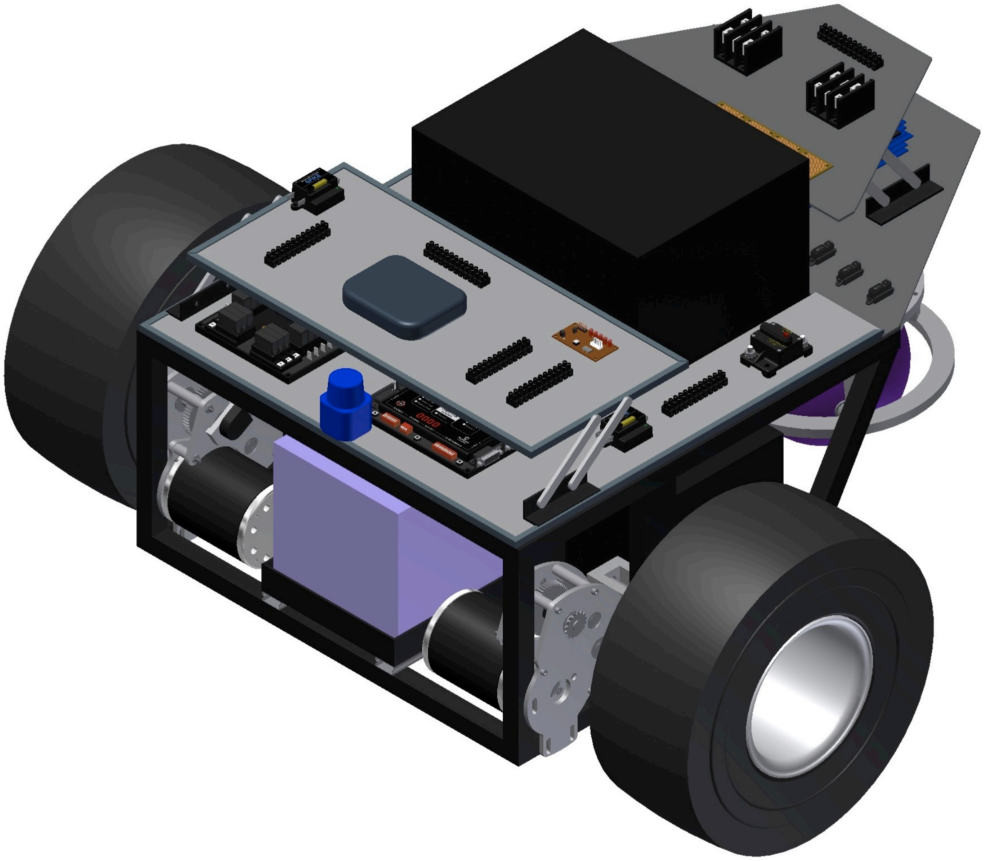
MARGARITA

The Glendale Community College Robotics Team's

entry in the

2012 Intelligent Ground Vehicle Competition



Team Members:
Ivan Aguilar
Ani Aslanian
Adam Berman
Araz Boghozian
David Buickians
Matt Conz
Shara Ebrahimi
Samuel Ekstrand
Aram Garibyan
Grant Gates
Arian Ghazari
Michel Grigory
Pedro Kim
Marek Litomisky
Rafi Meguerdijian
Aaron Merrill
Torkom Pailevanian
Antonella Wilby

Faculty Advisors:
John Gerz
Thomas Voden

Faculty Advisor Statement:

I certify that the engineering design of the vehicle described in this report, Margarita, has been significant, and that the student effort is equivalent to a senior design capstone project.

Thomas Voden, PhD
Mathematics Division
Glendale Community College

1.0 Introduction

The Glendale Community College Robotics Team is pleased to present Margarita for its inaugural entry into the Intelligent Ground Vehicle Competition. Our goal with this project was to provide Glendale College's engineering students with the opportunity to work on a real-world engineering project that would provide them with skills that would be invaluable in their future educations and careers. These opportunities exist at universities, but are far rarer at the two-year college level.

The robot was designed by our team of freshman- and sophomore-level engineering and computer science students from a variety of backgrounds. Most of our team's participants will be going on to complete their bachelor's degrees at four year-universities.

1.1 Team Structure

Our team is organized into three subteams — mechanical, electrical, and software. Communication between the subteams is facilitated by the team leader and faculty advisors.

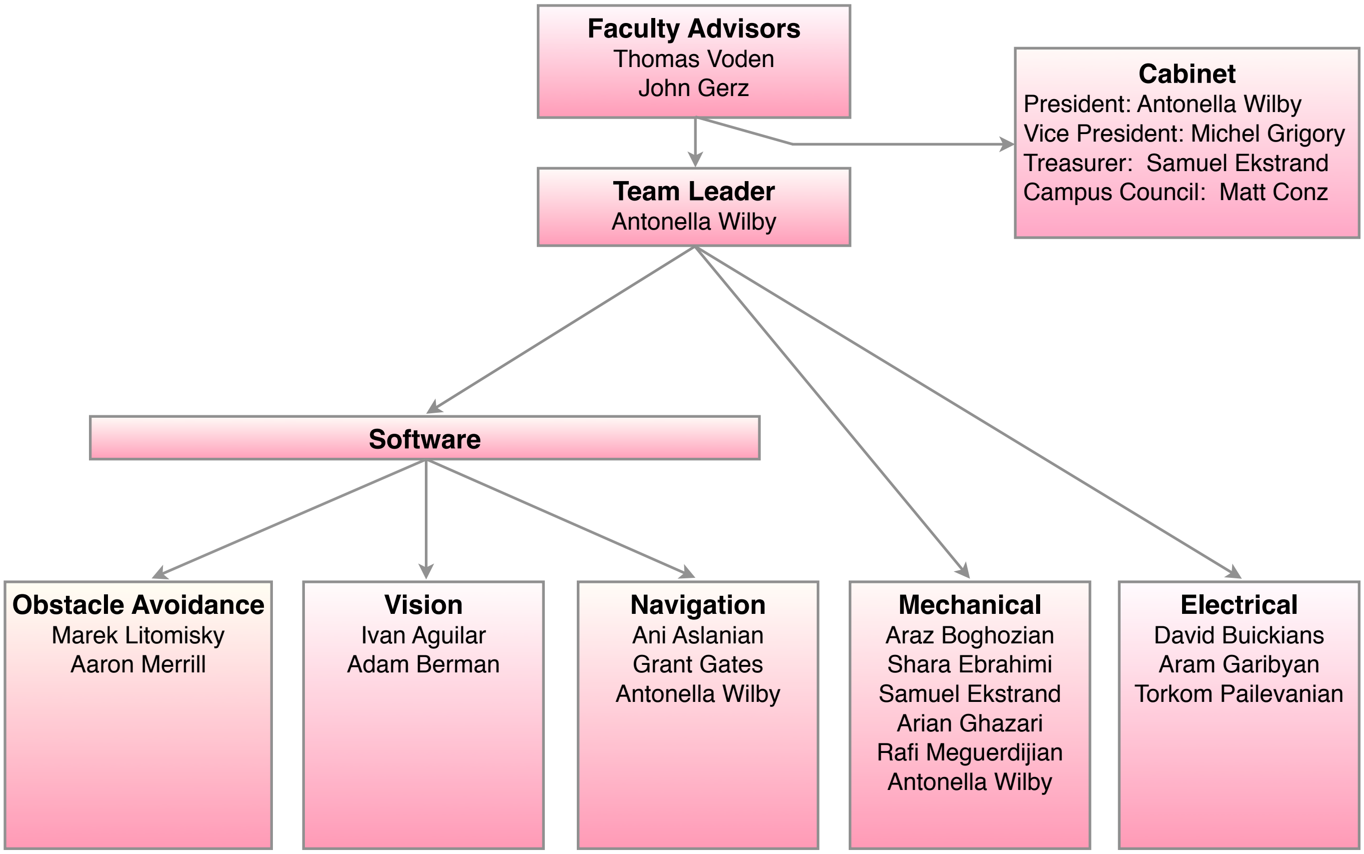


Figure 1.1 Team structure

2.0 Mechanical

The mechanical subsystem of the robot was developed in several stages over a period of two years. In the first year, the team designed and built a robot that was intended for last year's competition, but which ultimately functioned as a prototype robot and software development platform. During the second year, the team analyzed the design flaws of the prototype and used this understanding to improve the design of the competition robot.

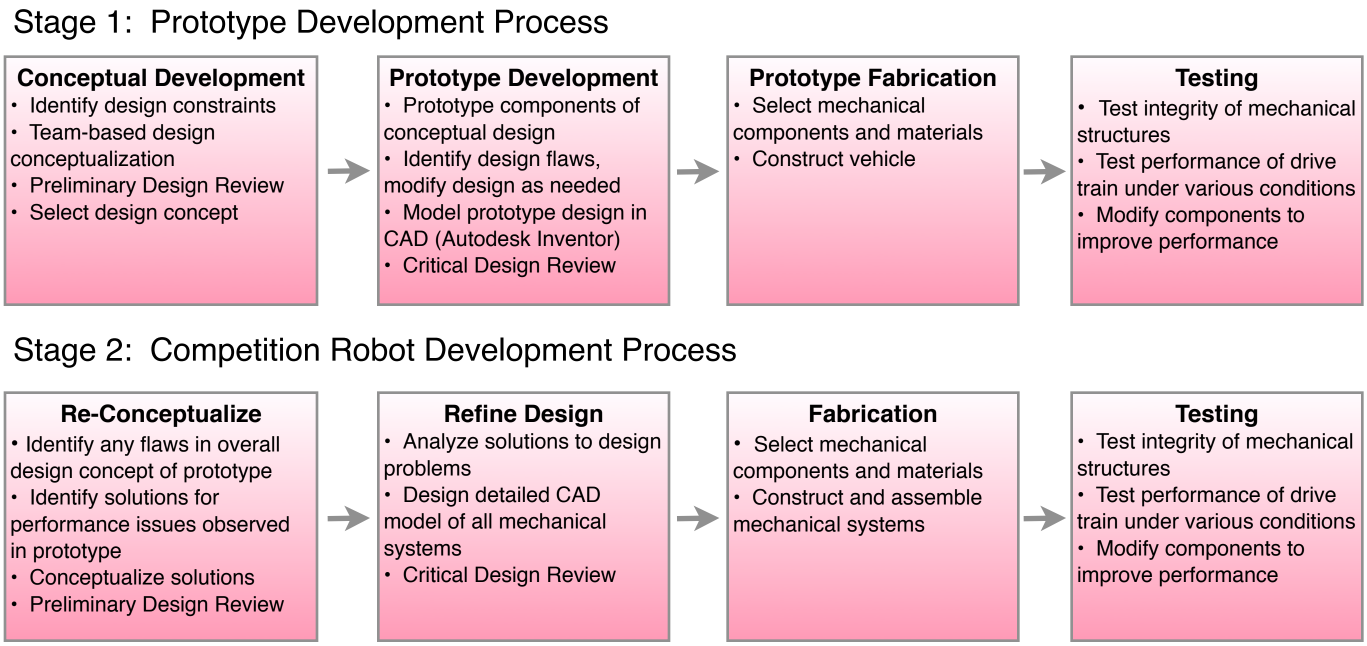


Figure 2.1 Mechanical development process

2.1 Design Constraints

Our robot was designed to meet the design criteria outlined by IGVC. Our vehicle measures approximately 38 inches long, meeting the minimum length requirement of three feet, and it is approximately 36 inches wide, meeting the minimum width requirement of two feet but not exceeding the maximum width requirement of four feet. Its height of approximately 24 inches (without GPS antenna; 60 inches with antenna) does not exceed the maximum of six feet. The drive train is geared to ensure that the robot can achieve the requirement of a minimum average speed of one mile per hour while not exceeding the maximum speed of ten miles per hour. Additionally, the requirements for the mechanical E-stop, the wireless E-stop, and the safety light were met. Fulfilling the requirements for lane following, obstacle avoidance, and waypoint navigation were the key objectives in the development of the robot's software.

In addition to the design constraints imposed by IGVC, we decided to impose our own design constraint of a maximum robot weight of 200 pounds without the payload (220 pounds with the payload). This decision was made to keep the robot light enough to be lifted and to ensure that it would not become so heavy that it would need bigger motors and/or batteries. Every component was designed with weight in mind.

2.2 Power Requirements

Based on the given design constraints, we calculated the necessary power for the robot's drive train. Gearing efficiency was assumed to be 90%, and the coefficients of rolling friction over plywood and grass were taken to be 0.2 and 0.3 respectively, based on published data. We assumed the robot weight to be 220 pounds at maximum. Power requirements for the motors were determined based on these assumptions, and a desired maximum robot speed of 6 miles per hour. We also factored in a margin of safety for the 15° incline, calculating the required power for a 17° incline instead.

	Net Force	Required Torque	Required Power	Motor Power	Power per Side
Case 1: Accelerating on grass to full speed	78.1 lbs	52.7 ft-lbs	939 W	1040 W	520 W
Case 2: Climbing a 17° incline	106 lbs	71.8 ft-lbs	1280 W	1420 W	710 W

Table 2.1 Power requirements

2.3 Chassis Design

Margarita's chassis was designed to be significantly more streamlined than that of our prototype robot. The chassis was made slightly narrower in order to make the robot better able to fit through small gaps on the course. This reduced space presented a challenge when determining the placement of the major components, but the optimum placement for all components was eventually reached. The heaviest components on the robot — in particular the motors, payload, batteries, and computer — were all carefully positioned to give the robot the lowest possible center of gravity and the tightest possible turning radius.

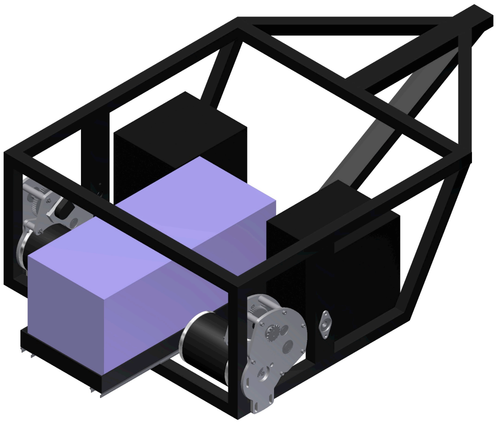


Figure 2.2 Chassis with payload, batteries, and gearboxes

The frame was designed to be streamlined, light-weight, and high-strength. Square aluminum 6061 tubing was chosen for the material because of its high strength to weight ratio. The chassis kept a similar shape to the prototype robot, but was entirely redesigned to have significantly fewer weld joints, both to reduce welding time and complexity and the chance of mechanical failure.

2.4 Drive Train

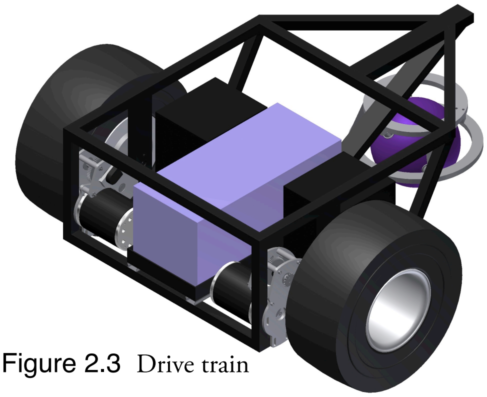


Figure 2.3 Drive train

A differentially steered drive system was chosen for the robot's drive train. In this configuration, the drive wheels are independently controlled and located on the same axis, and robot velocity and steering is determined by differences in each wheel's velocity. Because the system has two non-independent kinematic constraints, the problems of both mechanical design and programming the control commands are simplified.

The drive configuration has three wheels — two drive wheels located in the front of the robot, and a rear stabilizing caster. Two 16 inch turf tires, originally intended for use on golf carts or riding mowers, were purchased to use for the robot's drive wheels because of their demonstrated performance on grass and other rough terrain. Each wheel is powered by a 24-volt MY1020 electric scooter motor (the lowest-cost option that met our power requirements) which is geared down by a custom gearbox providing a reduction of about 24:1.

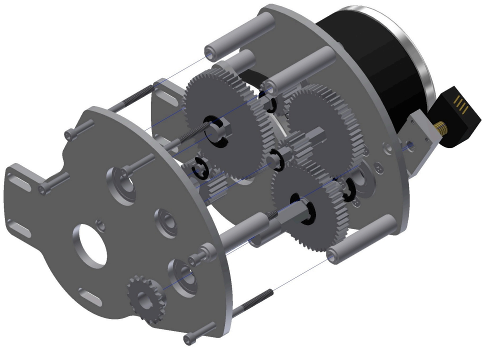


Figure 2.4 Gearbox (exploded view)

The gearboxes were designed because there were no simple off-the-shelf options available for gearing down our chosen motors. The main challenge faced in their design was to fit them in the small space available between the payload and the batteries. For this reason, the gearbox housing is split into three plates, so that the gearing stages occupy more of the space around the motor (see Figure 2.4). The curved profiles of the gearbox plates were cut using a water-jet, and the other machined parts were made using a Computer Numerical Control (CNC) milling machine and a manual lathe.

2.5 Ball Caster

The rear stabilizing wheel was initially designed to be a ball caster. Rather than using a regular caster, which must continuously realign itself with the direction of the robot's motion and which was observed to interfere with steering on our prototype robot, a ball caster offers 360 degrees of unimpeded motion. However, it presented significant design challenges, such as constructing the caster's socket so the ball can move as smoothly as possible; keeping the caster clean on sand and grass so the performance efficiency remains maximal; and minimizing the deformation of the ball so rolling remains unobstructed.

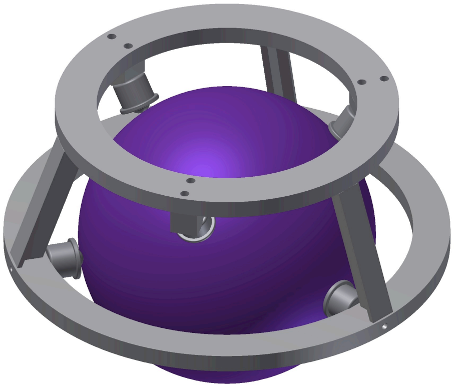


Figure 2.5 Ball caster modeled in Autodesk Inventor

Our design consisted of a water polo ball, chosen because it would provide some traction at the interface between the ball and the terrain, but still would move smoothly in the housing. The housing held six ball transfers, which kept the ball in place while providing 360 degrees of unobstructed motion at the contact points. The team completed a positive proof-of-concept test for the ball caster, but due to lack of time and design resources was unable to finish manufacturing the final product. We decided it would be a worthwhile tradeoff to use a regular caster and focus our design and fabrication resources on the rest of the drive train.

2.6 Weatherproofing

A light-weight weatherproofing shell was designed to enclose the entire robot. The shell was manufactured using vacuum-forming. First, the shape of the shell was cut by a CNC out of a large piece of foam. Then, a thin piece of plastic was vacuum-formed over the foam, creating the shell. This shell can be easily removed when parts inside the robot need to be serviced, but protects the electronics and other sensitive components from water when in place.

3.0 Electrical

The electrical subsystem is the interface between the mechanical systems of the robot and the software. The main objectives of the subteam were to ensure reliable operation of all electrical components and to properly isolate the expensive sensors from shorts, surges, or other damage.

3.1 Electronics Layout

The physical layout of the electronics consists of four electrical boards arranged in two levels. The lower electrical boards hold the larger and/or heavier components, such as the computer and motor controller. The upper electronics boards hold smaller components, like the wireless router, certain sensors (digital compass, GPS), relays, switches, and other components. The upper boards can slide outward, facilitating easier access to the components on the lower electronics boards, or slide inward and lock in place to keep components secure when the robot is moving.

3.2 Power Distribution

Figure 3.1 diagrams the power distribution system of the robot. The primary components that must receive power are the motor controller and the computer. The motors and sensors are considered secondary components in the power diagram because they are powered by the motor controller and the computer, respectively.

Both the mechanical and wireless E-stops are wired to cut power to the motor controller, thus stopping the motors. The computer and sensors are controlled only by the main switch.

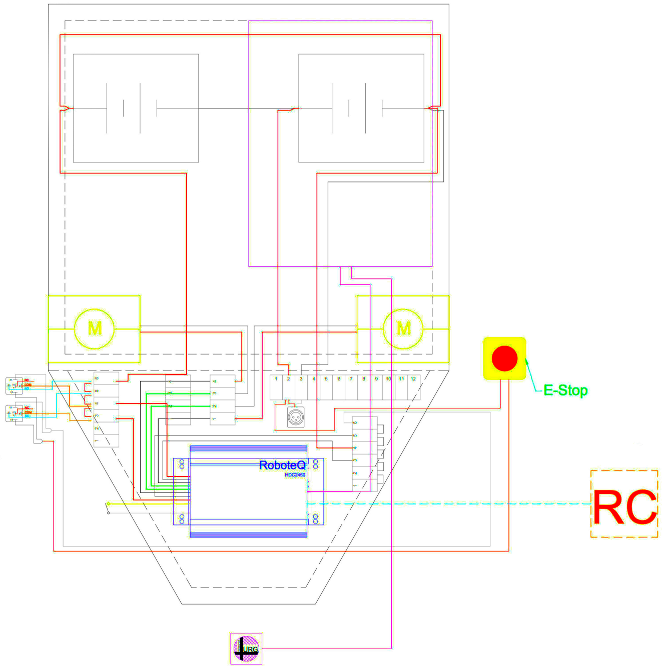


Figure 3.1 Power system schematic

3.3 Batteries

The robot's power is supplied by two 12 volt electric scooter batteries, connected in series to create a 24 volt electrical system. The batteries each provide 26 amp-hours, giving the robot approximately 45 minutes of operation time per charge.

Figure 3.2 12-V scooter batteries

3.4 Motor Controller

Actuation commands are sent from the motor controller to the motors after a path is determined by the path planning algorithm. The motor controller is a Roboteq HDC2450 two channel controller, with inputs for two encoders. It connects to the computer via a USB 2.0 connection.

Figure 3.3 Roboteq HDC2450 motor controller

4.0 Sensors

Margarita's sensor system incorporates six sensors — a laser rangefinder, an iSight camera, a digital compass, a GPS, and two optical encoders. The camera and the rangefinder are secured to the vehicle with vibration-dampening mounts so that the effects of vehicular motion and vibration are minimized, because these sensors are of particular importance to the software. Care was also taken in the placement and mounting of all the other sensors, especially concerning vibrations and weatherproofing.

4.1 Sensor Integration

Sensor integration is performed by Margarita's on-board computer. This computer was designed from scratch to be able to communicate with the sensors and to handle the computational tasks for all of our vision, navigation, and obstacle avoidance algorithms. It has a 64 GB solid state drive, chosen both for its speed and durability. Fast processing is achieved by a six-core AMD Phenom X6 3.2GHz processor with four GB of DDR3 RAM. The computer is equipped with several USB 2.0, firewire, ethernet, and serial ports to facilitate communication with all the sensors. The computer uses a DC to DC power supply, which eliminates the need for a DC to AC power inverter and allows the computer to run directly off battery power, improving the efficiency of our electrical system.

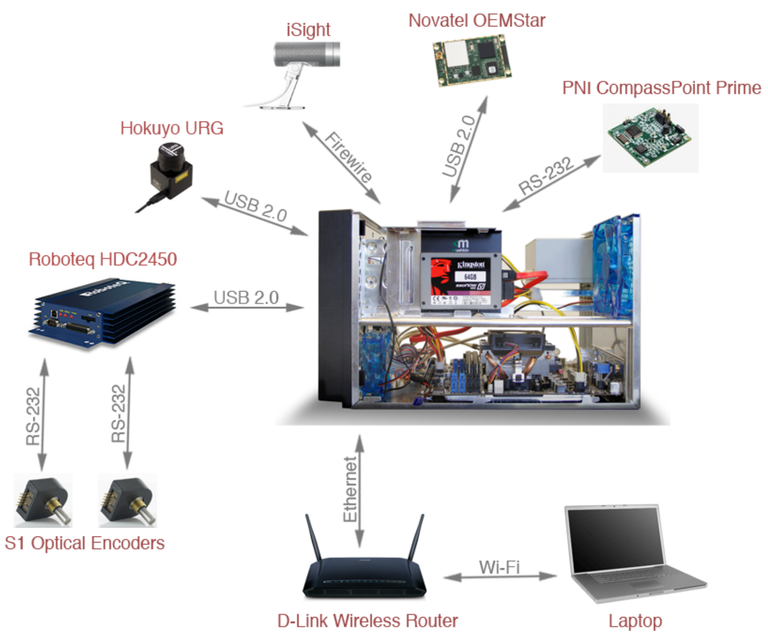


Figure 4.1 Sensor integration on Margarita

4.2 Laser Rangefinder

Obstacle detection is accomplished by a Hokuyo URG laser rangefinder. This rangefinder scans a range of 240° with an angular resolution of 0.36°. The rangefinder can detect objects as close as 20 millimeters and as far away as 5.6 meters. The obstacle avoidance algorithm was designed around these specifications. The rangefinder interfaces with the computer via a USB 2.0 connection.

Figure 4.2 Hokuyo URG laser rangefinder

4.3 Camera

Figure 4.3 Apple iSight camera

The vision system consists of an Apple iSight webcam. Its main purpose is for the lane following algorithm, the detection of potholes, and the detection of the red and green flags, but it also assists in obstacle detection. The iSight's 1/4" color CCD sensor has a resolution of 640x480 pixels and captures video at 30 frames per second.

4.4 Digital Compass

Figure 4.4 PNI digital compass

The PNI CompassPoint Prime digital compass helps determine robot heading. The compass provides a heading accuracy of 1° and samples at a maximum of 10 Hz. It communicates with the computer via an RS-232 connection.

4.5 GPS

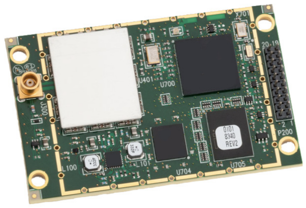


Figure 4.5 Novatel OEMStar GPS

The GPS provides positioning data for the Navigation Challenge. Our GPS is a low-cost Novatel OEMStar. Many teams in IGVC use a differential GPS unit, which provides much greater location accuracy, but budget constraints precluded use of one of these expensive sensors on our robot. However, the OEMStar still provides fair accuracy of 1.5 meters to 0.5 meters. We have not yet determined whether this accuracy will be sufficient in the Navigation Challenge. Because the waypoints are small in size (2 meters in diameter), our vehicle will have to navigate quite precisely to achieve success.

4.6 Encoders

Figure 4.6 Optical shaft encoder

The US Digital S1 optical shaft encoders provide feedback about the speed of each drive wheel to the software system. The encoders use x1 quadrature encoding to provide 1250 positions per revolution, which is a resolution of 0.288°. Based on the robot's drive wheel diameter of 16.2 inches, the encoders can detect the robot's changing position with approximately 0.0407 inches resolution. The encoders are mounted to the output shaft of the gearbox with a custom mounting bracket.

5.0 Software

At the time of this writing, the vast majority of our software packages are still under development. The following sections outline both work that has already been accomplished and the plan for work that will be accomplished in the next month before competition.

5.1 Software Architecture

Our software is based on the Robot Operating System (ROS), an open-source software framework for robotics applications which was developed primarily at Willow Garage. Margarita runs the full version of ROS on Ubuntu Linux. This framework was chosen because it takes care of much of the low-level communication between the hardware components, allowing the team's programmers to focus on development of higher-level algorithms.

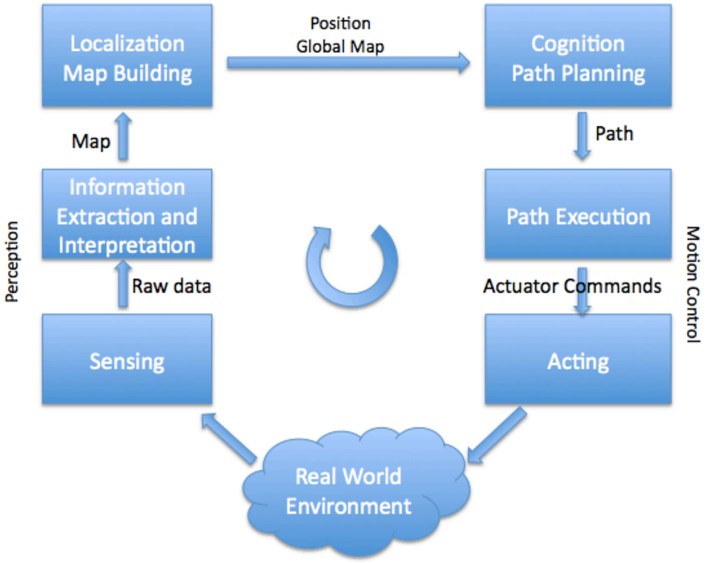


Figure 5.1 Software control scheme[4]

Within the framework of ROS, our software is organized into packages and stacks that provide all of our software functionality. We used both the C++ and Python programming languages, depending on the preference of each programmer, because ROS can seamlessly integrate the languages.

When designing our software, we primarily referenced the control scheme in Figure 5.1. The packages that make up our software fall into one of four categories in the scheme. Vision (including line and color detection) and obstacle detection fall in the category of *Perception*, where first the real world environment is sensed and then information is extracted from the raw data. This information is sent to an algorithm in the *Mapping* stage that continuously populates its global map and localizes the position of the robot within that map. This map, along with the robot's position, is then sent to the *Cognition and Path Planning* stage which plans a trajectory through the course based on the obstacle positions and lane directions in the map. In the final stage, *Motion Control*, the computer sends the commands for the path to the motor controller which sends actuation commands to the motors.

5.2 Vision

Vision processing on Margarita is a continuous process — vision data is acquired from the camera, the image is filtered and enhanced, noise is reduced, and lines and colors are extracted from the image. The lines are critical in path planning, and are continuously used to determine the robot's heading by the planning algorithm. Color detection is used only for the red and green flags at the end of the course.

5.2.1 Line Detection

Lines are detected in the streaming video using a Hough transform. This edge detection algorithm is frequently used to detect shapes such as lines, circles, or ellipses. Because the lane lines will consist of smooth curves and straight lines, this is a good algorithm for us to use.

Original, uncorrected image

Image with heightened contrast

Image converted to grayscale

Results of Hough transform

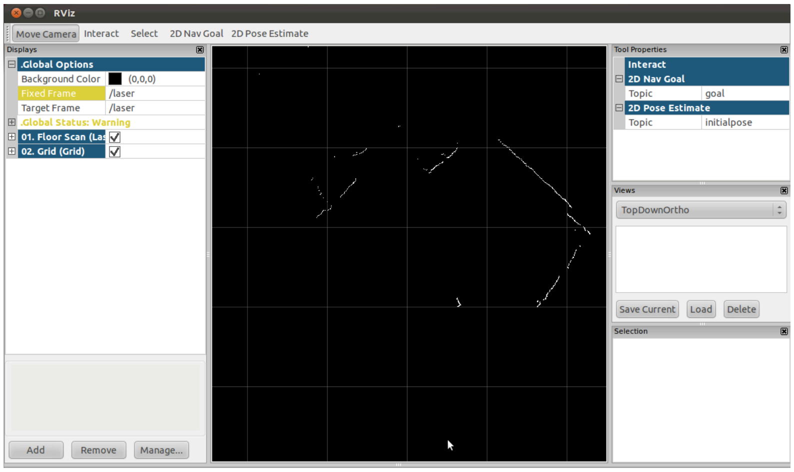
Figure 5.2 Hough transform process

5.2.2 Color Detection

Color detection is performed by the same camera that detects lines. Because this algorithm is only used to detect the red and green flags that appear at the end of the course, it does not affect the operation of the software (mapping, path planning, etc.) for the majority of the time. When the red and green flags *are* detected, it triggers a subroutine for the path planning algorithm that takes into account the locations of the flags when calculating the path (steering to the right of the green flags and the left of the red).

5.3 Obstacle Detection

Obstacle detection is performed using data from the laser rangefinder. The rangefinder continuously scans a 240° range for obstacles at a rate of 10 Hz, returning the locations of a cluster of points defining an obstacle and the distances to those points. This information is then added to the local map, which already contains the lane information.



.Global Options
Background Color (0,0,0)
Fixed Frame /laser
Target Frame /laser
.Global Status: Warning
01. Floor Scan (Las ☑
02. Grid (Grid) ☑

Interact
2D Nav Goal
Topic goal
2D Pose Estimate
Topic initialpose

TopDownOrtho

Figure 5.3 Raw output from the laser rangefinder. This data is used to populate the occupancy map.

5.4 Mapping and Localization

Our robot performs mapping and localization in its environment using the technique of Simultaneous Localization and Mapping (SLAM). The map is built using lane data from the vision system and obstacle data from the rangefinder. Our software uses a graph-based SLAM technique, which interprets the SLAM problem as a graph of nodes and constraints between nodes. Nodes represent robot locations and map features (obstacles, lanes, etc.). Constraints are the relative positions between robot locations and the features observed from those locations.[4]

5.5 Path Planning

In keeping with the control scheme in Figure 5.1, the path planning algorithm is fed the continuously updating global map and the robot's current position, both of which it uses to compute a path. The global map is simply a representation of the environment's free and occupied space. Given this occupancy map, the path planner must decompose the information into a graph that can be efficiently searched using any graph search algorithm. Our path planning software chooses to represent the occupancy map as a Voronoi diagram, a road map representation consisting of lines connecting all the points that are equidistant from two or more obstacles. One advantage of this representation is that the road maps that are computed always maximize the distance between obstacles and the robot. Another is that algorithms that find paths on the Voronoi diagram are complete (i.e., will always find a path if one exists) because a path's existence in free space implies that a path must exist on the Voronoi diagram. Because this approach requires a lot of memory, especially for large environments, it is not as efficient as (for example) a coarse resolution Cartesian occupancy grid might be; however, the advantages it affords in simpler motion control combined with our computer's increased memory make it a good option for computing road maps.

Once the Voronoi diagram is computed, the software uses the A* graph search algorithm to find the optimal path from the road maps in the diagram. Because it uses heuristics, A* achieves better performance than other graph search algorithms, making it ideal for our application. Once the path is computed, actuation commands are sent to the motor controller.

5.6 Algorithms Summary

5.6.1 Lane Following

Lane following is performed by the path planning algorithm based on the occupancy map built from both vision data and rangefinder data. Lanes are extracted from the video feed using a Hough transform, then the data is built into the occupancy map. General lane following can be performed by the path planning algorithm because lanes are treated as walls. Then road maps are computed and the A* algorithm finds the best path.

5.6.2 Obstacle Avoidance

Obstacle avoidance is performed by the same path planning algorithm that also integrates lane following. Once the map is populated by data from the rangefinder, road maps are computed and the A* algorithm finds the optimal path.

5.6.3 Waypoint Navigation

The waypoint navigation algorithm uses the path planning algorithm that combines both lane following and obstacle avoidance as its underlying avoidance algorithm while it navigates to the waypoints. Because the location of the robot in the global reference frame is known because of feedback from the GPS, the distance to each waypoint can be calculated. Thus the problem of choosing the order in which to navigate to the waypoints becomes the Travelling Salesman Problem, an optimization problem. Our approach for solving this problem uses the heuristic nearest neighbor (NN) algorithm. This algorithm always chooses the nearest unvisited waypoint as the next destination. While there are waypoint arrangements for which the NN algorithm returns the worst-possible route (as measured by length of actual route versus the length of the shortest existing route), it has the advantages of returning relatively short routes for many arrangements of waypoints and using logic that is simple to implement.

5.7 JAUS

Margarita's software does not incorporate JAUS protocols.

6.0 Cost

Subsystem	Part	Qty.	List Cost	Cost to Team
Mechanical	1" x 1" x 1/8" Aluminum tubing for chassis		95.71	0.00
	1" x 2" x 1/8" Aluminum tubing for chassis		148.80	0.00
	Gearbox materials and components		268.95	268.95
	MY1020 Electric scooter motors	2	179.90	179.90
	Kenda turf tires and wheels	2	99.98	99.98
	Pneumatic caster	1	20.00	20.00
	Miscellaneous hardware (approximate)		70.00	70.00
Sensors	Hokuyo URG-04LX-UG01 Laser Rangefinder	1	1173.00	1114.35
	Apple iSight Camera	1	160.00	0.00
	PNI CompassPoint Prime Digital Compass	1	400.00	0.00
	Novatel OEMStar GPS	1	123.50	123.50
	Antcom L1 Active GPS Antenna	1	180.50	180.50
	US Digital S1 Optical Shaft Encoders	2	191.34	170.34
	Shielded cable assembly, 5 pin locking connector	2	61.50	61.50
Electrical	Roboteq HDC2450 motor controller	1	645.00	645.00
	Computer (approximate)	1	1500.00	1500.00
	12V Electric scooter batteries	2	239.90	239.90
	D-Link wireless router	1	27.88	27.88
	LED strobe safety light	1	34.95	34.95

Subsystem	Part	Qty.	List Cost	Cost to Team
	XO Vision DC to AC power inverter	1	30.00	30.00
	ABS plastic sheeting for electronics boards		55.80	55.80
	Miscellaneous electronics (approximate)		100.00	100.00
Total			**5806.71**	**4922.55**

Table 6.1 Cost of Margarita

7.0 Acknowledgements

Without the contributions of several people at Glendale College and in the community, the construction of Margarita would not have been possible. The team would like to thank John Francis and his welding students for their work on welding the frame; Aram Ohanis for his hours of manufacturing time and instruction on CNC and manual machining; and David Black, for his guidance on the design of the mechanical subsystem. We would also like to thank John Gerz, our faculty co-advisor, for providing a workspace in the physics lab and for advising both our presentation skills and the team outreach effort. And finally, we would like to thank our faculty advisor Thomas Voden for his dedication to our project throughout this entire process. Without his long hours of support and guidance on team coordination, funding, leadership, design and construction, and in countless other areas, this robot would never have become a reality.

8.0 References

[1] H. Choset, K.M. Lynch, S. Hutchinson et. al., *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Cambridge, MA: The MIT Press, 2005.

[2] S. M. LaValle, *Planning Algorithms*. Cambridge, UK: Cambridge University Press, 2006.

[3] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Prentice Hall, 2010.

[4] R. Siegwart, I.R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*. Cambridge, MA: The MIT Press, 2011.

[5] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: The MIT Press, 2006.