APOLLO II: IGVC 2016 Design Report

Bluefield State College
Submitted: May 14, 2016

Team Captain:

Levi Poff	levipoff@yahoo.com

Team Members:

Charles Reeves	charles.m.reeves@gmail.com
William Lambert	lambert_cw@live.bluefieldstate.edu
Michael E. Goforth	goforth_me@live.bluefieldstate.edu
Ian Fields	fields_im@live.bluefieldstate.edu
Samuel Stevens	stephens_sr@live.bluefieldstate.edu
Jesse Edwards	haynes_je@live.bluefieldstate.edu
Brandon Tolley	tolley_bn@live.bluefieldstate.edu
Nathaniel Blankenship	blankenshipnathanielian@gmail.com
Chris Parkinson	parkison_cg@live.bluefieldstate.edu

Statement of Integrity from Dr. Robert N. Riggins:

I certify that the design and engineering of Apollo II by the 2015/2016 BSC robotics team has been significant and equivalent to 5 credits in a senior design course.

Robert N. Riggins

Dr. Robert N. Riggins
Faculty Advisor

1 INTRODUCTION

Bluefield State College is proud to enter Apollo II into the 24th annual IGVC. Apollo II, seen in Figure 1, is an improved version of Bluefield State College's 2015 entrant, Apollo. Apollo II boasts many improvements over its predecessor, such as a new control system, greatly enhanced safety features, and various software improvements. Like Apollo, Apollo II is an intelligent vehicle designed for the autonomous navigation of an obstacle course. To accomplish this, Apollo II comes equipped with an array of sensors, which includes a wide-angle camera, a laser measurement system (LMS), a digital compass, and a global positioning system (GPS). All these sensors interface to an internal laptop running LabVIEW. Apollo II has kept the innovative fiberglass body that was implemented on its predecessor, Apollo. This body is light-weight, very durable, and weather resistant. It is anticipated that Apollo II will be a strong competitor this year at IGVC.

Figure 1: APOLLO II

2 DESIGN PROCESS

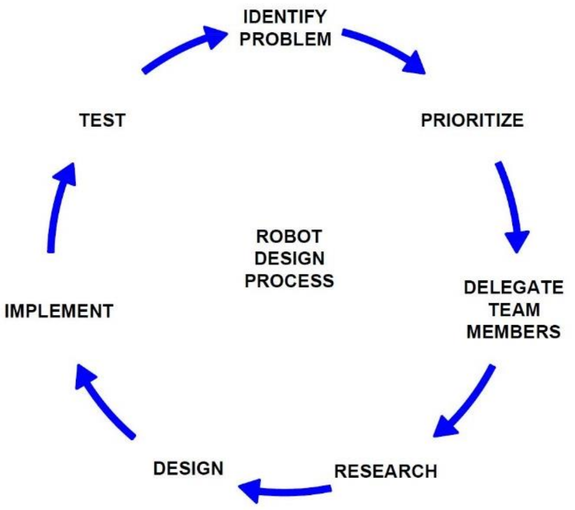

Figure 2: Design Process for Apollo II

The design team, seen in Table 1, for Apollo II used a 7-step strategy when designing the robot. The seven steps, illustrated in Figure 2, are: identify the problem, prioritize tasks, delegate team members to each project, research project solutions, design a solution, implement this design on the vehicle, and then test this solution. The first step in our design process is to identify the problem. This step usually occurs immediately after IGVC, when the team analyzes the problems faced during the competition and brainstorms solutions to these problems. The next step in our design process is to prioritize tasks. This involves determining the importance of a particular task and deciding whether we should pursue a solution or not. The next step is to delegate team members to each task. The number of students assigned to each task will vary depending on the importance of the task, the time the task will require, and the amount of students that are available. After the project has a design team assigned to it, the team can then move on to the research step of the design process. This step involves researching possible solutions to the problem and determining that we have the best answer. After we determine a possible answer, we then plan a design suitable for Apollo II. During this phase, we use information gained from the research step to design an implementation. The next step involves implementing this design on the robot and ensuring that everything is operational before moving on to testing. In the last phase, we rigorously test our implementation to ensure that it works correctly and efficiently. If test results are not optimal, we cycle back to the problem identification step and restart the cycle.

2.1 Design Team

Table 1: Apollo II Design Team

Team Member	Academic Major	Mechanical	Electrical	Software	Hours
Levi Poff (Team Captain)	Computer Science	X	X	X	320
Charles Reeves	Electrical Eng.	X	X	X	318
William Lambert	Mechanical/Electrical Eng.	X	X		240
Michael Goforth	Mechanical/Electrical Eng.	X	X		125

Jesse Edwards	Mechanical Eng.	X			56
Ian Fields	Mechanical/Electrical Eng.	X	X		40
Brandon Tolley	Mechanical/Electrical Eng.	X			30
Nathanial Blankenship	Electrical Eng.		X		20
Chris Parkinson	Electrical Eng.		X		5
				Total	1154

2.2 Cost

Apollo II is comprised of parts that have been purchased and donated to the Bluefield State College robotics lab. The costs that are shown in Table 2 are a representation of what the Robotics lab has spent on the Apollo II intelligent ground vehicle.

Table 2: Apollo II Cost

Item	**Cost**
Jazzy Wheel Chair Bottom	$1,031
Basler USB 3.0 Camera + Lens	$1,800
Maretron Solid State Compass	$600
2 Yellow Top 12V Batteries	$240
Dell Latitude Laptop	$1,800
Hemisphere GPS	$2,400
Hokuyo LMS	$5,600
Body Fabrication	$280
SaberTooth Motor Controller	$180
2 XBees and Antennae	$120
3D Printed Components	$20
Miscellaneous	$1,100
Grand Total	$15,171

3 INNOVATIONS

3.1 New Control System

Figure 3: Sabertooth Motor Controller

The biggest change from our previous robot was the creation of a new control system. Apollo II's drive chassis is an electric wheelchair bottom. The previous system interfaced with the wheelchair controller and joystick to control movement. During last year's competition, we had an issue related to this controller that became very time consuming to fix. It was after this incident that we decided it was best to move away from the original wheelchair controller and replace it with a highly customizable Sabertooth motor controller. Figure 3 shows an example of a Sabertooth motor controller. The new Sabertooth system gives us much more freedom in controlling our robot's dynamics. The old controller had safety systems that were very restricting to our performance, such as limiting our speed and dynamics controls. These safety systems were designed for a person riding in the wheelchair, and as such, were no longer needed. The Sabertooth's safety features are much more in line with our needs, such as stopping the motors if no signals are received and allowing us to control ramping speeds. The Sabertooth is also easily replaceable should something go wrong, whereas the old controller would be very difficult and expensive to replace.

3.2 New Ergonomic Wireless Manual Controller and E-Stop

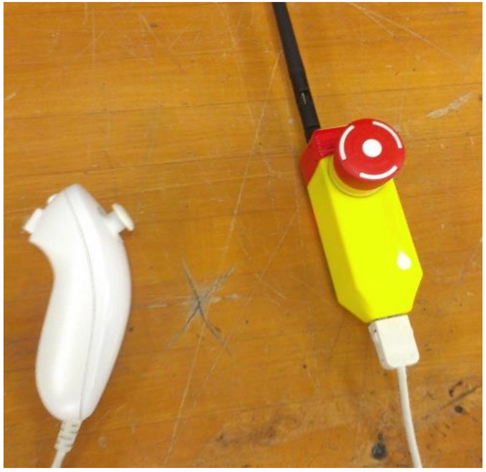

Figure 4: Manual Controller with 3D Printed Enclosure

When we replaced our old wheelchair controller, we also lost our ability to control Apollo using the wheelchair's joystick. It was for this reason that we created our new innovative controller. Our controller communicates with the vehicle via two XBee modules. XBee was chosen because of its ease of use, low power consumption, small size, and that it greatly exceeds the required range for IGVC. The XBee is connected to an Arduino processor, which processes movement commands received from a connected Wii Nunchuk controller. The Wii Nunchuk controller is cheap, ergonomically designed, and has the precise number of buttons we require for the operation of Apollo II. Figure 4 is a photo of the finished controller.

We also implemented a new wireless E-Stop system with this controller. Our old system was dated and did not meet our quality standards. We also needed to carry around two devices during operation: the E-Stop and the manual controller. Since this new system combines both controller and E-Stop, only one device is required. The Wii Nunchuk can also be removed from the controller, which alters the controller to function only as an E-Stop. With this new system, there is a lot of room for expansion. For example, the XBees can easily create a many-to-one network in which we can have multiple E-Stop systems communicating with the robot. This helps to ensure our robot is as safe as possible.

3.3 3D Printed Components

Some of Apollo II's enclosures have been 3D printed by the team. This allows us to fabricate parts that meet our needs exactly. These parts are light-weight, inexpensive, nonconductive, and easy to fabricate. 3D printing our own parts also saves us time and money which can then be spent on other important components. Examples of parts we have 3D printed include: the enclosure for our manual controller, an enclosure for the Sabertooth motor controller, and a mount for Apollo II's GPS module. Figure 5 shows our 3D printed Sabertooth enclosure. For all of these cases, it would have been very difficult to find parts that exactly meet our needs without wasting space or funding. These parts are also easily replaceable as we can simply print another if the need arises.

Figure 5: 3D Printed Sabertooth Enclosure

3.4 Opto-isolation

In order to isolate the high power motor and brake connections from the low power control system, we added opto-isolators. Using opto-isolators has the added benefit of minimizing counter-EMF and transient signals feeding back into our processor. For instance, we have an opto-isolator in the brake monitor circuit to isolate the brakes from the controller. In case the brakes are applied simultaneously with motor commands it will stop the motor controller by providing an opto-isolated signal to the controller. Figure 6 shows a simple opto-isolation circuit.

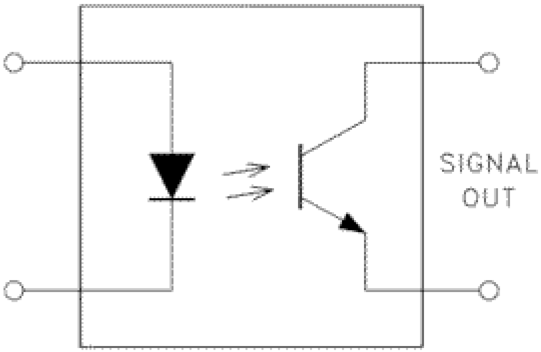

Figure 6: Simple opto-isolater circuit, with an LED opening or closing a separate circuit.

3.5 Software Modularity

Every piece of code in Apollo II's software is designed to be modular. All information in the program is kept together in a global information cluster that travels from subprogram to subprogram. Each subprogram takes this cluster as input, accesses the data in this cluster, and then outputs this same cluster. Since all of our code has the same input and output, we can easily swap in or take out code without having to worry about the previous step's output or the next step's input. Using this strategy allows multiple programmers on our team to write programs simultaneously without needing to know the specifics of any other's program. One way we are taking advantage of this is with our navigation programs. We currently have 3 different navigation programs each running a different algorithm that we can easily swap between to meet the situation.

3.6 Software Simulation

With Apollo II being down for long periods of time, due to the changes to the control system, we required more sophisticated simulation software in order to test. With our new simulation program we can now create our own course and simulate Apollo II's navigation through this course the same way it would react to actual data. This allows us to test various aspects of software even when Apollo II is non-operational. The simulation also allows us to test on courses and scenarios that would not be possible otherwise. Figure 7 is an example of our simulation software running on one of our user-designed courses.

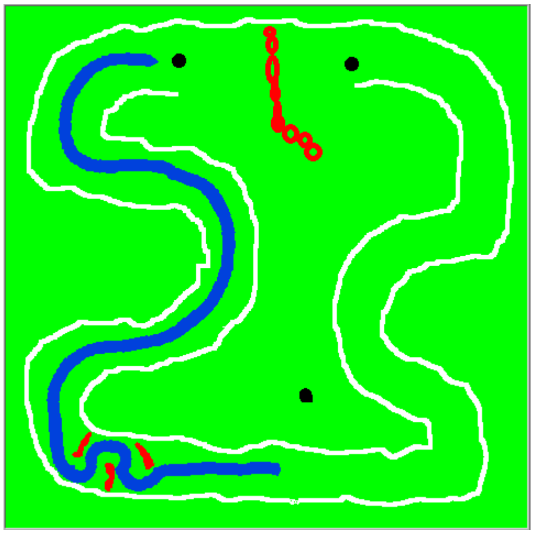

Figure 7: Apollo II simulation running on user generated course. The blue triangle at the top is Apollo's current position while the blue line is the path Apollo has taken. Black circles are waypoints and red objects are obstacles.

4 MECHANICAL DESIGN

4.1 Overview

Apollo II's mechanical design focuses on being lightweight, modular, highly maneuverable, and easy to assemble and disassemble. Apollo II consists of two main integrated mechanical fabrications: the drive chassis and the body. The drive chassis has been highly modified from an electric wheelchair bottom that allows for high speeds and zero-degree turns. The body is fabricated from wood and fiberglass and is designed to be light-weight, strong, and weather resistant. Overall, the key components of Apollo II's mechanical design encompasses simplicity without any trade-off in performance.

4.2 Drive Chassis

Apollo II's drive chassis, as shown in Figure 8, is developed from a Jazzy electric sports wheelchair base. This model allows Apollo

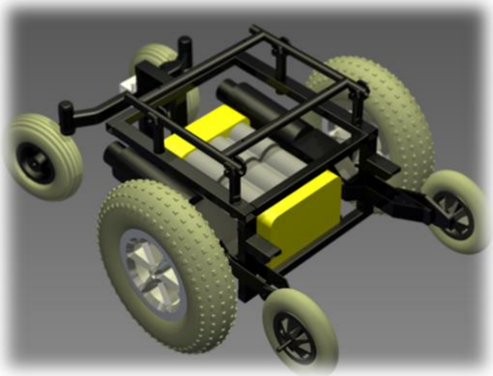

Figure 8: 3D Drawing of Apollo II's Drive Chassis

II to achieve the upper limits of competition speed without sacrificing maneuverability. Also, the suspension allows for traversal of various terrains and environments. The frame has been modified to accommodate a student-designed and built battery tray. This tray holds our two 12V batteries and slides out for easy access. We have also added a payload slot to the front of the frame that will accommodate the 20-pound payload.

4.3 Body

Apollo II's body is student-designed and fabricated out of wood and fiberglass. It is strong, light-weight, and its modularity allows for easy assembly and disassembly. Figure 9 shows Apollo II's body during various stages of its development. It has a curved shape, which makes it safer by not having any sharp edges. This shape also helps with weather resistance, as water will follow the curvature and run off the robot. The body is coated in a professional automotive white paint and clear gloss finish which aids in the reflection of radiant heat from sunlight. This, coupled with an internal fan, helps keep all of our electrical components cool during operation. Having a light-weight body means that the motors do not have to work as hard to sustain full mobility. This reduces drain on the batteries, allowing Apollo II to run for hours. The body's top portion can effortlessly be removed, without tools, providing access to all of Apollo II's internal components. The body also has a modular design so parts can be added or removed depending on situational demands. For example, the mast is connected to the body via a simple internally reinforced PVC pipe connection that can easily be swapped out for another mast design or different component altogether. This is simply one example of the many ways in which Apollo II's design allows for flexibility in the field.

Figure 9: Top: Apollo II's Wooden Frame; Bottom: Fiberglass Exterior is Added

4.4 Cooling

Apollo II is required to operate in the sun for long periods of time so it is critical that we keep our internal components cool. Apollo II's body is designed to meet this goal. Its white color and reflective finish redirect heat away from the robot. We have also installed a high static pressure fan in the bottom portion of the body, underneath all our internal components. This fan circulates air around the laptop, electrical components, and sensor systems to aid in cooling. Also, it pushes air out of every crevice where air can escape, which helps with weather proofing.

4.5 Water Resistance

Apollo II was designed to be able to operate in various weather conditions. To meet this end, Apollo II's body was designed with a curved shape in mind. This curved shape causes water to roll off of the robot and away from areas susceptible to moisture damage. The fan additively discourages water entry by driving air between every seam in the body. The body was kept to a minimal number of parts but where there are assembly points, weather stripping is incorporated.

5 ELECTRICAL DESIGN

5.1 Overview

The electrical system of Apollo II has been overhauled completely from the previous rendition of Apollo. The original wheelchair control system was removed and a completely new design was integrated. The control system consists of an XBee wireless transceiver, Parallax Propeller 8-core microprocessor, and a Sabertooth 2 x 60 motor controller all receiving information from a full suite of sensors. Various safety features are also an important part of the new system. The focus of this design is on safety, reduction of power consumption, and fully customizable dynamics control.

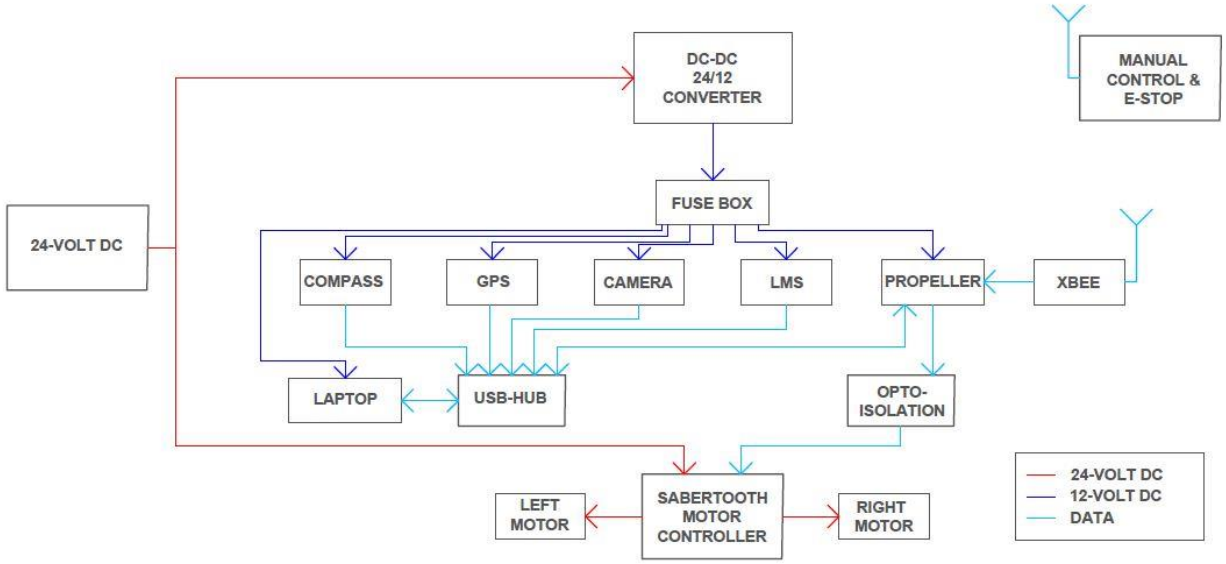

Figure 10: Apollo II Electrical Diagram

5.2 Power System

All components on Apollo II are powered by a single source: two Optima Yellow-Top 12VDC batteries connected in series to produce 24VDC. As Figure 10 shows, the 24VDC power supplies the motors and brakes through the Sabertooth motor controller, and supplies the 24-to-12V DC-to-DC converter. The 12VDC output of the converter supplies all the other components on Apollo II. Having a single source for everything, coupled with an easy "plug and charge" system makes charging Apollo II simple. Battery life between Apollo from IGVC 2015 to this year's Apollo II has doubled due to the complete overhaul of the control system and the new computer system. A new power distribution unit with switches also aids in power savings using an integrated switch panel for all sensors and devices. Only sensors being used during testing are turned on, allowing for quick disconnecting/connecting during testing. The motor controller employs synchronous regenerative braking to recapture energy during downhill descents and braking and also reduces heat.

5.3 Safety Systems

Apollo II has many safety features. For example, we have four independent ways to stop the robot quickly. Two physical switches reside on the body of the robot itself: a soft E-stop gives the processor a software stop command and the hard E-stop physically cuts power to all motors and sensors. A wireless E-stop sends stop commands well exceeding the minimum requirement of 100 ft. There is a firmware timeout should the Sabertooth fail to receive a valid signal within 100 milliseconds that will also stop the robot. Another safety feature the Sabertooth provides is the ability to limit our maximum speed. We have also added to Apollo II a separate brake/motor monitor circuit. Anytime the brakes and motors are not synced properly, the multi-core Propeller microcontroller will command the robot to stop. This scheme is particularly effective since the brake/motor monitor is controlled by an independent core of the Propeller processor. Figure 11 shows these safety features.

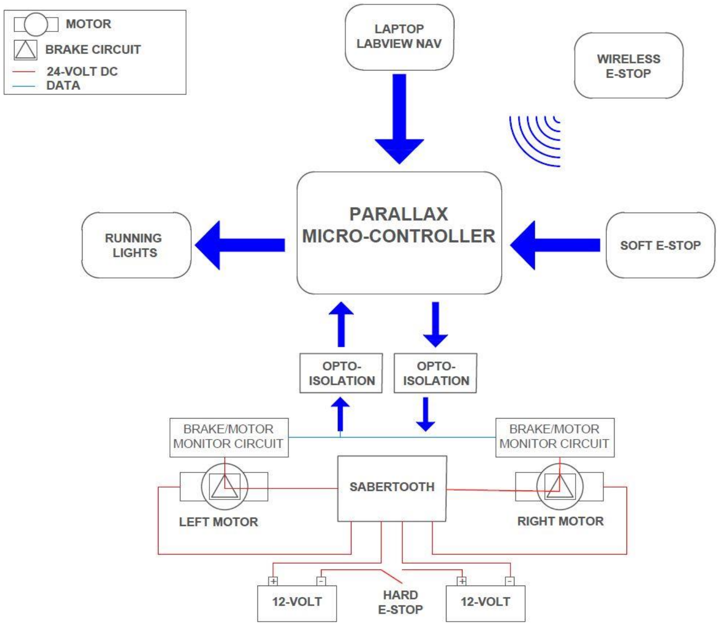

Figure 11: Apollo II Safety Systems Diagram

5.4 Control System

As mentioned in the Mechanical Design section, Apollo II's bottom was originally an electric wheelchair. As such, the original control system was the wheelchair controller, complete with its own joystick and other components. The biggest electrical challenge of the year was the complete overhaul of this control system to a new and improved system, designed by our robotics team. The new control system was designed around the Parallax Propeller 8-core microprocessor coupled with a Sabertooth Motor controller. Extra consideration was put into transient suppression and electrical isolation of parts to prevent future component failure. We used opto-isolators between the motor controller and microprocessor to physically separate the two assemblies since they operate on different levels of voltages and currents. A ferrite core toroid on the motor wiring harness reduces transients from reverse voltage of the motors. The brake relay uses a flyback diode to reduce transients, a capacitor to reduce arcing, and an opto-isolator ensures operation of the brakes while keeping the higher voltage isolated from the microprocessor. The control system is depicted in Figures 10 and 11.

5.5 Sensor Systems

Apollo II uses four sensors: an LMS, a camera, a GPS receiver, and a compass. These sensors are used to interpret information from the outside world, and were chosen by the team for their proven accuracy and speed, perfect for performing at IGVC. These sensors are described below:

- **Camera**: A Basler USB 3.0 outdoor camera, as shown in Figure 12, has been selected for multiple reasons. With a frame rate of 90 frames per second and high-speed data transfer, this camera provides more than enough speed for all of our needs. Additionally, automatic white balancing, gain adjustment, and shutter speed control allow for excellent vision in any lighting condition. When combined with a horizontal 125-degree field-of-view lens that has only 3% distortion, the Apollo II vision system has exceptional precision and versatility.
- **Laser Measurement System (LMS)**: Apollo II uses a Hokuyo LMS, seen in Figure 13, for object detection. With a 270-degree field-of-view at 0.25-degree increments and a detection range of up to 30 meters, the LMS provides extremely accurate object detection. It cycles at 40 Hz, allowing ample time for Apollo II to detect any obstacles in its path. The LMS also features data clustering, specular measurement, and adjustable resolution levels for maximum customization.
- **GPS**: To obtain positioning data, Apollo II uses a Hemisphere GPS and the A21 antenna (L1, GNSS, L-Brand) from Blueplanet Geomatics shown in Figure 14. This provides position, direction, and speed data, allowing Apollo II to track both its own position and those of user-defined waypoints. The GPS antenna and the A21 antenna used for differential corrections are housed in the same

Figure 12: Camera

Figure 13: Hokuyo LMS

Figure 14: Hemisphere GPS

location. This GPS unit runs at 20 Hz, making it easy for Apollo II to navigate through the course at high speed.

- **Compass**: A Maretron Solid State Compass, shown in Figure 15, assists in determining vehicle heading. Since the heading data provided by a GPS unit is less than reliable when the vehicle is stationary or moving at low speeds, it is supplemented with this compass. It provides an accuracy of 0.1 degrees, and updates at 100 Hz to verify our direction. The Maretron compass is designed to function with pitch and roll up to 45 degrees, preserving its functionality on inclines.

Figure 15: Maretron Solid State Compass

6 SOFTWARE STRATEGY AND DESIGN

6.1 Overview

Apollo II's software is developed using LabVIEW. Using LabVIEW allows us to easily create a sophisticated graphical user interface that makes it simple for us to monitor data, change settings, and debug our software. LabVIEW also has a visual programming environment that is very familiar to our electrical team members, which allows more of the team to be able to write code. Apollo II's software strategy revolves around mapping sensor data to an 80 x 80 2D grid of weighted nodes that represent the area around the vehicle and is used in path planning. Obstacles are detected by the LMS and lines detected by our vision system are mapped accordingly to our 2D grid map. Location and heading information received from our GPS and compass are used, in combination with the obstacle data in the grid, to select a goal on the map that will progress the robot to the next waypoint while still avoiding obstacles. The path planner is then used to create a safe path from the robot to the goal using a weighted shortest cost path equation. The last step is to smooth out the path, which determines the commanded heading and speed of the robot.

6.2 Obstacle Detection and Avoidance

Apollo II uses a Hokuyo laser measurement system (LMS) to detect obstacles. This LMS scans for obstacles in a 270-degree field-of-view and can detect obstacles up to 30 meters away but our software limits this range to 7 meters. Detected obstacles are then mapped to our software map, a 2D grid representing the area around Apollo II. Obstacles are mapped in relation to Apollo II using the angle and range data provided by the LMS. After this data is correctly added, our map will contain Apollo II's location, empty passable squares, and squares representing mapped obstacles, which are not traversable. Obstacle avoidance is then done by the goal selection algorithm and the path planner. The goal selection algorithm takes into account the obstacle locations and the gaps between obstacles. The goal selection algorithm is more likely to choose goals near large gaps over those near smaller gaps. After a suitable goal has been selected, the path planner creates a path from the robot to the goal. This algorithm looks at the weights of each node it passes through to make sure it chooses the most efficient path. By adding additional weight to nodes adjacent to obstacles, the path planner is more likely to choose a path that distances itself from obstacles. Another thing we do to aid in obstacle avoidance is adding "fat" to the obstacles on the map. This treats the layers around obstacles as obstacles also. This means that the robot thinks that the obstacles are bigger than they actually are, which makes the robot stay further away from the real object.

6.3 Line Detection

Line detection is done by processing the vision data received from our Basler camera. We detect the lines by looking at the red green blue (RGB) values of each pixel and mapping the white pixels to the software map. The first step in this process is to mask out the robot, as the robot is white and would give us false positives. After the robot is successfully masked out, the next step is to color threshold the image to create a binary image. This binary image will be two colors, black and white, white being pixels that are detected as white and black is any other pixel. While this detects the white lines it also detects a great deal of noise that needs to be removed in order to use this image effectively. We reduce this noise by removing small objects in the image. After the noise is removed, we can then map the detected lines to the software map as obstacles. Figure 16 shows our line detection strategy as well as the images during various phases of image processing.

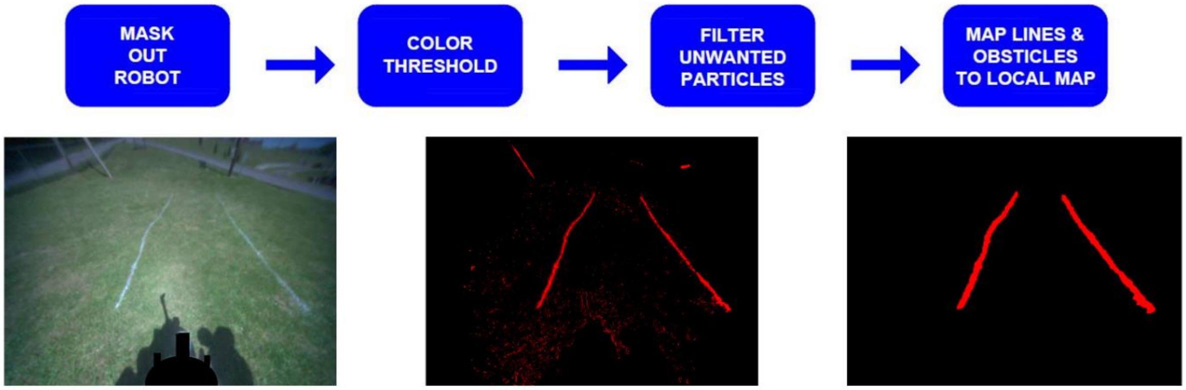

Figure 16: Apollo II Line Detection Strategy

6.4 Navigation

Apollo II's navigation focuses on creating a path from the robot to a specified goal in our software map. Our software map is an 80 x 80 2D grid that is comprised of nodes that represent the robot, the goal, and any detected obstacles. These nodes represent 0.1 meters each and all have a weight associated with them based on if they are close to obstacles and how close they are to the goal. A path is generated to the goal by adding up the weights of the nodes on the path and selecting the most efficient path. Figure 17 illustrates the stages of our navigation strategy as well as our software map during each stage.

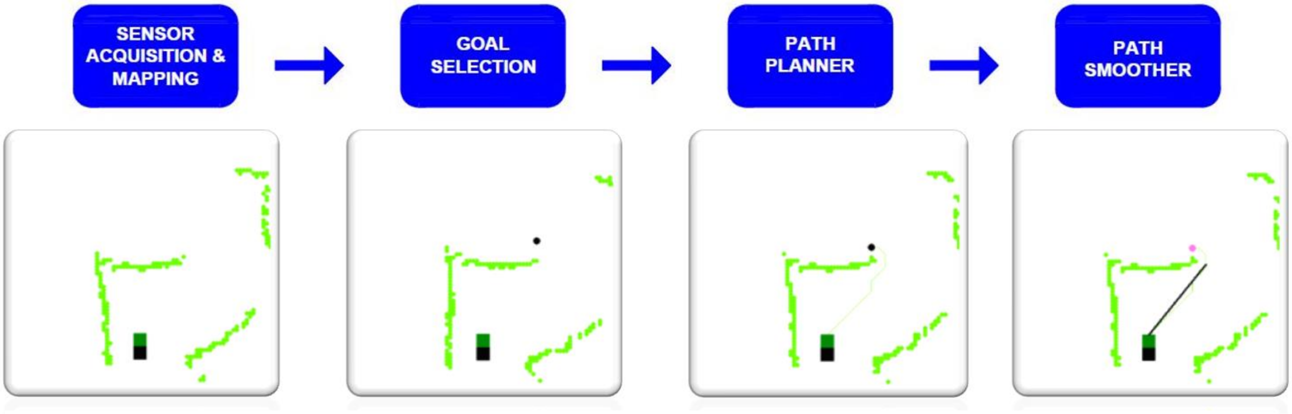

Figure 17: Apollo II Navigation Strategy

The first step in our navigation strategy is to map our sensor data to our software map. This step will add obstacles to our 2D map. The next step is to select a goal. The goal selection program uses a weighted algorithm to analyze several candidate goals on our software map. Each candidate gets a weight based on user-defined weights and the goal with the most weight is selected. Some weights include: how close this candidate is to the next waypoint, if this candidate is between a large gap between obstacles, and is this goal near a lot of obstacles. We can easily modify how our goal selection program selects goals by changing the weights on our graphical user interface (GUI). After a goal is selected, we then go to the path planning program. Our main path planning algorithm uses A* (A "star") to create a path from the robot to the goal. A* is a graph traversal algorithm that uses weights to find the shortest or most efficient path between two nodes on the graph. It uses Dijkstra's algorithm with an added heuristic function to keep the program from expanding in all directions. Our path planner begins by giving each node a weight based on its distance from the goal. Then, additional weight is given to nodes that are close to obstacles so that the path is more likely to distance itself from obstacles. After all the nodes have a weight, then the

A* algorithm can be applied to get the best path to the goal. Lastly, we smooth our path into a single line to obtain a heading and speed.

7 FAILURE: MODES, POINTS, AND RESOLUTIONS

7.1 Vehicle Failure Modes and Resolutions/Strategies

- Failure: Apollo II has a tendency to turn around in an attempt to go through the course backwards.
  Resolution/Strategy: We drop weighted GPS points at a set distance behind the robot; the GPS points make it appear to Apollo II that the route behind it is longer, making going through the course the desirable path.
- Failure: Glare can cause Apollo II's sensors to believe there are obstacles in front of it when there are none, trapping the robot.
  Resolution/Strategy: Anti-glare filters in both the hardware and software eliminate this problem.
- Failure: Apollo II can get stuck in a loop in an attempt to take the shortest route when running parallel to a goal and when multiple paths are available.
  Resolution/Strategy: Using a dynamic weight scheme to allow Apollo II to recognize when a path is not viable.

7.2 Vehicle Failure Points and Resolutions/Strategies

- Failure: Previous robots had an insufficient battery life.
  Resolution/Strategy: Apollo II is outfitted with a completely new low power control system and makes use of regenerative braking.
- Failure: Previously used E-Stops were not user friendly and at times caused problems for those unfamiliar with them.
  Resolution/Strategy: A highly reliable and intuitive ergonomic wireless controller with separable E-Stop was designed and is the current method of manual control for Apollo II.
- Failure: Any part that fails due to unforeseen circumstances can cause serious delays and failures at IGVC.
  Resolution/Strategy: There is now a replacement on hand for every part of Apollo II.
- Failure: In previous robots, maintenance was difficult and would often cause delays at IGVC.
  Resolution/Strategy: The new completely modular design of Apollo II allows for fast and easy maintenance of all of the robots parts.

8 SIMULATIONS

As mentioned previously in our innovations section, Apollo II has a new simulation program, displayed in Figure 7, that allows us to create our own courses and test how Apollo II's software reacts to different situations. The robotics team has created several different courses to test Apollo II's software on. This has proved to be invaluable to our software development because we don't have the means to create complicated courses on campus to test Apollo II on. It is also much easier and faster to create a virtual course than to create a real course. Our simulations were critical to the development of Apollo II during our vehicle's downtime while its systems were being redesigned. Without our simulations, we would not have met the deadline for the competition.

9 PERFORMANCE TESTING AND ASSESSMENT

Table 3: Testing and Assessment

Category	**Analysis Method**	**Predicted Performance**
Speed	Jazzy 1170 specs	5 MPH max
Ramp climbing ability	Verified empirically	30% slope
Reaction time	Limited by software cycle time	25 ms
Battery life	See below	2.75 to 5.5 hours (depending on usage)
Distance at which obstacles are detected	Hokuyo spec is 30 meters. Limited by software.	7 meters
Accuracy of arrival at navigation waypoints	Hemisphere specs and verified empirically	2 ft. 67% of the time

Apollo II's estimated average current draw for normal operation is between 10-20 amperes depending on usage. The batteries are rated at 55 amp-hours, therefore the estimated battery life is 2.75 to 5.5 hours, as calculated by the equation below. Run time is measured in hours, battery energy in amp-hours, and average current draw in amperes.

$$Run\ Time = \frac{Battery\ Energy}{Average\ Current\ Draw} \rightarrow \frac{55}{10} = 5.5 \text{ hours}$$

Our current assessment of Apollo II's performance is that our robot performs satisfactorily using the new control system and mechanical improvements. The autonomous function performs well using LMS only at this point, but by the time of competition we expect to have Apollo II fully functional in every aspect. This conclusion is based on field experience as well as success in software simulation.