ZERO^2

Embry-Riddle Aeronautical University
Team Members:
Marco Schoener, Nicholas Middlebrooks, Remy-Quinton Phillips, Yates Simpson
Faculty Advisors:
Charles Reinholtz, Patrick Currier, Eric Coyle

1 INTRODUCTION

Zero^2 (Zero Squared) is an innovative and improved successor to the vehicle named Zero that Embry-Riddle entered in the 2014 IGVC. As with the original Zero, Zero^2 (Fig. 1) is an autonomous, differentially-steered vehicle used for intelligent navigation. The sensor suite used on Zero^2 includes computer vision, LIDAR, a digital compass and a differential GPS. These sensors interface with an enhanced software system that now includes a health monitoring system. This system detects missing or erroneous sensor data and attempts to reset the sensor while adapting to the missing information. The circular (zero-like) shape allows the vehicle to make zero-radius turns in close proximity to objects without the danger of a sharp corner or edge catching the object. The elimination of corners is also an important safety feature that mitigates the possibility of injury in the unlikely event of collision with a human. The "Zero" shape is achieved by attaching arc-shaped side pods to the sides of the chassis to protect the exterior wheels from debris and obstacles. Zero^2's design incorporates novel mechanical, software, and electrical systems features, with an emphasis on simplicity and optimal utilization of sensors, power and computational resources. This report outlines the development of these systems and the methods used for system integration.

Figure 1: Model of Zero^2 with side pods attached.

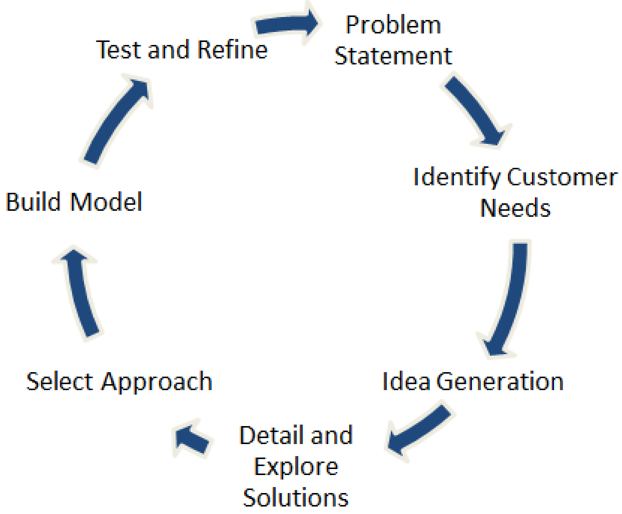

Figure 2: Design methodology.

2 DESIGN PROCESS

The development of Zero^2 used a seven-step design process that began with determining the problem presented by the competition. For the IGVC competition, the problem is to develop a robot that can successfully navigate through an obstacle course and a series of waypoints, while reacting to visual cues from painted lanes, potholes and colored flags. Primary customers include the IGVC competition judges, faculty advisors, and future team members. With those customers in mind, the next step was to develop specifications to address the competition requirements, and also to improve upon the previous year's platform. The customers' requirements, functional requirements, preferences, and comparisons with competitors were all used as part of the mechanical system design matrix shown in Table 1.

The primary considerations for Zero^2's design compared to Embry-Riddle's previous entry lie in the desire for improved structural stability and better electronic component accessibility. The matrix presented here highlights key features such as changes in the sensor mast, wheel placements, and wiring connections and access. The columns in the design matrix shown in boldface represent the design choices made by the team.

Table 1: Design Matrix for Zero^2 on a scale of 1 to 5 (most important).

	Circular Fiberglass Pole	**Square Fiberglass Pole**	**Sheath Pole Attach.**	Hinge Pole Attach.	Interior Wheels	**Exterior Wheels**	Wire Tubes/Channels	**Bridge Connectors**
Engineering requirements								
Competition size	5	5	4	N/A	4	4	N/A	N/A
Accessibility	4	5	5	3	5	5	1	4
Testability	4	3	3	3	3	5	2	4
LIDAR Visibility	5	5	5	N/A	N/A	N/A	N/A	N/A
GoPro Visibility	5	5	4	N/A	N/A	N/A	N/A	N/A
Aesthetics	5	4	4	4	4	4	4	3
Weight	4	4	3	4	4	4	N/A	N/A
Internal Wiring Organization	3	5	3	N/A	3	4	4	4
Waterproof	5	5	4	N/A	5	3	5	4
Cost	4	4	2	3	2	4	4	4
Score	44	**45**	**37**	21	30	**33**	20	**23**

2.1 Improvements

Table 2 shows a summary of major changes in this year's platform compared to Zero^2's predecessor; it focused on improvement through simplification and optimization.

Table 2: General improvements for Zero^2.

Mechanical	Software:
• Weatherproofing • Accessibility • Optimized payload placement • Smaller chassis size	• Adjusted algorithms and software to new robot design • More robust software functionality • Course settings added for specific navigation actions • Sensor health monitoring and adaptive control in the event of sensor failure

2.2 Innovations

2.2.1 Sensor Status Lights

The sensors on Zero^2 can now represent their current status (working, error, or off) through an Arduino-controlled LED system. This allows for an external detection of established, lost, or reset communication of a sensor that can be determined quickly.

2.2.2 Side-Chassis Pods

Zero^2 has a sleeker design uses exterior wheels to increase structural stability. The side-chassis pods is an aesthetic protection for the wheels to create the circular Zero planform shape. The pods protect the exterior wheel from debris and obstacles while increasing safety by eliminating sharp edges along its side.

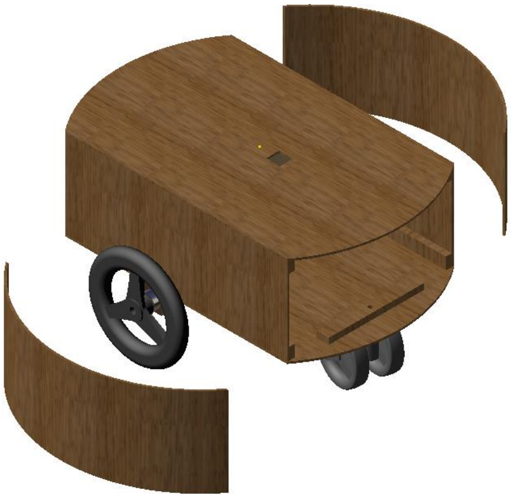

Figure 3: Side-Chassis pods shown on the side of the main chassis body.

2.3 Vehicle Cost

Most of the components on Zero^2's were adopted from its predecessor Zero. The new components, along with this year's cost, are denoted with an "*" next to the component name listed in Table 3.

Table 3: Cost of all mechanical and electrical components.

Zero^2 Component	Retail Cost	Team Cost
Sensors & Electrical		
DELL Latitude Laptop Computer	$780.00	$0.00
Go Pro HD Hero Camera	$400.00	$133.00
* Spartan GEDC-6E Digital Compass	$1,350.00	$0.00
TORC SafeStop ES 220 Wireless E-Stop System	$2,000.00	$0.00
LiPo 6 Cell Battery Packs	$60.00	$60.00
* Hemisphere A235 GPS	$2,500.00	$2,500.00
Keyspan Serial to USB	$88.00	$88.00
Hokuyo UTM-30LX-EW	$6,500.00	$5,250.00
Custom Power Board	$140.00	$140.00

Wires & Mics.	$200.00	$200.00
***** Arduino Uno32	$26.95	$0.00
* RGB High Power Output LEDs	$74.75	$74.75
Sensors & Electrical Subtotal:	**$14,018.00**	**$8,371.00**
Mechanical		
Quicksilver Motors	$2,200.00	$1,550.00
Caster Wheel	$15.00	$15.00
Skyway Wheels	$120.00	$60.00
* Wooden Frame	$200.00	$200.00
* Fiberglass Pole	$400.00	$400.00
Water Proof Cover	$50.00	$50.00
* Aluminum Channel	$40.24	$40.24
Mechanical Subtotal:	**$3,025.24**	**$2,315.24**
Total of New Components:	**$4,591.94**	**$3,214.99**
Total:	**$17,043.24**	**$10,686.24**

2.4 Team Composition

Table 4: Team Member list and each member's area of concentration for study.

Areas Of Concentration							
Team Member	Academic Major	Mechanical	Software	Electrical	Document	CAD	Hours
Marco Schoener (Mech/Soft Lead)	Mechanical Eng.	x	x	x	x	x	500
Matt Greene	Mechanical Eng.	x	x			x	300
Nicholas Middlebrook	Mechanical Eng.		x	x			500
Yates Simpson	Mechanical Eng.	x		x	x	x	450
Remy-Quinton Phillips	Civil Eng.		x		x	x	300
Brandon Reichert	Aerospace Eng.	x	x	x	x	x	300
Yakubu Pam	Mechanical Eng.	x	x				150
Eugene Denezza II	Mechanical Eng.	x			x	x	150

3 MECHANICAL

3.1 Vehicle Chassis

Zero^2's design focuses on light weight, high maneuverability, and easy maintenance. The chassis is fabricated from recyclable natural composite plywood to reduce weight, cost and to simplify manufacturing. The mechanical portion of the robot is composed of three primary assemblies: the drivetrain assembly, the box frame, and the sensor mast. Each subassembly is easily removed for individual maintenance while leaving the rest of the robot intact. In addition, the ability to disassemble Zero^2 aids in convenient transportation. Finally, Zero^2 has a circular frame with the minimum required dimensions for the competition, making it possible to execute zero radius turns without impacting obstacles and enabling the vehicle to easily fit through tight gaps, including doorways (Fig. 4).

Figure 4: Photo comparison of Zero (left) and Zero^2 (right).

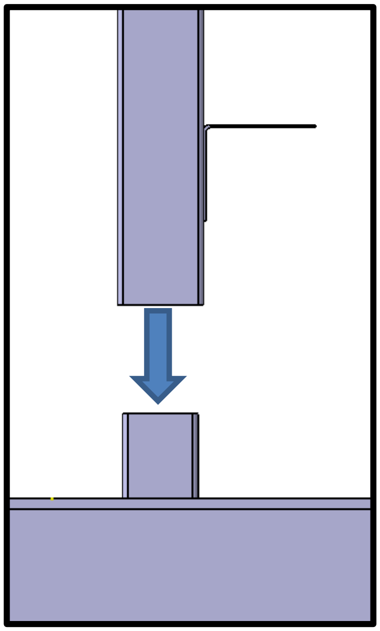

Figure 4: Side view of sensor mast where the top pole slips over the bottom pole.

3.1.1 Sensor Mast

The sensor mast is made of 1 ¾” and 2” square pultruded fiberglass channel mounted upright from the bottom base. Fiberglass was chosen to avoid electromagnetic interference and because it is a light and durable. The mast configuration includes a smaller 1 ¾” tube mounted inside the vehicle that protrudes through the top base by 2”. A slightly larger 2” tube slides over the top of the smaller tube and extends an additional 31” from the top base (Fig. 5). The mast holds the GoPro Hero camera, Hemisphere GPS, Hokuyo LIDAR, safety light, and stop button. The wires for the components are fed through the pole into the robot with connectors located at the point where the sections of mast mate together. This allows the top portion of the mast to be easily removed for transport or service.

3.1.2 Motor Assembly and Drivetrain

The motor assembly consists of a pair of 24 Volt Quicksilver motors and OEM NEMA 23 Series gearheads connected to two 12 ½” diameter Skyway tires. The aluminum clamps attached to the gearheads are fixed onto an aluminum channel support which is mounted to the robot base. Supporting the motors in this way provides mounting security and prevents warping in the robot frame. The assembly is quickly removable with four quarter inch bolts for maintenance and storage (Fig. 7).

3.1.3 Quick Service Tool Box

A tool box is placed inside Zero^2’s chassis towards the back. The tool box contains: safety glasses, a screwdriver, an adjustable, and cutters. This set of tools is sufficient to perform most servicing operations on the robot, but there is additional space available for future tool needs (Fig. 6).

Figure 6: Toolkit for Zero^2.

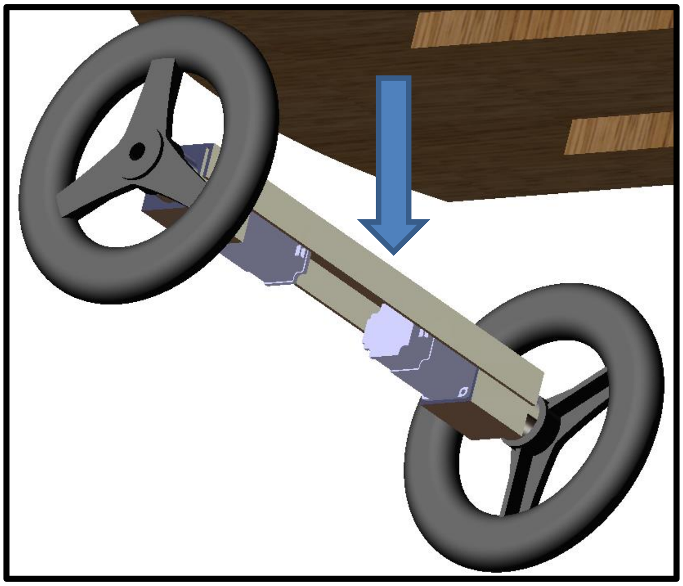

Figure 5: Drivetrain assembly below chassis.

3.2 Waterproofing and Durability

Because the vehicle must operate in light rain at competition, waterproofing is paramount. All wires to external sensors are routed through the sensor mast and into Zero^2 chassis. Wire penetrations are designed with water-resistant grommets or silicone sealant. Zero^2 also has a removable, water-resistant Sunbrella® fabric covering with a single penetration for the sensor mast. Although the top of the chassis is sealed around the sensor mast, a skirt around the base of the mast directs water away from the mast and off of the fabric cover.

4 ELECTRICAL AND SENSING SYSTEMS

4.1 Custom Power Distribution and Control Circuit

The central hub of Zero^2's power system is a custom developed power board. Unregulated 24V power flows from the batteries to the power board, which can provide regulated 24V, 12V, 5V, and 3.3V to the sensors (Fig. 8 and Table 5).

Figure 7: Zero^2 power distribution and motor control system.

Table 5: Operating Voltage Chart for all the sensors and electrical components in Volts (V) and Watts (W).

Sensor Voltage Chart				
Sensor Name	**Power Consumption**	**Voltage Range**	**Operating Voltage**	**Source**
Hemisphere A235 GPS	4.6W	7 – 36V	12V	Power Board
Sparton GDEC-6E Compass	0.32W	3.3V	3.3V	Laptop via USB
Hokuyo UTM-30LX-EW	~8W	10.8 – 13.2V	12V	Power Board
GoPro HERO	1.5W	3 – 5V	3.7V	Battery Pack
Quicksilver Motors	150W	12 – 48V	24V	Power Board
TORC Robotics SafeStop	8W	10 – 40V	12V	Power Board

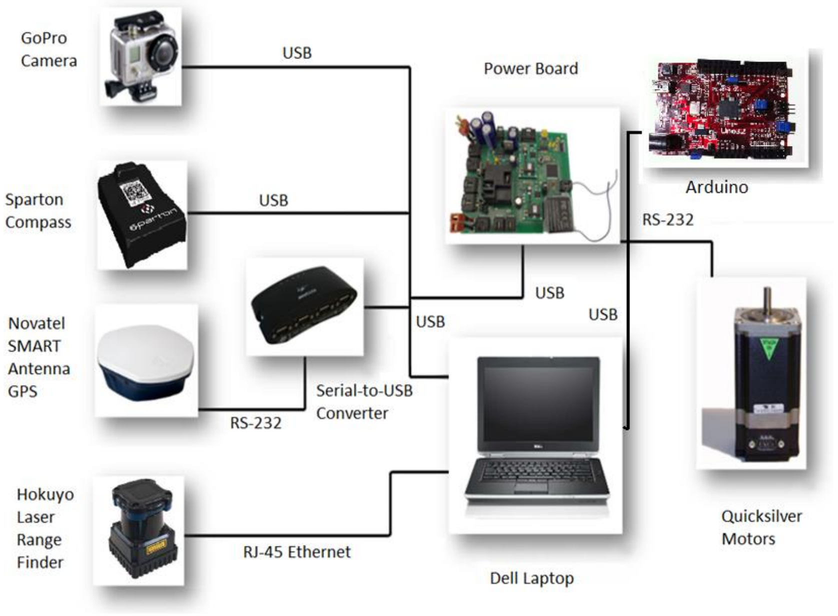

Figure 8: Electrical diagram for Zero^2.

The regulated 24 volts is distributed to each motor. The regulated 12 volts is sent to the GPS, SafeStop, LIDAR and LEDs. The regulated 3.3 volts is sent to the Sparton GEDC-6E compass. The regulated 5 volts is used for powering the Arduino. Each of these connectors has an individual fuse to avoid damage to the sensors. The electrical system (Fig. 9) is one of the more complex subsystems in a robot, and therefore has a high number of potential failure points. For this reason, the team spent substantial time working to design and document the electrical system of Zero^2 before implementing it in hardware. The team designed and manufactured a custom power distribution and control circuit board to provide all necessary operating voltages for each of Zero^2's components. Zero^2 can run between 1.5 to 2 hours on a set of batteries. An innovative feature retained from last year's vehicle is hot-swappable batteries. This feature eliminates the need to shut down and repower the robot for battery changes. This feature is most important during the dynamic events on Monday, where teams must be ready to run when they are called to the starting line. In addition, each voltage output port has an extra socket to allow for integration of future components. The board also provides remote control function from an R/C transmitter and both wired and wireless

Emergency Stop (E-Stop) capability. This all-in-one board is critical to the compact packaging layout in Zero^2.

4.2 Safety Systems

Zero^2 incorporates the SafeStop emergency stop system from TORC Robotics. E-Stop buttons are located on the sensor mast and controller. The controller (software) Emergency Stop has a range of 0.25 miles; when the robot is out of that range, the robot enters “safe mode” and is automatically stopped. As implemented, the SafeStop system provides a pause mode and a “hard” emergency stop mode. The pause mode rapidly brings the vehicle to a controlled stop without cutting power. The “hard” emergency stop opens a relay, disengaging all power to the power board. A light on Zero^2 indicates to bystanders when the system is under autonomous control.

4.3 Motor Interface

As Zero^2’s predecessor, Zero, the remote control solution and the command interface from the computer have been integrated into one microcontroller on Zero^2’s custom power and control board for safety and interoperability with other software packages being developed at Embry-Riddle. This board communicates with the motor controllers through an RS-232 serial line.

4.4 Sensor System and Integration

The central point of integration for all of Zero^2’s sensors is a DELL Latitude Laptop with a Core i5 2.50 GHz processor, 4 GB RAM, and 256 GB solid state hard drive. The LabVIEW programming environment is a critical tool used to receive and organize data from the sensors and run all software algorithms. Each sensor has a separate data acquisition block that is polled for current sensor readings. Zero^2 uses four commercial-off-the-shelf (COTS) sensors: a Hokuyo UTM-30LX-EW, a Hemisphere A235 GPS system, a Spartan GDEC-6E compass, and a GoPro HD Hero camera. An Arduino system takes the software status of the sensor and relays the outputs to a set of LEDs to determine state of sensors from the outside of the robot.

4.4.1 LIDAR

The Hokuyo UTM-30LX-EW laser range finder scans for obstacles in a 270° planar sweep in 1° increments at 20 Hz. The maximum sensing rage is 30 m, but Zero^2 limits detection to obstacles within 15 m. Resolution is 1 mm, and accuracy from 0.1-30m is ±50mm. Time-of-flight technology is used to calculate the distance to an object from the vehicle. This sensor scans in front of the vehicle and is used for obstacle detection and avoidance algorithms. The LIDAR collects angle and distance information of obstacles over the entire 270° plane and transmits this data to the laptop via Ethernet using TCP/IP protocols.

4.4.2 GPS

The Hemisphere A235 is a single unit GPS receiver and antenna that can gather GNSS and GLONASS L band signals and updates every 10 to 20 Hz. The uncorrected accuracy is typically between 1 to 2 m. However, the corrected accuracy with OmniStar HP brings the CEP down to around 0.1 m when combined with a Kalman filter. GPS data is transmitted to the laptop via RS-232 and a serial-to-USB converter.

4.4.3 Digital Compass

The Sparton GEDC-6E digital compass is a six-axis accelerometer and magnetometer that provides heading, pitch, and roll information with 1° RMS accuracy at 0.1° resolution. Zero^2 accesses the orientation data at 20 Hz via RS-232 and a serial-to-USB converter.

4.4.4 Digital Camera

The GoPro HD Hero is an outdoor sport, consumer grade 5 megapixel digital camera with a very wide 170° field of view lens. The GoPro on Zero^2 is configured to output 720x480 standard definition video. This video is streamed to the computer with a digitizer and captured at 20Hz. The GoPro camera runs off its own battery power with a typical use time of one hour continuous streaming.

5 SOFTWARE STRATEGY

5.1 Structure

Zero^2's software system was developed using National Instruments LabVIEW. LabVIEW was chosen because it provides an intuitive Graphical User Interface (GUI) which allows the user to easily monitor, modify, and debug software and easily handles sensor integration. The GUI is helpful in verifying that all of Zero^2's sensors and components are fully operational before the autonomous program is run.

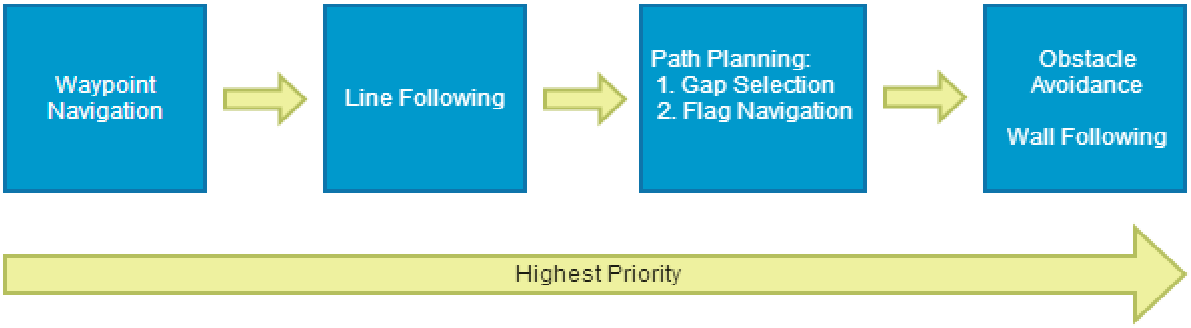

Figure 9: Flowchart of Zero^2's Software Architecture.

The code is organized into four major sequential steps, with each later process able to make use of and subsume the previous decision. The steps are waypoint navigation, line following, path planning, and obstacle avoidance (Fig. 10).

5.1.1 Software Improvements

The software has been improved to simplify making changes in parameters such as sensor placements and vehicle dimensions. While these parameters are not likely to change during competition, such flexibility aids in vehicle development and testing. The software is designed to execute functions in order of precedence (Fig. 10). An important innovation this year is the ability of the software to reconnect sensors that are not updating or that are generating erroneous data. If a sensor is not producing valid data, the software uses estimation theory to generate "dummy" variables. This allows the robot to continue while sensors are attempting to restart or in the event of a total failure of one sensor.

5.2 Waypoint Navigation

The first part of Zero^2's software structure consists of Waypoint Navigation. The GPS and compass provide Zero^2's current position and heading, respectively. With this information the angular error and the distance to the target waypoint can be calculated. Without the presence of any obstacles, independent PID control loops are used to control Zero^2's angular velocity and speed based on the angular error and distance to waypoint feedback as depicted in Fig. 11.

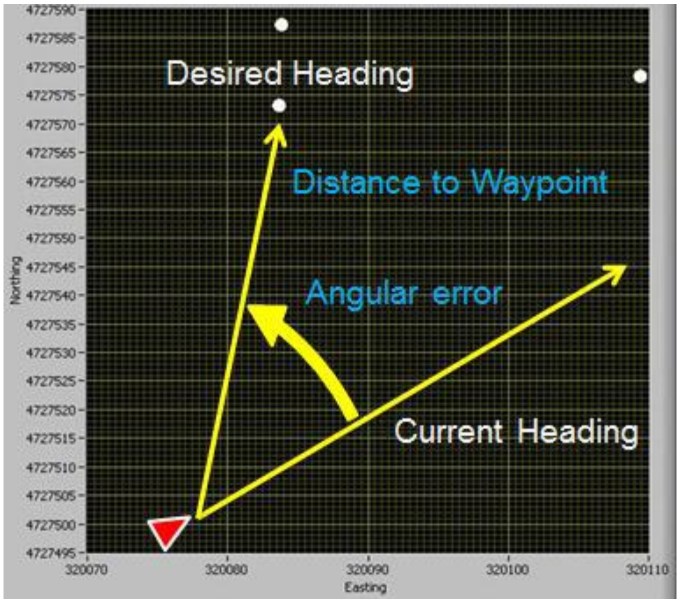

Figure 10: Waypoint Navigation graph of the robot's heading vs. the desired heading.

5.3 Line Following

Once the direction to the waypoint is determined, the next section of code implements line following. The line following flow diagram, shown in Fig. 12, illustrates the primary steps in the line extraction algorithm. First, box covers are placed at the top and bottom of the image to block out the horizon and vehicle, respectively, since both can have very bright pixels that can saturate the image and are not lines. Next, the image is down sampled down to 720 x 480 to blur some noise and reduce processing time. A 4:1 plane threshold of blue and green filters is performed to obtain a grayscale image. The image is also split into a left and right half, since there are potentially two dominant lines in the image.

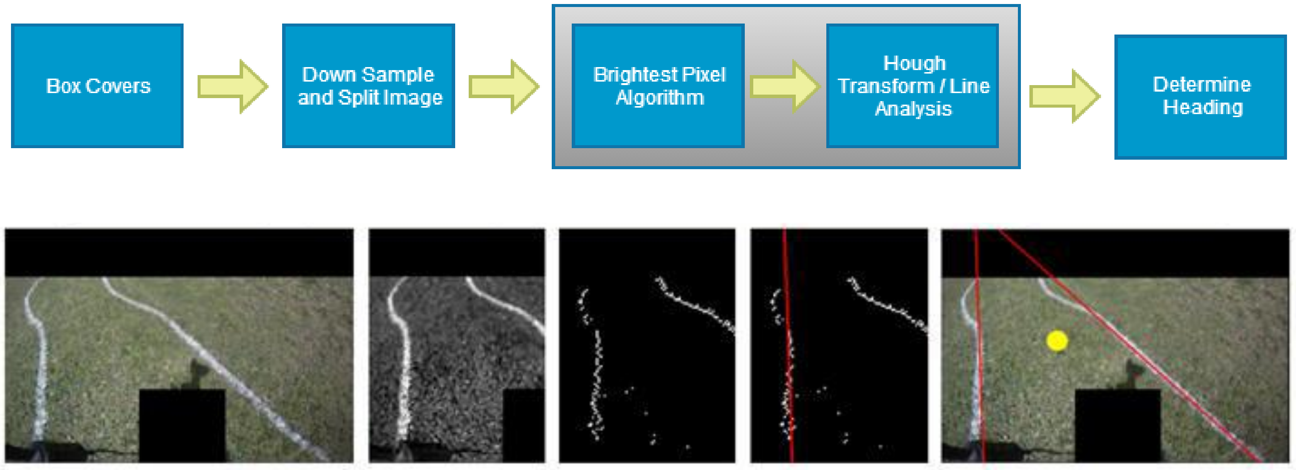

Figure 11: Line detection algorithm.

A brightest pixel algorithm isolates the white pixels by scanning both horizontal and vertical lines for the pixel(s) of highest value. Then, a Hough transform uses a voting system to determine the slope and distance to the dominant line traced by the pixels of each half-image. It is possible that no line is detected in the image if no candidate receives a minimum number of "votes" in order to be considered a line. If lines do exist, they are categorized as horizontal or vertical, and compared with each other as parallel or intersecting. The last step is to recombine the half-images and use a decision tree to select the heading given the possible combinations of lines in half-images. For example, if both images detect a line, the heading should be between them. If only one image contains a line, then the heading should be a few feet left or right of this line as appropriate to stay within the course.

5.4 Path Planning

Path planning uses the previous data from the waypoint navigation and line heading and adds one of two scenarios: either Gap Selection or Flag Navigation.

5.4.1 Gap Selection

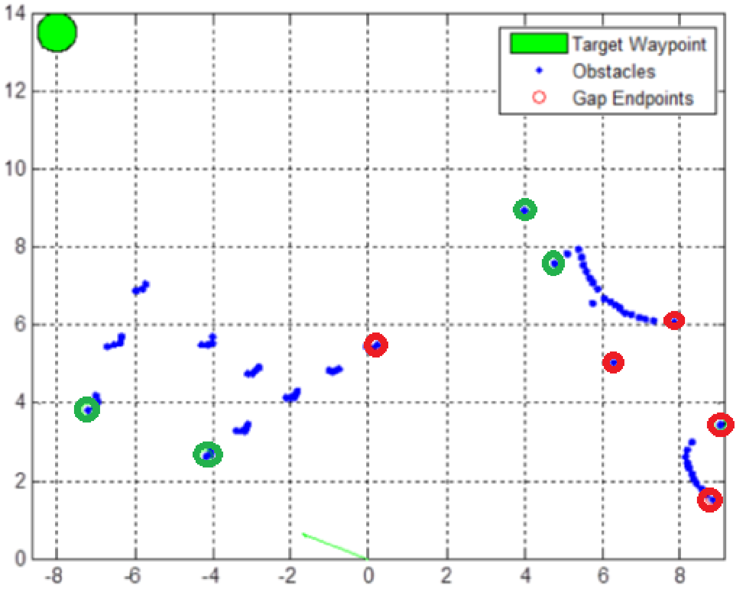

Figure 12: This is a screenshot of the obstacles (in blue) that the LIDAR sees and shows the openings it sees (in red circles).

Zero^2 makes use of a long range optimal heading algorithm for gap identification and vehicle maneuvering. Although Zero^2's LIDAR system sees a 180° FOV of objects at up to 30 meters away, the obstacle avoidance algorithm only makes use of data points within a set 2 meter distance threshold of the vehicle. This results in somewhat clumsy paths that can be characterized as simply straight lines towards the next waypoint until an object is within 2 meters, at which point the vehicle will make a sudden left or right turn. The path planning algorithm eliminates this sub-optimal behavior by making use of data within a range of 15 meters. The algorithm analyzes the obstacle data and segments objects so that any gap greater than the vehicle's tolerance width for passing through, about 1.5 meters, is marked as either a left-handed or right-handed opening. Fig. 13, is a screenshot of a MATLAB simulation of what the code performs. Zero^2 is in the center at the bottom (0, 0) with a green arrow. There are small green circles represent left side openings and red circles mark the right side openings. The green arrow shows the heading that the algorithm has determined leads to the optimal opening. With this algorithm, Zero^2 can drive straight to the optimal opening instead of simply driving straight until it is in close proximity to an obstacle. Given the limitations of the LIDAR, namely that it cannot see through objects, this technique provides improved behavior going towards unknown parts of the course.

5.4.2 Flag Detection

For the advanced course in the competition, blue and red flags are arranged in a complex row arrangement. The flag detection algorithm uses three simple steps (Fig. 14). First, it retains the same box covers as the line detection algorithm to block out parts of the image that are near the horizon or vehicle. Then it performs a mixed-plane threshold based on hue (color), RGB ratios, and HSL values to determine pixels that qualify as either blue or red. Finally, a particle filter is used to eliminate blobs that are too small or too large to possibly be flags. The results are overlaid on the GUI so that the user can immediately see what has been detected as a flag and make adjustments to code as needed. This year, the field of view for detecting the flags has been reduce to lower the probability of detecting objects off the field.

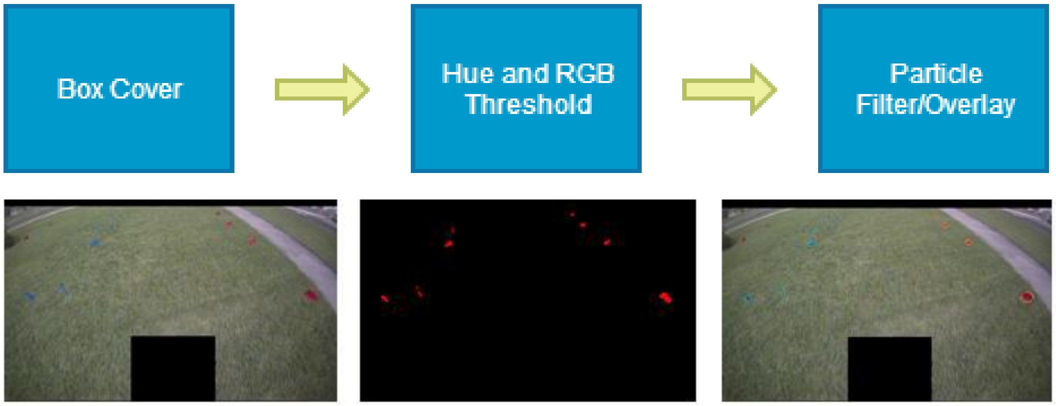

Figure 13: Flag detection algorithm.

5.5 Obstacle Avoidance

Zero^{2}'s obstacle avoidance algorithm implements control over motor commands when the vehicle is within 2 meters of an obstacle, as measured by the LIDAR. The LIDAR's 180° field of view is broken into five zones: center, middle left/right and far left/right. Fig. 8 below shows the vehicle with the zones defined. The segmentation of these zones can be modified by the user but are currently set at: 0° (due right), 30°, 65°, 115°, 150°, and 180° (due left). A zone is considered occupied when an obstacle is within 2 meters. An occupied zone indicates to Zero the instruction to turn in the opposite direction. While each zone is labeled as occupied or unoccupied, the algorithm continuously uses a decision tree to decide the path to avoid obstacles. The main check on the decision tree is to check if the center cone is occupied. If that cone is occupied, it goes on to check if the previous command was left or right. Next, the middle left or right, respectively, is checked to see if it is occupied. This decision tree continues on for all possible combinations of cones and objects (Fig. 15).

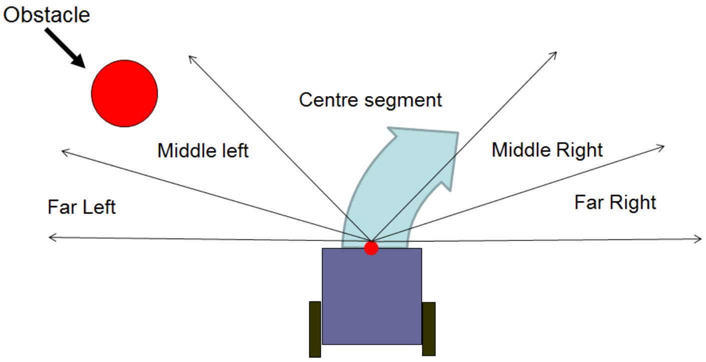

Figure 14: A diagram that shows the different obstacle detection zones that the LIDAR sees and controls obstacle avoidance.

5.6 Wall Following

The wall following algorithm activates if the following two criteria are met: line detection has been turned off by achieving the first waypoint leading into “No Man’s Land” and the first three waypoints in the middle field have been achieved. The wall following algorithm uses the LIDAR, through the Gap Selection algorithm, to detect obstacles and openings in the wall. Zero^2 scans the wall and follows it to its opposite end while maintaining a distance between 1.2m to 1.6m. If an obstacle, or the wall, is less than 1.2m, it will move away from the wall and continue to follow the length until it finds the opening. Once Zero^2 moves through the opening, it will continue through to the waypoint out in the field and change back to obstacle avoidance mode.

5.7 Complex Obstacles

There are two designated complex obstacles the system will experience at competition, switchbacks and dead ends. Zero^2’s approach to each is briefly discussed below.

5.7.1 Switchbacks

A switchback occurs when the field requires the vehicle to successfully navigate through a zigzag like obstacle course. Zero^2 handles switchbacks using the waypoint navigation and the five zone obstacle avoidance. The waypoint navigation knows the waypoint’s location and uses the angular error to calculate the desired heading. Zero^2 effectively avoids obstacles by using the five zone obstacle avoidance; the vehicle will recognize the minimum gap of five feet and be able to make it through without hitting any of the obstacles. Fig. 15 depicts Zero^2’s desired path for a switchback situation.

5.7.2 Dead Ends

A dead end is a complex situation that requires Zero^2 to use more than just the waypoint navigation and the five zone obstacle avoidance. With only these, Zero^2 would continuously circle in the dead end. The dead end algorithm is broken down into three sections: recognizing the dead end, the action to take in order to get out of the dead end, and turning off the algorithm. Zero^2 recognizes the dead end by constantly checking to see if there is an obstacle in the three most forward zones for more than fifty percent of the time. This, in conjunction with any turn greater than 120°, will activate the dead end algorithm. The plan for getting out of the dead end is to identify the obstacles by using the gap planning algorithm to determine the desired direction and optimal path to navigate out of the dead end. The code will turn off when Zero^2 moves a certain distance away from the dead end obstacles.

5.8 MATLAB Data Log Simulator

The team has also developed a data logging system which outputs all sensor and algorithm information into a text file that can be imported into MATLAB and replayed. The data logging system helps immensely with the test and refine process by identifying problems that cannot be immediately noticed by vehicle performance inspection during a test. The output of the program is shown in Fig. 16. The black rectangle represents the robot, while the blue represents obstacles, and the green circle is the target waypoint. The green semi-circle extending from the vehicle is the obstacle avoidance range, which will cause a reaction from the robot. The red dots show the vehicle’s GPS trail. On the left hand side are numerical values that can be customized to whatever the user wishes to see, including elapsed time, wheel speeds, and latency.

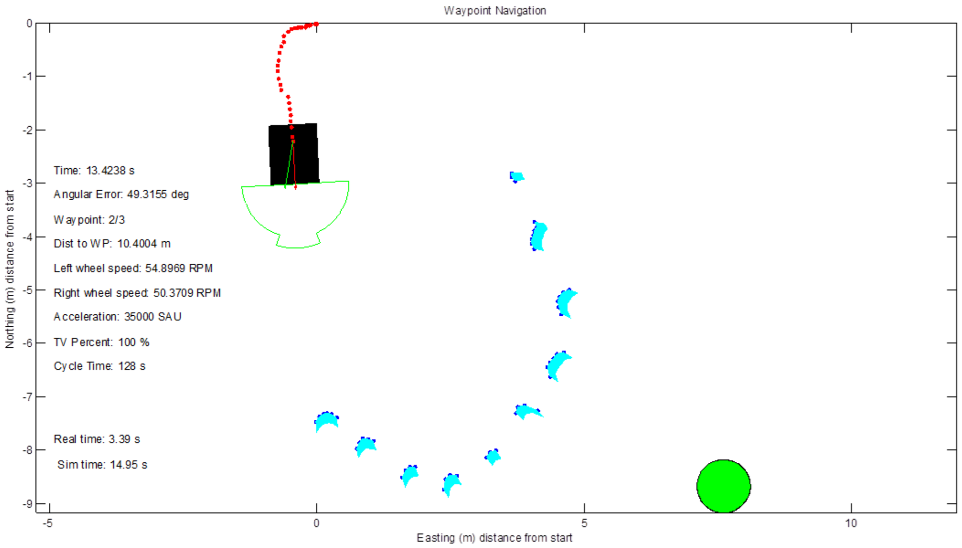

Figure 15: The screenshot depicts the robot (black) driving towards the waypoint (green), but detects obstacles (blue) in its path.

6 COMMUNICATIONS

6.1 JAUS Protocol

The Joint Architecture for Unmanned Systems (JAUS) is an SAE standardized communication protocol that has been implemented on Zero^2. The Interoperability Profiles (IOPs) integrates JAUS's services to help it perform the four main IOPs: the Overarching IOP, the Communication IOP, the Payloads IOP, and the Controls IOP. The attributes that spawn from these categories are the Platform Databus Attribute, the Transport Attribute, and the Mobility Attribute. This software requires a sequence structure, which creates a timeline of events. Zero^2 incorporates the Core Operability attributes and Platform Management Attributes and the Navigation and Reporting Attribute group. The first event opens the port and UDP connection to the controlling unit by connecting to the Judge Testing Client (JTC) using a special team Subsystem ID (SSID). Zero^2 then broadcasts a Query Identification every 5 seconds. Once the control unit responds, the next sequence is started.

The second event parses, sends and receives JAUS messages. Zero^2 receives messages faster than it can process the messages. Even so, all of the messages are processed in the order of reception and placed into an event queue. Once the message is removed from the queue, the first action required is to determine the validity of the messages by checking the origination identity, as well as the sequence number to ensure messages being received only once. Once a message is determined to be valid, the message identity is determined and the remaining message data is handled appropriately. Responses are placed into another event queue, sequenced into a header and trailer, and sent to the control unit.

6.2 Latency (Reaction Times)

Zero^2's software code is able to run at about 9 Hz on an Dell Laptop with an Intel i5 2.30 GHz dual core processor and 4 GB RAM running on Windows 7 (x64). The vision algorithms take about two-thirds of this processing time. Similarly to Zero, Zero^2 can sense at 20 Hz or faster depending on the sensor, so the limiting factor is the speed of Zero^2's main algorithm process.

7 CONCLUSION

Zero^2 is an improved successor to the successful Zero platform. The new Zero^2 product implements an effective, affordable, safe and maintainable integrated systems design that meets all of the requirements and challenges of the 2015 Intelligent Ground Vehicle Competition. Improvements to Zero^2 included new innovative features such as robust operation in the event of sensor failure, sensor status LEDs and side-chassis pods. Zero^2 is optimized for the 2015 Intelligent Ground Vehicle Competition, but the software has been developed facilitate improvements for other uses and for future competitions.

8 REFERENCES

[1] IGVC Rules committee, "IGVC Rules 2014," http://www.igvc.org/rules.htm (accessed April 15, 2014).

[2] Bacha, Andrew R. "Line Detection and Lane Following for an Autonomous Mobile Robot." Thesis. Virginia Polytechnic Institute, 2005. Print.

[3] Ulrich, I. and Nourbakhsh, I., "Appearance-Based Obstacle Detection with Monocular Color Vision," *Proceedings of AAAI 2000*, 2000.

[4] Vadakkeptat, P., Kay, Chen Tan., Wang, Ming-Liang., “Evolutionary Artificial Potential Fields and Their Applications in Real Time Robot Path Planning”, *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on,* On page(s): 256-263 Volume: 1, 2000.

[5] Hurst, A., Greene, M., Schoener, M., and others, “Embry-Riddle Aeronautical University – ALVIN,” http://www.igvc.org/design/2013/Embry-Riddle%20Aeronautical%20University%20-%20Alvin.pdf. *Annual Report.* Embry-Riddle Aeronautical University, 2013. Web.

[6] Schoener, M., Phillips, P., and others, “Embry-Riddle Aeronautical University – Zero,” http://www.igvc.org/design/2014/5.pdf. *Annual Report.* Embry-Riddle Aeronautical University, 2014. Web.

EMBRY-RIDDLE
Aeronautical University
DAYTONA BEACH, FLORIDA

Department of
Mechanical Engineering

Tel: 386-226-6667
Fax: 386-226-6011

May 15, 2015

Intelligent Ground Vehicle Competition
AUVSI Foundation

Dear IGVC Judges,

I certify that the engineering design of Zero^2, as described in the accompanying report, has been significant and is equivalent to that required of a senior design project.

Sincerely,

Eric Joe Coyle
Assistant Professor of Mechanical Engineering
Embry-Riddle Aeronautical University

600 S. Clyde Morris Blvd.
Daytona Beach, FL 32114-3900

embryriddle.edu