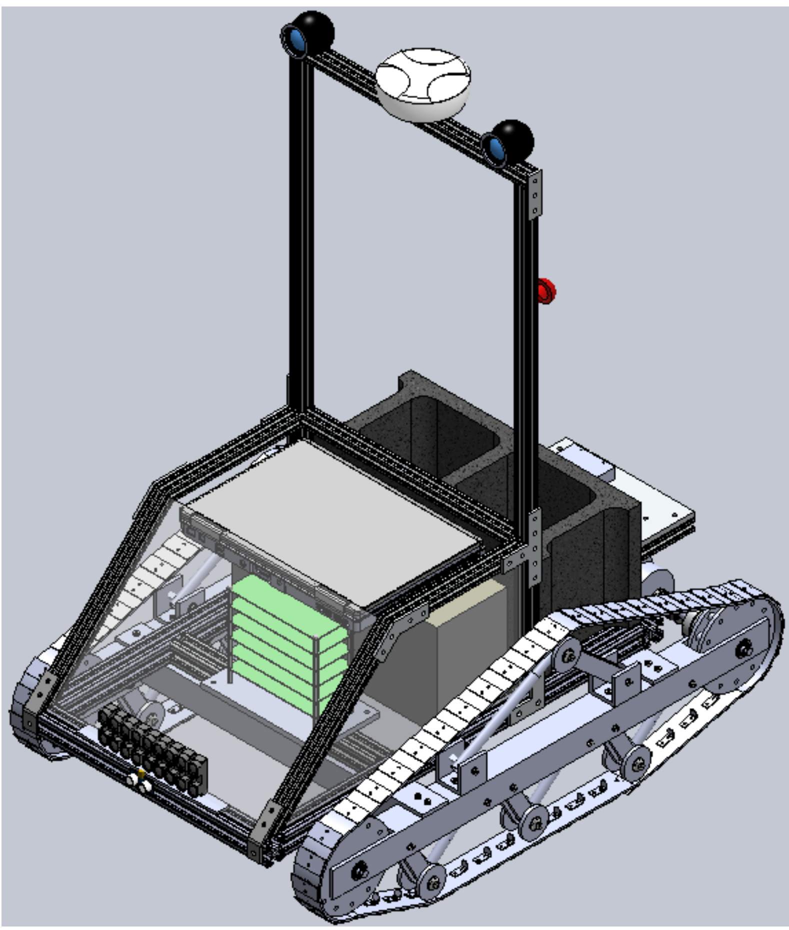
YORK COLLEGE
of Pennsylvania

YORK COLLEGE
GREEN LIGHTNING

Intelligent Ground Vehicle Competition
IGVC



York College of Pennsylvania Design Report 2010

Team Overview

Introduction

York College of Pennsylvania (YCP) has entered the intelligent ground vehicle competition (IGVC) for the first time with the intent of providing the senior class with a challenging opportunity to engage in cross discipline collaboration. The team is comprised of students from the class of 2010 which spans multiple physical science disciplines. The vehicle being entered in the competition is named *Green Lightning* and has been fully designing, fabricated, and assembled by the team members. The YCP vehicle has been constructed to fully comply with all the regulations established by the 2010 IGVC rules and has been equipped with a multitude of design innovations that will allow *Green Lightning* to succeed at this year's 2010 competition.

Team Structure

Table 1 shows the team overview and displays the discipline dispersion. Among the four majors represented on the project: Computer Engineering (CE), Computer Science (CS), Electrical Engineering (EE), and Mechanical Engineering (ME).

Table 1: Team Structure Technical Responsibilities

IGVC Team 2010				
Name	**Major**	**Class of**	**Technical Area**	**Est. Hrs.**
Matthew Adams	EE	2010	Wireless Communication	350
Jennifer Britnell	CE	2010	Short Range Detection	400
Jeremy Deschenes	CS	2010	Navigation Integration	200
Van Hare	EE	2010	Vision System	550
Steve Kirpatrick	CS	2010	Vision System Integration	100
Chris Matthews	CE	2010	Embedded Systems Controller	950
Jessica Matthews	CS	2010	Navigation	100
Anh Nguyen	EE	2010	Long Range Detection	600
Anwar Ross	ME	2010	Track System	800
Ben Siedel	ME	2010	Frame	480
Josh Stuart	ME	2010	Drive Train	550
			Total Hours	5160

Each of the aforementioned technical areas will be described in greater detail in the following sections of the design report.

Structure

<u>Design</u>

The goal of the frame design was to create a well-built and rigid structure that could support the vehicle's weight while housing all of the components required for a successful vehicle. To accomplish this task, the team decided to make the structural frame from modular aluminum components from 8020® Inc. Using this material allows for a lightweight, durable, and rigid structure that can also be easily reconfigured. The 8020 aluminum is easy to machine and has great aesthetic appeal. Once the material was chosen, an overall vehicle size was decided on. Design work began using SolidWorks 2009 to assemble the frame and integrated enclosure. We wanted to create a more interesting looking vehicle that was not simply a box with wheels. In order to develop a more streamlined vehicle, the decision was made to use custom joining plates to link the 8020 pieces. To further the aesthetic appeal of the vehicle, the frame design incorporates Plexiglas plates that can be placed within the channels on the 8020 to create an environmentally sealed and physically protective enclosure for the electric components.

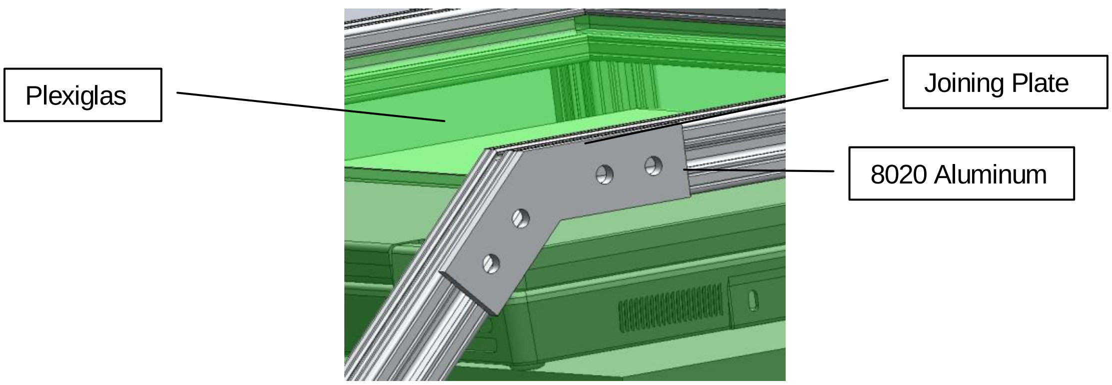


Figure 1: Structure Close-up

Figure 1 provides a detailed view of a joining plate made. It also shows how the Plexiglas fits within the channels of the extrusion.

<u>Fabrication</u>

To fabricate the frame, the first step was to make manufacturing drawings from the 3D modeled parts. An aluminum plate was purchased and sent to an outside source to have the joining plates water jetted. The 8020 was cut and machined using band saws and mills. Using full size drawings, the Plexiglas pieces were cut and sanded to create a snug fit into the T-channel of the aluminum extrusion.

<u>Integration</u>

Significant communication and collaboration regarding system design was conducted to ensure effective and efficient vehicle integration. By working closely with team members and sharing 3D model files, the team was able to create a single vehicle assembly that implements all the components in a well-organized and sleek design.

Track System

Design

The IGVC takes place on a course that presents a wide variety of terrain for vehicles to traverse. While navigating the various terrain types, vehicles must also maneuver around a collection of different obstacles. Consideration of both the maneuvering and traction elements of the competition has provided adequate reasoning to pursue a tracked vehicle drive system for *Green Lightning*.

A test bench was designed to assess track thrust generation on a variety of surface conditions, which included loose soil, grass, and gravel. Based on the outcome of the data provided, a dual climbing track configuration with McPherson "like" suspension and a traction-pad retrofitted belt was selected.

Fabrication

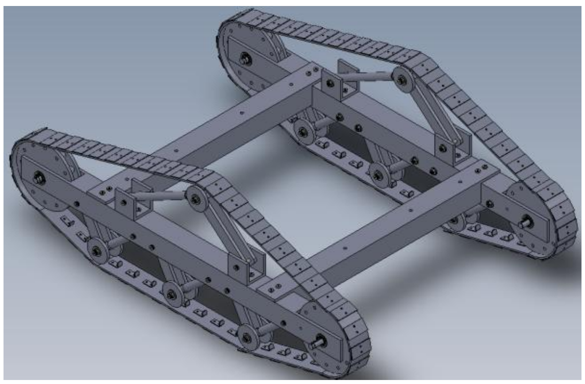


Figure 2 – Overall Chassis

The entire propulsion system includes: a left track assembly, a right track assembly, and two cross members.

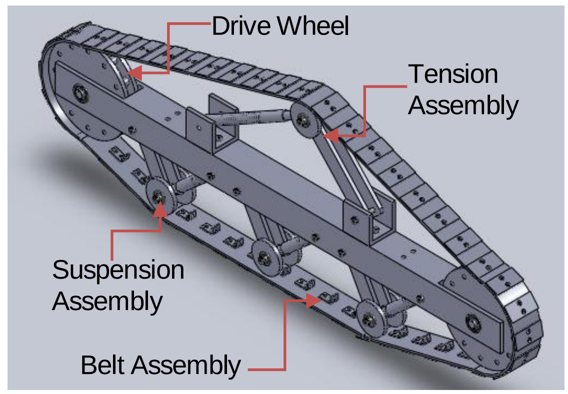


Figure 3 – Left Track Assembly

The left track assembly is shown in greater detail above. The right track assembly is a mirror image of the left assembly.

As can be seen in Figure 3, the track assembly is composed of four main components: a belt assembly, dive/idler wheel assemblies, a mounting beam, and tension/suspension assemblies. The belt was chosen to be flexible and to accommodate the addition of traction pads for increased terrain handling effectiveness. The drive and idler wheel assemblies were designed to easily interface with the belt as a low cost, high efficiency solution. The mounting beam was designed to be structurally sound as a rigid support member. The suspension and tensioning assemblies were designed to mitigate the effect of low frequency terrain perturbations and decrease sensor disturbances.

Implementation

The traction system is designed as a standalone system so that the other systems can be modularly attached. The main vehicle frame can be modified and changed independent of the chassis and attached via simple mounting points located on the cross members. The drive train can also be reconfigured and vary independent of the track system with little to no impact on the design of the chassis again with the implementation of a basic mounting point design platform.

Drive Train

Design

In order to drive the tracks, a drive train and motion control system will be needed. The motor drive train consists of motors, gearboxes, encoders, and couplings to link the gear boxes to the drive shafts, while the motion control system consists of a motor driver and controller. Based on a variable acceleration analysis of the vehicle's stopping distance vs. power and team discussions, a 4 ft stopping distance, from 5 mph, was chosen. This reduces the maximum power requirement, while maintaining a short stopping distance.

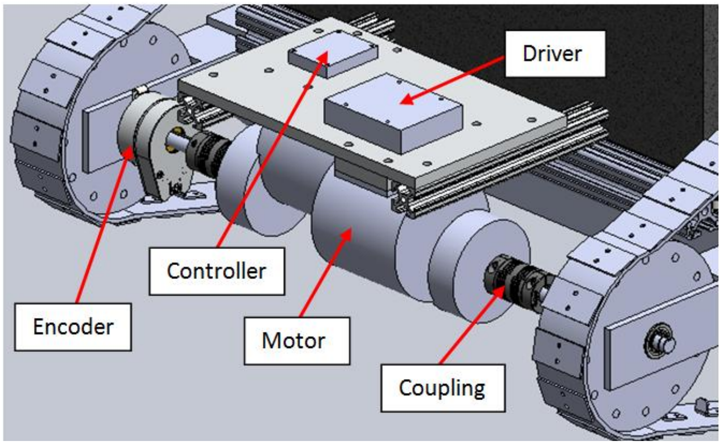


Figure 4: Drive Train Assembly

The radius of the drive wheel on the track system is an important variable to determine the speed and torque requirements. Standard DC motors typically have high operating speeds and provide low torque, thus the wheel radius has been limited to 3 in to reduce the amount of gearing needed. The power required by the vehicle propulsion was calculated to be 230 W (min.) while the torque was calculated to be 70 in*lbf. The peak torque of the system was found to be 110 in*lbf which occurs 2.5 mph.

Fabrication

The MY1016Z3 brushed DC motor from Monster Scooter Parts was selected as the best option for the drive train. The gearing ratio needed to reduce the motor's speed and increase the torque can be determined by using the specifications for the previously mentioned motor. The speed the drive shaft needs to rotate to maintain the maximum velocity can be calculated based on the wheel radius of the drive wheel. This can then be used with the output speed of the motor to determine the appropriate gear ratio (9:1). These calculations confirm the MY1016Z3 (ratio of 9.778:1) as a valid option that meets all the requirements. This motor, with attached gearbox, is also light and compact making the impact on the overall vehicle weight and size minimal.

Implementation

As shown in Figure 4, the fully assembled drive train consists of the motor and gearbox directly connected to the drive shaft of each track assembly. The motors are mounted on the rear of the chassis via an overhung beam and mounting plate. This plate was designed to be a multipurpose platform doubling as a mounting position for two components of the Motion Control System (MCS). A custom Motor Controller was designed to relay data from the Embedded Systems Controller. The Nubotics WheelCommander directly controls the motors via encoder feedback and control loops. Finally the Sabertooth Motor Driver applies the appropriate voltage based off of a Pulse Width Modulated (PWM) signal from the WheelCommander.

Embedded System Controller

Design

The Embedded Systems Controller (ESC) is the data and controls hub of the entire vehicle. It was decided early on that the vehicle's controls should not be placed entirely on a general purpose computer. If the Operating System were to freeze or even hang for just a few moments it could create potentially hazardous effects. Thus in order to safeguard vehicle functions and improve safety, a dedicated systems controller was implemented.

Fabrication/Implementation

The ESC was designed with two primary functions in mind, the collecting and relaying of vehicle data to the computer and to monitor vehicle vitality. To accommodate the first function, protocols were developed using USB for communications between the PC and the ESC and the Serial Peripheral Interface (SPI) capabilities of the ESC to communicate with the rest of the sub-systems on the vehicle. This SPI Network (SPIN) and its underlying protocol was evaluated and tested using a variant of Rate Monotonic Scheduling Analysis (RMA) to verify that in every possible instance of execution, the protocol would not cause lapses in communication. Figure 5, shows an example of the RMA analysis performed on the SPIN firmware being used by the sub-controllers of the system. The figure shows the critical moments (solid vertical lines) where critical tasks must be finished or else a failure occurs. In this example, “Slave Sync” and “Slave Reset” are the critical tasks that must complete before their respective critical moments. For this particular analysis, the “Slave ISR” and “Slave cDat()” represent possible sources of delay for the aforementioned tasks. SPIN was evaluated with this procedure as referenced above and system requirements were established. Since SPIN is custom designed, it also very flexible and should slaves not be able to meet established criteria, SPIN can be adjusted for that particular device. If during operation SPIN were to fail, safeguards have been put in place so that the ESC would take notice of these failures. This highlights the second purpose of the ESC, which is to monitor and report vehicle status as a whole. Should a severe error be detected, the ESC will request an E-Stop from the E-Stop Controller. Due to the ESC’s location within the Electronics Hierarchy (Figure 6) it has the capability to constantly monitor and report vehicle safety and operation, thus fulfilling its second objective.

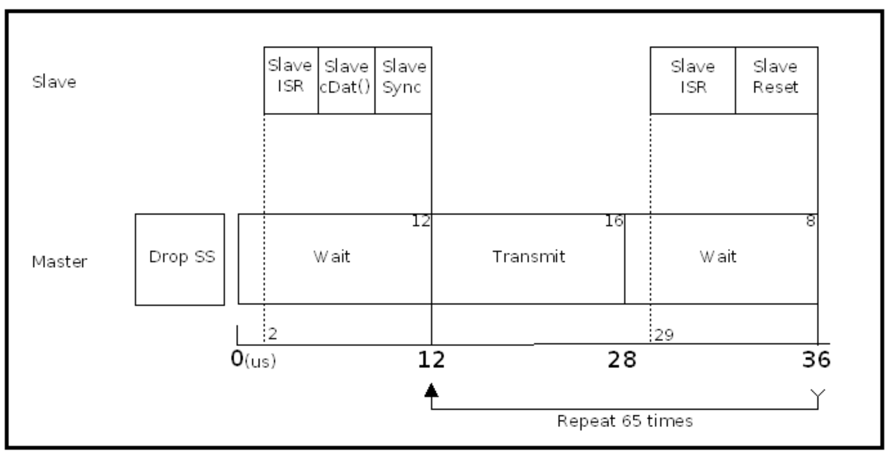
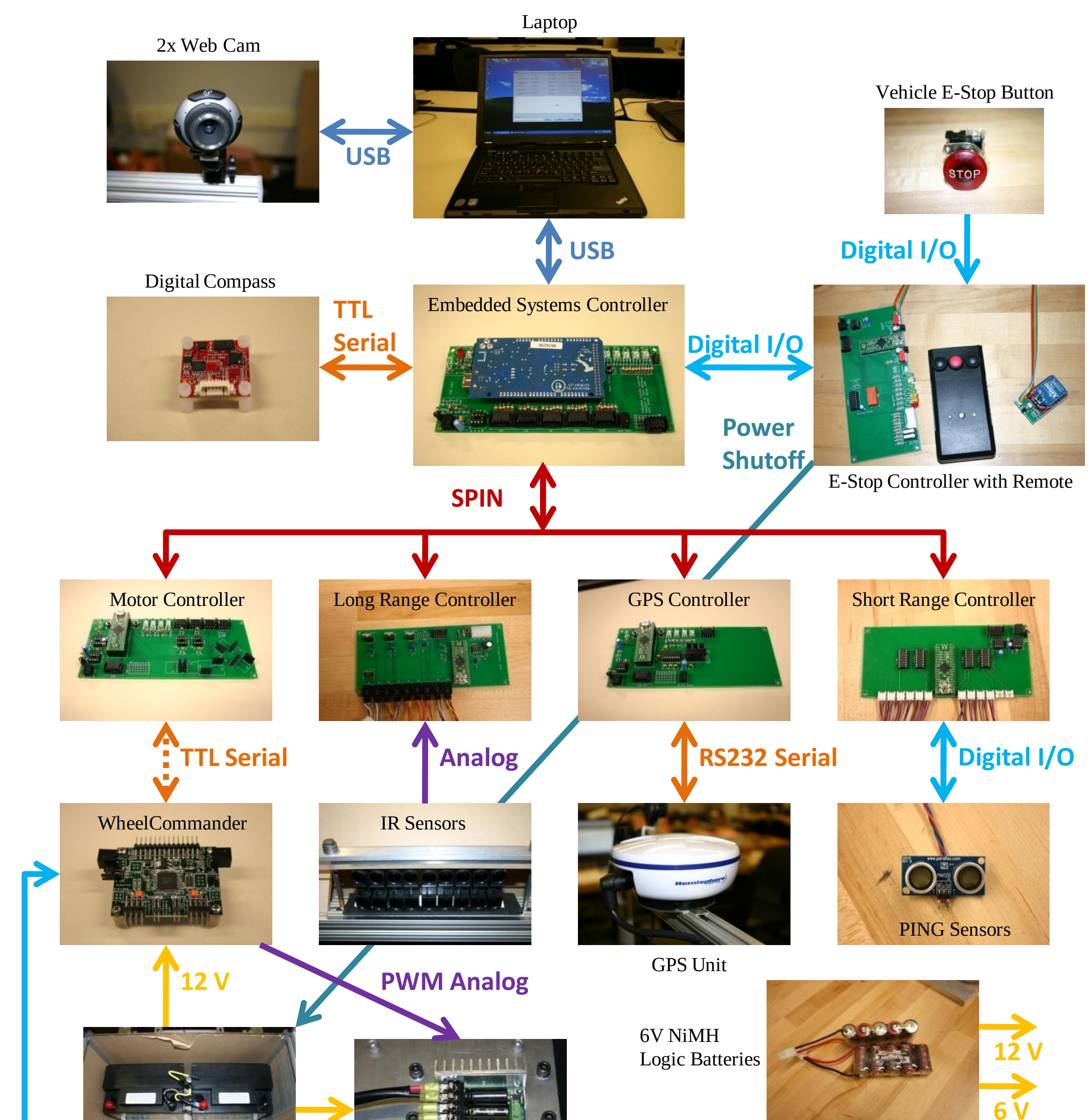
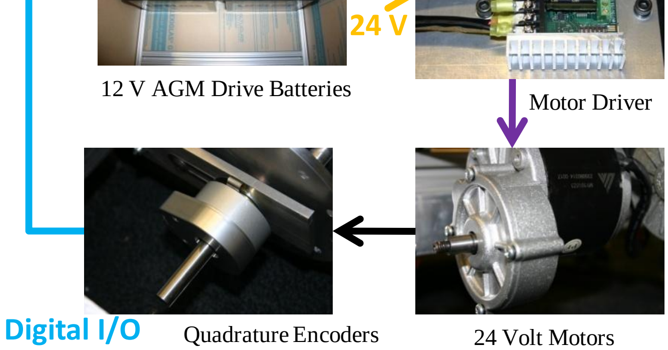


Figure 5: Timing Analysis Graphical Example

Vehicle Electronics Overview





Power System Summary:
Two power sources exist in the vehicle, one set of batteries for the motors and one for the electronics. Each system is electrically isolated, and the TTL Serial link between the WheelCommander and Motor Controller is achieved using optical isolators. All outgoing power leads pass through appropriate fuses rated for each system in which it feeds. The "Logic Batteries" supply power to all embedded controllers via a 6V input. The only device requiring 12V is the GPS unit, which is controlled using the GPS Controller and an onboard power driver. All subsystem input power is internally regulated with respect to each controller.

Figure 6: Electronics Overview

GPS

GPS Subsystem

The team selected the Hemisphere A100 Smart Antenna GPS unit, shown in Figure 7. This unit is capable of determining our vehicles location with 0.6 meters accuracy at 95% confidence. Positional accuracy is augmented using SBAS differential corrections. In order to lighten the load on the ESC, a sub-processor is used in parallel with the GPS controller to collect data from the GPS unit and report to the ESC when requested. This allows the ESC to service other tasks as the sub-processor collects data from the GPS unit.

Figure 7: Hemisphere A100 GPS

Wireless Emergency Stop

Design

When deciding how to implement an emergency stop system many factors were taken into consideration including price, effective range, and reliability of operation. The final decision was to use Xbee wireless transceivers to implement a wireless emergency stop.

To drive the Xbee transceivers, they were coupled with processors to execute the code used to ensure reliable communication. In order to ensure reliable operation the following design rules were implemented:

1. A synchronization signal was transmitted to the receiver periodically. If this signal is not received the vehicle is brought to a stop. This ensures that upon power failure to the transmitter the vehicle will stop.
2. If power to the receiver fails, the vehicle is brought to a stop.
3. If a stop command is issued from the transmitter or the stop button on the vehicle is pressed the vehicle is brought to a stop.
4. Once a stop "state" has been reached the vehicle cannot be placed in a "go" state unless the vehicle is approached and the receiver is manually reset. This ensures the vehicle cannot begin to move again unless intentionally reset.

Implementation

The vehicle is brought to a stop by the removal of power to the drive motors. This is to ensure that the emergency stop has top priority over the vehicle drive motors and cannot be overridden by any of the other vehicle components. This is also ensured in that the ESC can not interrupt the emergency stop process by requesting communication. The ESC may request an emergency stop if the motor controller becomes unresponsive, however the emergency stop does not depend on the ESC and software to function properly.

Long Range Detection

Design

The long range detection system detects any obstacles with a minimum width of 6 inches in front of vehicle. The minimum detection range is 2 feet while the maximum detection range is 15 feet with an accuracy of +/- 1 foot. The coverage angle is 16 degrees with a 2 degree resolution correlating to a system scan time of approximately 110ms.

Fabrication

The system includes 9 Sharp R316-GP2Y0A710YK infrared sensors, a Teensy++ microcontroller board, interfacing cables, a Microchip MCP3001 ADC IC, 4 voltage regulators, 3.3v reference voltage source IC, and a custom printed circuit board.

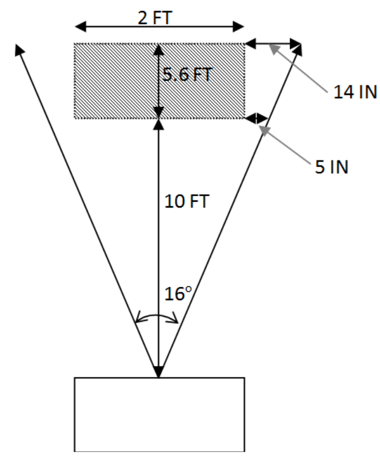


Figure 8: Long Range Field of View

Implementation

Interference avoidance

Each sensor's power source is controlled independently by the microcontroller. Any two sensors which are next to each other cannot be turned on at the same time.

Sensor data sampling

When the sensor power is turned on, the sensor takes 25ms to complete measurements. The sensor output is an analog voltage, ranging from 0 to 3.3v depending on the distance of obstacle from the sensor. The microcontroller uses internal ADC channels and an additional ADC chip to sample the sensor's outputs and place in a data array.

Convert sampled data to distance

The Sharp infrared sensors use a triangulation measurement method. The data reported by the sensors is not linear with the detected distances. As the obstacle moves further away from the sensor, the rate of voltage change decreases. To correctly interpret sensor data, the following equation is used:

$$Distance = \frac{A}{ADC_{value} + B}$$

Constants A, B are determined through sensor calibration.

Object Detection

Instead of reporting all single sensor detections, the system merges collections of detections into a single sensor report. The rule is that if two objects are close and have a distance within +/-1 foot of each other, they are combined to a single object. The system need only report a maximum number of five objects.

Short Range Detection

Design

The short range detection system was designed around the idea that the vehicle should have 360 deg. sensor coverage to protect from bumping into obstacles. In reviewing other team competition runs, it became clear that many other teams did not have a secondary obstacle detection system that could stop the vehicle from hitting obstacles that the vehicle was trying to steer around. For our vehicle, the short range detection system will act as a “bumper” to make sure the vehicle does not make contact with any obstacles. In the initial design, it was considered necessary for the system to be able to detect obstacles at least 10ft away. This consideration was based on a traveling speed of 5mph, an approximated 1second processing time, and a 4ft stopping distance once the stop command has been sent. Many types and models of sensors were considered, but the Ping sensor from Parallax was chosen as the short range detection sensor.

Fabrication

The Ping sensor was chosen because it has a 10.826ft max. sensing distance, operated with 5volts, and used ultrasonic sound which can detect most materials. Analysis was done on the sensor in order to determine the sensing angle using objects based on typical competition course obstacles. Using this sensing angle, a sensor layout for the vehicle was created, which maximized the sensing region, utilized sensor cone overlapping to increase obstacle detection probability, and minimized blind spots. The layout uses 10 sensors spread around the right side, left side and front of the vehicle. The control program initializes the sensors in a manner that prevents adjacent sensors from running at the same time, therefore eliminating adjacent sensor interference. A PCB was created based on the 10 sensor layout, shown in Figure 9. The PCB contains connectors for the sensors, a Teensy++ microcontroller which runs the control program and handles communications to the ESC, multiplexors, communication line connectors, and power regulators. The multiplexors were added to the circuit so that multiple sensor signal lines could be connected to the same external interrupt pin on the Teensy++, since the amount of interrupt pins were limited.

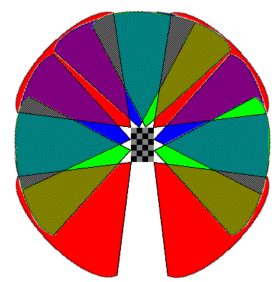


Figure 9: Short Range Field of View

Implementation

The short range detection system takes approximately 60ms per cycle when there are no obstacles in the field of view, and operates variably faster if there are obstacles in the field of view. The sensors are sampled in 3 groups, and the distances seen by the sensors are recorded after each group is sampled. The ESC polls the short range system periodically, at which time the individual distances are retrieved, parsed, and passed to the navigation system. Navigation avoids areas based on distances passed combined with the known sensor cone locations from the sensor layout.

Vision System

Design

The vision system of the robot was designed to be an inexpensive and easy solution, using open source software for image processing and two basic webcams for capturing images. The team decided during the initial design phase that a basic webcam with a pixel resolution of 320x240 would provide enough accuracy so that the vehicle could successfully be navigated through the obstacle course. Implementing the two webcams provides the desired field of view for the system. The images captured with these webcams are processed using Intel's open source software OpenCV. OpenCV is a free solution, and provides many easy to use functions necessary for image processing.

Fabrication

Since the only components needed for the system are the cameras and a laptop for processing, the only fabrication required was for mounting the webcams. The cameras are mounted on the vehicle approximately 4 feet above the ground on a mount which has pan and tilt functionality and is used to position the cameras to achieve the desired field of view.

Implementation

Image processing is done using OpenCV to detect boundary lines and other obstacles the range finding sensors could not detect. The relative position of these detections is then passed to the navigation program to be processed for path planning. The entire image processing algorithm relies on four steps: 1) image is smoothed to filter out noise; 2) all pixel values below a certain color threshold are made black; 3) a Hough transform is applied to the image; 4) the location of these detections with respect to the cameras is relayed to the navigation system. In more detail, the filtering algorithm averages values of pixel clusters to determine the most likely value of each pixel. This process limits noise in the form of random "white pixels" seen in the grass. The next step in the algorithm finds all pixels, thus objects, which are white and/or yellow by looking for certain groups of pixels. Focusing only on certain objects in the image makes the processing done in the next step easier. A Hough transform is then applied to the image to find all lines in the image. The Hough transforms returns these detections in the form of two points defined by x/y coordinates. These coordinates, in pixel space, are then transformed to points in geometric space that are passed to the navigation system. The lengths of these detections are used to determine whether the detection is a boundary line or an obstacle to be avoided such as pot holes and white rings around cones. This information, along with the location of each detection and information from the other sensing equipment, is used by the navigation system to guide the vehicle through the obstacle course.

Figure 10: Camera captures image. White tape was used to simulate boundary lines.

Figure 11: The image is gray scaled, and top half of the image is ignored to minimize false detections. The image is smoothed using OpenCv function cvSmooth(). This function looks at each pixel's neighboring pixel intensity to determine its actual intensity. This eliminates the random white pixel in the grass, and makes the boundary lines a more consistent intensity.

Figure 12: All pixels that don't meet a certain intensity threshold (i.e. aren't white or yellow) are made completely black, and a Hough transform is applied to the image to detect line segments. These line segments are then used to find geographical coordinates the navigation system can use to guide the vehicle.

Vision System Integration

Design

The communication between the navigation and vision systems was implemented using sockets, specifically with the WinSock API for Windows. The vision system was implemented using C++ and the navigation system uses Java. A communication system needed to be chosen that would work well in both languages. While other methods were considered for communication, the WinSock API and sockets were chosen for their ease of use and implementation in each language.

Fabrication/Implementation

To allow uniform communication, the same scene format had to be created in both systems. Scene, line segment, and point classes had to be implemented in both systems with the scene object being composed of an array of line segment objects and each line segment object being composed of a start and finish point. Each point is a double data type value. These classes incorporate the functions and methods necessary to build up each subclass and create the scene object to be passed via the communication system as well as break down the scene for use in plotting the map. In the vision system, classes were also created to replicate the DataOutputStream, DataInputStream, and Persistent classes that are used for socket I/O in Java and included in its libraries.

The Navigation System is the server. When the Navigation is started, the server is set up and waits for the client to connect. Once the client connects it reads the scene objects from the socket and adds them to the PhysicalObject list used to create the navigation system's map. The vision system creates the socket connection when the vision system program is started. After the vision system analysis has run, the scene is created. After each scene is created it is serialized and sent over the socket to the navigation system and used as described above.

Navigation

Obstacle Avoidance

The navigation algorithm recursively determines a path to some given destination. Shown in Figure 13, the algorithm starts by assuming that a turn is impossible before “d” distance. This distance is a determined by a multiplying the current speed of the vehicle and the response time of the motors. Next, it plots five possible turnings from “d” ahead, and calculates whether or not there are any obstacles in the way of those five paths. Any path containing obstacles within a set distance is eliminated, where the distance is chosen through testing to be the most practical distance to go before deciding to turn again. At the endpoints of the all remaining paths, the process is repeated. Turnings are calculated, paths are projected, and paths with obstacles less than “a” distance away are eliminated. The algorithm repeats the process a total of four times, and each final point is given a weight based on how far away it is from the vehicle's ultimate destination. The lowest weight path is chosen, and the motor is given the initial direction command. The algorithm is rerun at predefined intervals to ensure the vehicle is always doing its best to move towards its destination.

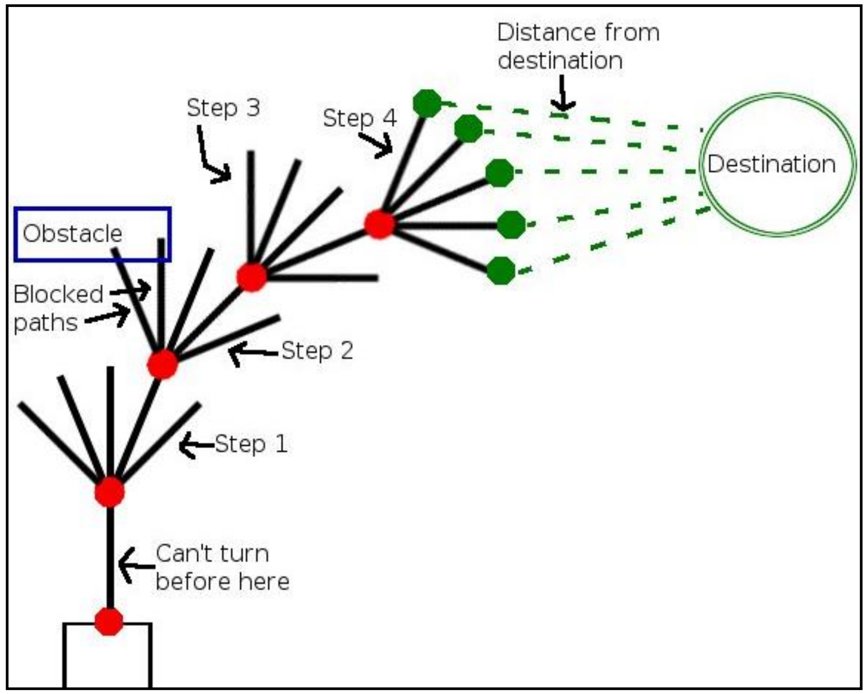


Figure 13: Navigation Path Planning Algorithm

Destination determination

Waypoint Challenge

For the waypoint challenge, the destinations are previously defined, and the vehicle plots a path to which ever happens to be the closest at the time. There is a greedy algorithm in place that determines which destination is closest. This algorithm is rerun periodically so that in cases where the vehicle must go far out its way to reach destination “x”, it does not ignore destination “y” if the vehicle is suddenly closer to it than to destination “x”. Destinations that have already been reached are removed from the list of destinations so that the vehicle doesn't attempt to repeatedly return to the same destination merely because it's closest.

Obstacle Course

For the obstacle course, the destinations are generated based on where we would end up if we followed the line nearest to us for 20 feet. The idea is that the vehicle should stay between the lines on either side, and that by running parallel to the nearest one, the vehicle can continue to move forward without fear of accidently crossing a boundary line. As with the navigation algorithm, this destination calculation is repeatedly regenerated, so that, as long as the vehicle is within 20 feet of a line, the vehicle will continue moving. The vehicle should only stop if it has moved twenty continuous feet without seeing any boundary lines.

Foreseeable problems

Although the vehicle detects lines as obstacles, it has no knowledge of imaginary boundaries. So it is possible that, during empty dashes between lines, the vehicle might go outside of the boundary to avoid an obstacle.

Conclusion

Vehicle Specifications

Table 2 consists of a general overview of vehicle specifications. Certain values were assumed and used in theoretical analysis. After the construction and testing of the vehicle those initial specifications were either verified or refined.

Table 2: Vehicle Specifications

Vehicle Specifications		
Specification	**Predicted**	**Actual**
Obstacle Detection Range	10'/20' (short/long)	10'/15' (short/long)
Outer Dimensions	2'W x 3'L x 6'H	2'W x 3'L x 4.5'H
Payload Capacity	210lbf	TBA
PC Battery Life	45mins	TBA
Response Time	<100ms avg.	TBA
Speed	5mph	TBA
Suspension Travel	0.634"	0.65"
Waypoint Accuracy	0.6"	0.6"
Weight	210lbf	TBA
Wireless E-stop Range	300'	250'

Vehicle Cost

Table 3 represents a generalized overview of the costs associated with constructing the YCP IGV. This table includes both list prices and those paid by the YCP team for various material and equipment integral to development and operation of the vehicle.

Table 3: Budget

Project Budget Overview					
Item	**Manufacturer**	**Quantity**	**MSRP**	**Total**	**Cost to YCP**
Frame	8020 Extrusion	1	$80.00	$80.00	$80.00
Chassis	~	1	$700.00	$700.00	$700.00
Motors	Zinhjiang Unite	2	$69.95	$139.90	$139.90
Gear Box	Anaheim Automation	2	$511.40	$1,022.80	$1,022.80
Encoders	US Digital	2	$215.25	$430.50	$430.50
Motor Drive	Sabertooth	1	$124.99	$124.99	$124.99
GPS	Hemisphere A100	1	$1,495.00	$1,495.00	$1,000.00
Ping Sensors	Parallax Ping)))	12	$28.99	$347.88	$347.88
Cameras	Microsoft VX2000	2	$39.99	$79.98	$79.98
IR Sensors	Sharp	10	$29.99	$299.90	$299.90
PCBs	Sunstone Circuits	6	$89.16	$534.96	$35.00
Controllers	PJRC/Arduino	7	$25.71	$179.97	$179.97
MISC. Electronics	Digikey/Sparkfun	1	$700.00	$700.00	$700.00
MISC. Hardware	McMaster	1	$100.00	$100.00	$100.00
Laptop PC	Lenovo ThinkPad T61	1	$1,195.00	$1,195.00	$0.00
			Total	$7,430.88	$5,240.92

Summary of Project

At this stage in the project there is a limited understanding of the vehicle's shortcomings. Currently there are concerns that the testing period for the vehicle will be shorter than originally anticipated. Part of any project incorporates time allocation. As this was the first year the YCP Engineering program took part in the IGVC, there was little feel for how much time should have been assigned towards different aspects of the project. From that perspective there really have been no major pitfalls during this year's competition preparation. This leads into the sense of accomplishment that the team does feel. This year, YCP undertook a new senior design competition with its newest addition to the Engineering curriculum, the Electrical Computer Engineering program (ECE). It was a challenging endeavor to pursue, not only from a technical standpoint but also from an organizational one. Prior to this project, most if not all of the preceding senior design projects have been composed of a single discipline. This undertaking has both forced and allowed the team members to experience working in collaboration with students from other technical fields, effectively expanding each team member's knowledge base and communication skills. The vehicle in its entirety is a design innovation. Many other vehicles entered in the competition are composed of preassembled materials. *Green Lightning* has been completely designed from the ground up by the YCP team. The structural elements that comprise the vehicle such as the track system and frame were constructed from stock materials. The track and the frame material were machined in house to accommodate design needs. The tracks offer IGVC specific terrain customized performance, in that they are designed to accommodate the terrain present as well as the expected types of perturbation. The frame has been designed in such a way that it can accommodate the mounting of all the required sensors and minimize the overall size of the vehicle. The electronic components beginning with the ESC have all been designed as custom equipment, specifically tailored for the vehicle. The vehicle status is monitored via a custom written java application. The vision system implements custom written code to detect lines and obstacles. The detection equipment both long and short range utilize new approaches towards implementation of their sensors. The navigation program was completely developed by the YCP CS students and contains no prewritten code. In summary, *Green Lightening* is a representation of the collaborative efforts of the YCP Engineering and CS disciplines and showcases the talents that York College students in these fields have to offer.