate gear and sprocket combinations at no load speed. The maximum loaded speed of the system is 9.69 mph.

The reliability and durability of CLETIS was improved using modularity and redundancy. CLETIS is a six-wheeled robot that has the ability to turn in place. The left and right drive module can operate independent of each other. Two motors are used for each gearbox; therefore, CLETIS has enough torque

to climb ramps greater than a 20° incline. CLETIS uses a double chain system to drive the wheels. Like the motors and chains, if a drive wheel becomes completely inoperable the system can still function, but at a reduced pace because there is another respective component that allows CLETIS to continue to function. No system is completely maintenance free; therefore, all the components deemed critical were designed and located to offer easy access for repair or replacement.

The feedback sensors used were placed in appropriate places to provide data even if the system had a malfunction occur in the mechanical design. The encoders were placed to take readings off of sprocket outside the gearbox as opposed to off the wheel gears; therefore, allowing for accurate speed readings even if chain derails off of one wheel. In addition, each drive module speed is read independently of the other and a gyro is used to provide angular changes.

5. Hardware

5.1 Innovations

The second approach used to extract distance information from the environment was through the use of an infrared depth sensor, an integral part of the Xbox 360 Kinect camera. The Kinect contains a variety of sensors that include an RGB camera, depth sensor, and microphone array. For obstacle detection and avoidance purposes, only the depth sensor was utilized. The depth sensor consists of an infrared laser projector combined with a monochrome CMOS sensor, which captures depth data under a broad range of ambient lighting conditions. A grid of infrared light is projected from the camera, reflected back, and collected at multiple points on the sensor. The duration of this process is proportional to the distance of the illuminated points on any obstacles present. In this manner, a medium-resolution „picture“ can be generated containing depth information in place of color information, images of which are shown in Fig. 6 and Fig 7.

Fig. 6: Depth image from the Kinect camera.

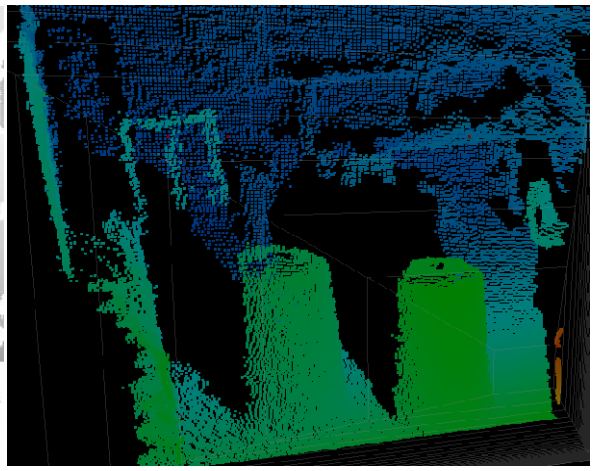


Fig. 7: 3-D rendering of the depth image.

However, this measurement system has inherent limitations. The infrared output of the sun has the potential to overwhelm the CMOS sensor. Physical filtering techniques were used to mitigate some of the interference, but outdoor operation during full daylight hours was found to be impractical. Thus, it was decided that the primary mode of obstacle detection and avoidance would rely upon the data gathered by the ultrasonic range finding sensors.

5.2 Electrical System

The sensors used interfaced with the USB-6009 data acquisition cards, but some custom circuitry was designed to give filtering capabilities that reduced noise in some of the sensors. A timing circuit for ultrasonic array was designed and implemented to mitigate the interference between each sensor. A voltage controlled pulse width modulation circuit was designed and implemented, however a more reliable commercial circuit was used to improve the reliability. A block diagram of the electrical system is shown in Fig. 8.

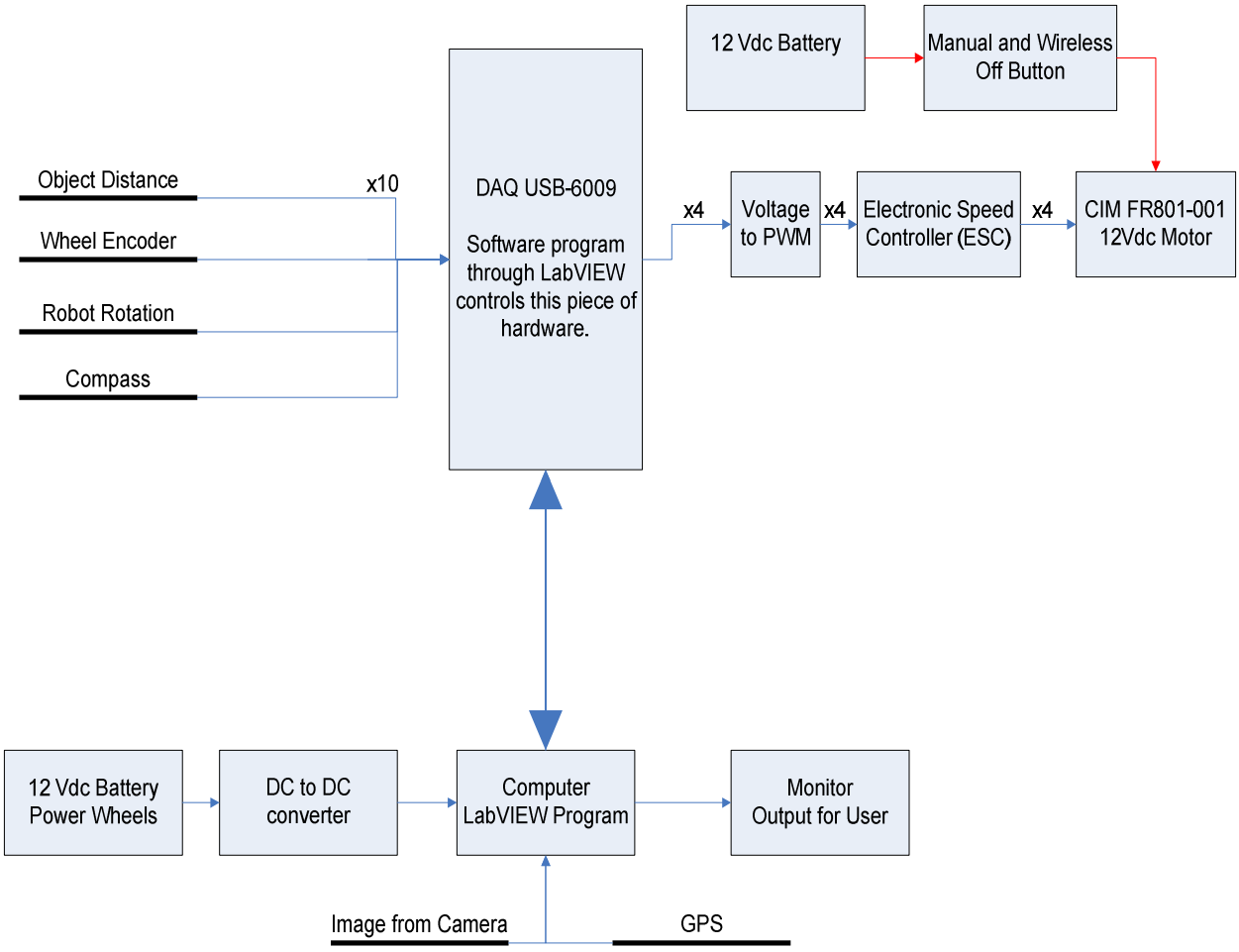


Fig. 8 Electrical Block Diagram

6. Software

6.1 Innovations

Global Positioning System (GPS) data is essential to provide the autonomous vehicle with simultaneous localization and mapping information. To begin with the GPS implementation into the system, it was first determined which unit would be best suited to provide the most accuracy while using a single GPS unit and adhering to budgetary constraints. The U-Blox 5 unit was chosen for its high accuracy, interface protocol, and the ability to utilize satellite-based augmentation system (SBAS) for GPS receiver error correction to account for anomalies. The specific type of SBAS correction provided by this unit is known as wide-area augmentation system (WAAS). The U-Blox receiver incorporated into the final system has a maximum error radius of two meters about the intended position. The U-Blox utilizes the standardized NMEA 0183 interfacing protocol. The returned data stream was easily integrated into the program and provides an acceptable update rate for accurate vehicle positioning. In addition to selecting the receiver with optimal accuracy and ease of interface from the satellite system, the algorithm used to determine the position and distance to the next waypoint must be considered. There are a variety of methods to approximate distance between two points given latitudinal and longitudinal global positioning data. The methods of approximation investigated were Vincenty's, Bowring's, Lambert's, and adaptive oblate spheroid formulas. These methods were implemented using global positioning data with and without angle compensation provided by an on-board gyroscope. The method that yielded the fastest and most accurate results was incorporated into the final system design which was determined to be adaptive oblate spheroid method. Shown in Fig. 9 are the average values of run time and accuracy measurements for each of the distance approximation methods. The average run times remained consistent across the multiple methods tested.

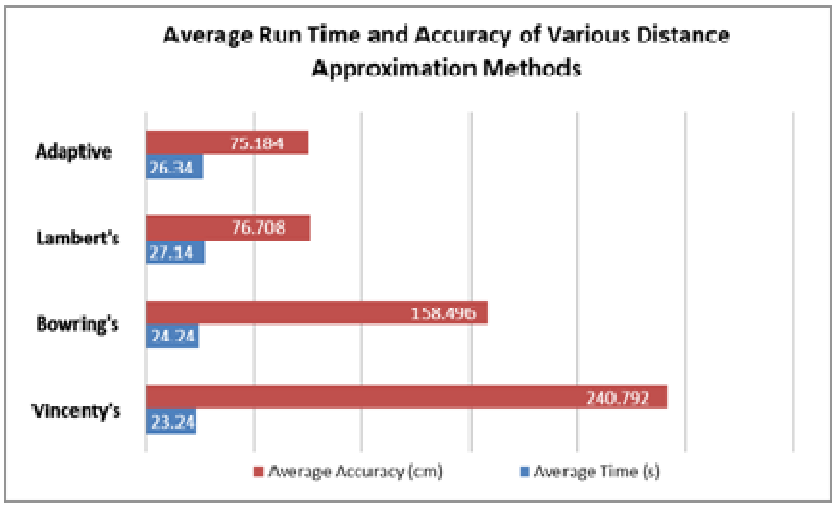


Fig. 9: Test data from testing the various distance approximation methods.

The adaptive method and Lambert's resulted in a similar system response while the remaining two methods were less accurate. Preliminary results indicated that GPS data with gyroscopic correction was superior in all cases compared to using GPS sensor information alone. Therefore, the data set was reduced to the GPS plus gyroscope condition which was evaluated using each of the four methods. The method that yielded the best combination of run time and accuracy over the trials conducted was the adaptive oblate spheroid method. This is the method that was incorporated into the finalized autonomous system as it has proven to yield optimal results and has an added advantage of coding simplicity. It is explained in further detail below.

The latitudinal and longitudinal data of the current position and destination waypoint are defined as phi and lambda, respectively. Although this formula is a spherical estimation of the distance, a simple formula was derived to determine the radius of an ellipse which utilizes the major and semi-major axes. This allows small parts of a two-dimensional ellipse to be transposed on a three-dimensional sphere which yields a simplified version of the distance calculation compared to iterative or complex methods of ellipsoidal distancing estimations.

$$r = \frac{(6378137)(6356752.3)}{\sqrt{\{(6378137*\sin[\phi_1\left(\frac{\pi}{180}\right)])^2+(63567523*\cos[\phi_1\left(\frac{\pi}{180}\right)])^2}} \quad (1)$$

$$Latitude\ Difference = \frac{\Delta\phi}{360} * 2\pi r \quad (2)$$

$$Longitude\ Difference = \frac{\Delta\lambda}{360} * 2\pi r \quad (3)$$

$$Straight\ Line = \sqrt{Lat\ Diff^2 + Long\ Diff^2} \quad (4)$$

Once the magnitude of the component vectors are found a root-sum-squared operation is used to find the straight line distance to the destination waypoint.

6.2 Strategy

The choice of the team was to use National Instruments LabVIEW programming environment. This was because the learning curve is fairly short and the ability to program many things in a short time allowed for fast prototyping and systems improvements quicker. The data was further improved through software by implementing averaging and low pass filters. Signal processing and conditioning allowed for better decisions to be made by the program and reduce some unwanted erratic behavior. In addition National Instruments provided a software architecture that was implemented into the design of CLETIS's program.

7. Systems Integration

7.1 Lane Following

In order to remain within a specified region of space defined only by white lines on the ground, the vehicle must be able to detect those lines, identify them as such, as well as define them as obstacles. This task was accomplished through the use of image acquisition hardware and image processing software. The image was acquired by an onboard color camera and transferred into software via USB interface. During image processing development, the acquired image went through multiple screens and filters. To effectively discern between the white lines from other color components, the blue color field was extracted from the image. The resulting extracted information was compared to that of the remaining image, and those components that fell below a particular threshold were removed.

What remained was an image of the line on the ground as seen by the camera. By providing a calibration image with known spatial values of previously designated positions on the image, each pixel location was converted into a real world distance.

Once the line information was isolated, the line location needed to be identified and associated with their real world positions. The image was partitioned into five distinct sections, four horizontal and one vertical. Each section was processed through a Hough transform to find the best fit curve of the acquired line. This broke the line into manageable line segments with the endpoints of those segments returned by the procedure.

Later in the paper, the Vector Field Histogram (VFH) and its mode of operation will be discussed in more detail, however, the information it requires is a distance, r, and an angle θ, associated with that distance, where 0° is straight ahead of the vehicle, and +90° is to the right and -90° is to left of the vehicle in 1° increments. With the defined degree array, the line segment endpoint information was converted into distance data that corresponded to the static angle positions. This array subset represented the surrounding environment of the vehicle.

When converting Cartesian coordinates into polar, the following equations apply:

$$r = \sqrt{x^2 + y^2} \tag{5}$$

$$x = r\cos\theta \tag{6}$$

$$y = r\sin\theta \tag{7}$$

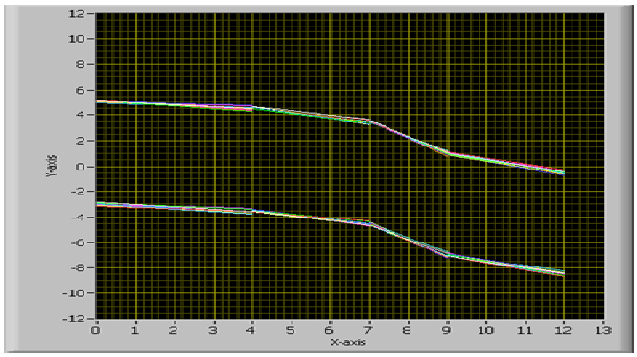
Through substitution and solving for r:

$$r(\theta) = \begin{cases} \frac{-b}{m\cos\theta - \sin\theta}, & \alpha < \theta < \pi \\ \frac{-b}{m\cos\theta - \sin\theta}, & -\pi < \theta < -\alpha \end{cases} \tag{8}$$

Where α is the angle of the line segment with respect to the unit circle. The range of angles to find the distance r of each line segment was determined by:

$$\tan^{-1}\frac{y_1}{x_1} < \theta < \tan^{-1}\frac{y_2}{x_2} \qquad (9)$$

Previously, a single line on each side of the vehicle was used, extending beyond the endpoints given. This introduced superfluous information overcomplicating the entire process and occasionally created errors at ±∞. By parsing the line into multiple segments, higher quality information about line location was acquired as can be seen in Fig. 10 and 11. If the vehicle encountered a curve, it was more easily identifiable by the distance algorithm. Additionally, by limiting the distance information to the angles specified in (9), the possibility of accidental calculation of discontinuities was effectively eliminated.



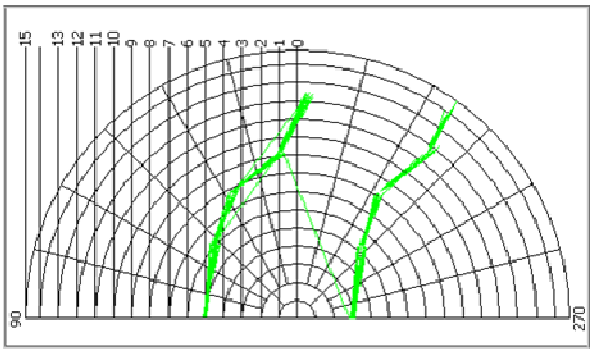


Fig. 10: Line segments interpreted from endpoint data Fig. 11: Polar plot of line data from function r(θ)

7.2 Obstacle Detection and Avoidance

For an autonomous ground vehicle to successfully navigate through an environment with unknown obstacles, distance information needs to be extracted from the surroundings. Two approaches were utilized to gather this data which include both sonic and infrared range finding. An array of ultrasonic range finding sensors were used that returned a voltage proportional to the distance from the source to an obstacle. The beam pattern of the array is shown in AutoCAD Inventor in Fig. 12.

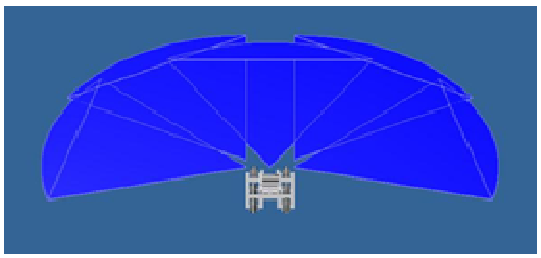


Fig. 12. Beam pattern of ultrasonic array.

The resulting raw data was further processed to yield an accurate real world distance that was then used to inform obstacle avoidance and motion control algorithms. Using an array of ultrasonic

sensors resulted in interference between the individual ranging echoes but was successfully mitigated by use of a timing circuit to periodically synchronize the multiple sonic pulses. Short duration, abrupt changes in returned distance information were identified and discarded using another software technique which involved analyzing the time rate of change of the incoming signal. Also, a moving average window was implemented in software to further improve the quality of distance data. These various signal conditioning techniques resulted in reliable distance data critical for obstacle detection and avoidance.

7.3 Waypoint Navigation and Mapping Technique

The navigation system is tasked with guiding the vehicle towards the goal location while preventing collision with obstacles. Design of a fast and efficient procedure for navigation of autonomous ground vehicles in the presence of obstacles is one of the paramount problems in robotics. Given the initial and final orientations of a mobile robot, the navigation algorithm must be able to determine whether there is a continuous motion from one orientation to the other, and find if such a motion exists [2].

Anytime Dynamic Star (AD*) is a heuristic-based algorithm used for mobile robot path planning. AD* combines the anytime algorithms, including the recent Anytime Repairing A*algorithm [6] with the incremental re-planning algorithms of D* and D* Lite [4].

The anytime algorithm is a process that returns a possible solution even if interrupted before the end. The solution of the anytime algorithm is an approximation [6], and the longer the algorithm runs the more optimal the solution becomes. This type of process allows for a continual solution in a dynamic environment.

The incremental re-planning uses previously selected paths to narrow the search parameters which results in decreased processing time. The incremental planning algorithm moves the vehicle from its current location towards a defined goal location. The locations are assigned on a grid [5], and the vehicle moves from the current location to the goal location along the shortest path when its adjacent grid block changes cost values as can be seen in Fig. 13.

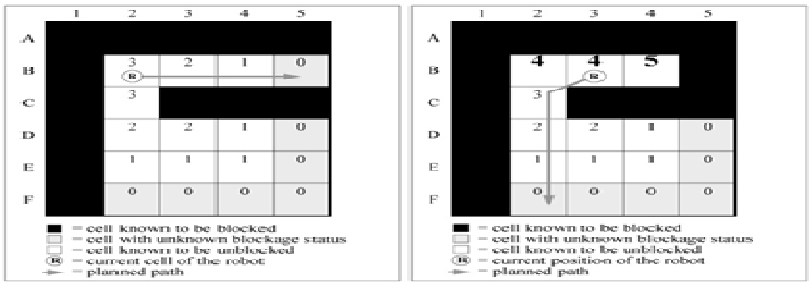


Fig. 13: Example Mapping Task.

The vehicle uses AD* to develop a computer generated map of its past and current real world environment. A start location and goal location are given in x and y coordinates. It takes sensor data from the camera, ultrasonic rangefinders, and Kinetic camera depth field. This data is fused together to create a relative environment to the robot. This relative environment assigns particular cost values to a localized grid. This grid is stored in the Cartesian Coordinate Grid (CCG) which is overlaid on a static global reference frame. When the vehicle changes position, the localized grid adjusts accordingly and updates. The CCG. AD* computes the optimal path to the goal location, and each grid space on the CCG is a defined real world distance (ex. 0.5m). As the vehicle approaches the goal location, then the new goal location is incremented, and the robot proceeds planning to the next goal; seen in Fig. 14. This process is continued until the robot reaches all waypoints and returns to the start.

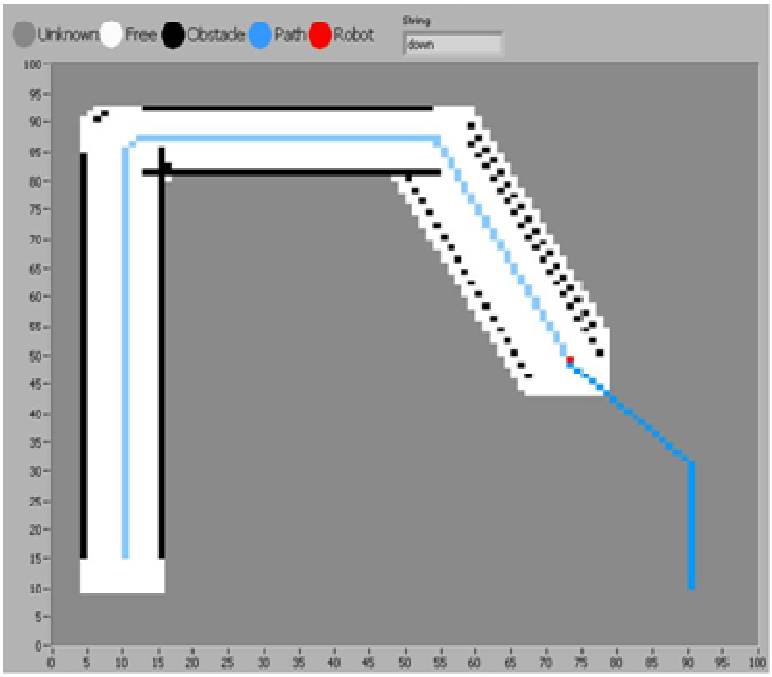


Fig. 14: Example of incremental goal achievement.

The path planning capabilities of AD* provides quicker times to a solution and converging results. Additionally, the localized reference grid was used to provide real time relative mapping for the vehicle in conjunction with the vector field histogram. This improvement allows for updates to the CCG when the vehicle needs to remap over a previously surveyed area. It remaps the area without replacing the data that is out of range of the sensor data acquired in the past; therefore an accurate map is created and stored.

The Vector Field Histogram (VFH) and Advanced Vector Field Histogram (AVFH) is a common way to use object distance data for collision avoidance. In addition, the VFH and AVFH can provide the direction to

the largest cleared area within the vehicle sensor's field of view. The AVFH can be given additional information about the target location, which in turn provides a better directed angle to the goal location.

The VFH method uses a two-dimensional Cartesian histogram grid as a relative world model. This model is updated at a 20 Hz rate with range data from on-board ultrasonic rangefinders and 30 Hz rate from the camera. The VFH method uses a two-stage data reduction process to compute the desired direction angle to the center of the gap. The first stage of the histogram grid creates a one-dimensional distance histogram overlaid onto the vehicle's momentary location. The sections of the histogram are divided into 1° sections. The second stage of the algorithm selects the largest section with the obstacle distances outside the defined threshold. The angle to the center of the identified gap is then determined and used as the vehicle's relative desired angle of travel [5].

The AVFH method is similar to the processes used by the VFH, but allows goal location data to be incorporated. The target tracking is performed when the vehicle encounters an obstacle within the defined threshold distances. The algorithm then calculates a new heading when the heading is blocked because an obstacle is within an inner threshold distance. The AVFH takes in to account sensor distance noise. The angle section is determined free of obstacles when the sensors show the obstacle is outside the outer threshold distance. The VFH is used to implement reactionary movement without a goal.

The simulation in Fig. 15 represents an autonomous vehicle that encounters an obstacle within its inner threshold distance and shows the open (green) and blocked (red) angles that result. The histogram data represents the position at which the object becomes an obstacle given the obstacle clearance distance of the robot, and the sensor data represents the obstacle itself [6].

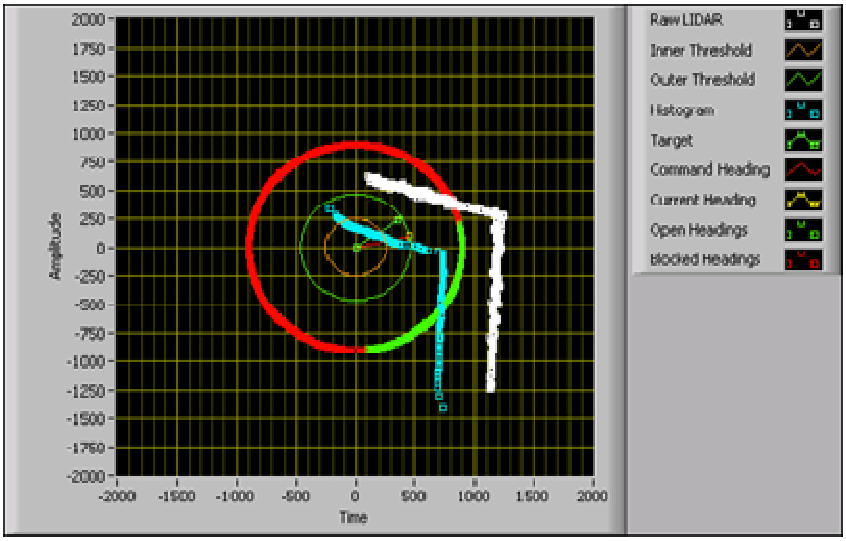


Fig. 15: Example of Advanced Vector Field Histogram.

9. Bill of Materials and Budget

Components	Distributor	Part Number	Quanity	Price Per ($)	Amount ($)
Labview Professional	National Instruments Corp.	776678-35	1	$ 4,300.00	$ 4,300.00
Robotics Module	National instruments Corp.	781220-35	1	$ 2,000.00	$ 2,000.00
IFI Speed Controller Victor 884	Innovation First Intl, Inc.	VICTOR	4	$ 100.00	$ 400.00
Andymark Tough Box	Andymark, Inc.	AM-0145	2	$ 88.00	$ 176.00
Connectors 45x45mm	Rexroth	93543-07	40	$ 3.00	$ 120.00
Extruded Aluminum 45x45mm	Rexroth	74351-22	25	$ 3.00	$ 75.00
Chain	Northern Tool	136410	12	$ 1.00	$ 12.00
Motors	Innovation First Intl., Inc.	CIM FR801-001	4	$ 28.00	$ 112.00
Robot Control System IFI	Innovation First Intl., Inc.	ROBOT-RC	1	$ 1,200.00	$ 1,200.00
Ultrasonic Sensor	Acroname Inc.	R93-SRF04	1	$ 30.00	$ 30.00
Camera	Super Circuits	PC88WR-2	1	$ 100.00	$ 100.00
GPS GS407	Sparkfun	GPS-09436	1	$ 70.00	$ 70.00
Battery	West Marine	1231109	1	$ 150.00	$ 150.00
Bearings	Igus	EFOM-BB1-P12-101	12	$ 12.00	$ 144.00
Wheels w/ Sprocket	Northern Tool	13340	4	$ 40.00	$ 160.00
Wheels w/o Sprocket	Northern Tool	136411	2	$ 10.00	$ 20.00
Monitor	Newegg	N82E16824114004	1	$ 200.00	$ 200.00
Computer	Newegg	N82E16856107072	1	$ 500.00	$ 500.00
DAQ USB 6009	National Instruments Corp.	USB-6008	1	$ 280.00	$ 280.00
Accelerometer	Innovation First Intl., Inc.	DAA BD	1	$ 30.00	$ 30.00
Gyroscope	Systron Donner Inertial Div.	QRS14	1	$ 35.00	$ 35.00
Proximity Sensor	IFM Electronics	IF5905	4	$ 85.00	$ 340.00
Lexan Side Panels	Lowes	1PC0036A	1	$ 70.00	$ 70.00
DC to DC Power Converter	Mini-box	M4-ATX	1	$ 102.56	$ 102.56

Mounting Hardware

Components	Distributor	Part Number	Quanity	Price Per ($)	Amount ($)
Vesa Mount	McMaster-Carr	M4x12	8	$ 0.63	$ 5.04
Computer Bracket	McMaster-Carr	M5x16	4	$ 1.25	$ 5.00
Victor Mounts	McMaster-Carr	M3x22.23	8	$ 0.63	$ 5.04
Power Block	McMaster-Carr	M5x14.29	2	$ 2.50	$ 5.00
USB-6009	McMaster-Carr	M3x17.46	2	$ 2.50	$ 5.00
Gyro Mount	McMaster-Carr	M6x22.23	4	$ 1.25	$ 5.00
Ultrasonic Mount	McMaster-Carr	M3x9.53	14	$ 0.36	$ 5.04
DC to DC Power Converter	McMaster-Carr	M5x9.94	4	$ 1.25	$ 5.00
Toughbox	McMaster-Carr	M6x55.56	8	$ 0.63	$ 5.04
Main Disconnect	McMaster-Carr	M6x20.64	2	$ 2.50	$ 5.00
Power Cube	McMaster-Carr	HNG-90	4	$ 4.00	$ 16.00
				Total	$ 10,692.72

The capstone senior design course only allowed a budget of $800. This designed constraint made it very difficult to purchase expensive components.

8. Conclusion

As the field of robotics advances and the associated applications thereof become more numerous, systems will need to have the capability of carrying out specified tasks with an ever increasing degree of autonomy. The best approach towards the realization of the autonomous ground vehicle described previously was to use a variety of sensors to construct an accurate internal

representation of the surrounding environment. This was accomplished by integrating the data from multiple ultrasonic rangefinders, video camera, gyroscopic compass, wheel encoders, and GPS into a path planning algorithm in order to make intelligent navigational decisions. Off the shelf parts incorporated into the modular design scheme allows for ease of interchangeability of components thereby reducing the cost of maintenance and downtime.