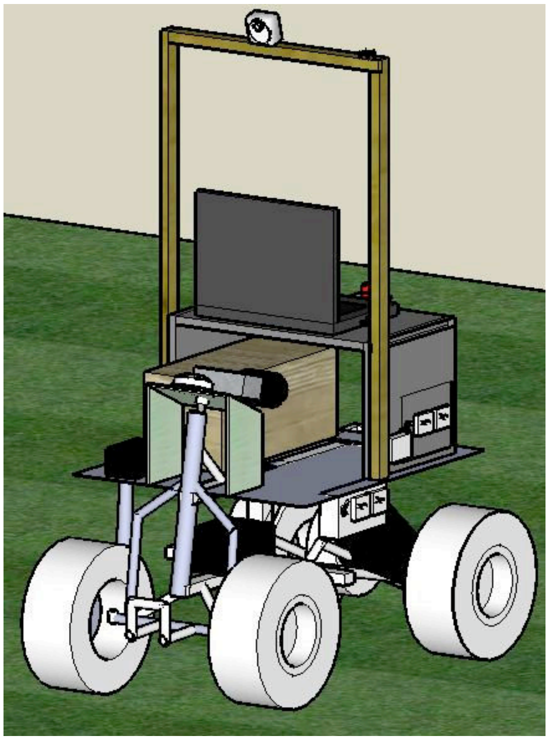
The Bob Jones University
Engineering Team
Presents to You

cornelius



IGVC 2008

Team Members:

Benjamin Robinson–Computer Engineering

Zachary Mason–Computer Engineering

Richard Reece–Computer Engineering

Andrew Marsh–Electrical Engineering

Lance Sweeley–Electrical Engineering

Peter Keew–Electrical Engineering

Jeremy Gray–Electronics and Computer Technologies

Table of Contents

I. Innovations

Cornelius was created completely from scratch this year with a new design approach in mind. For the first time ever Bob Jones University has decided to create a robot with Ackerman steering. This decision brings many challenges with it; however, these challenges are the exact ones that the United States Armed Forces face trying to automate weapons platforms.

Cornelius has several new and innovative features. A folding tower allows for easy transport and storage. Also a solar powered battery charger was designed to recharge our drive motor batteries. A homemade infrared sensor array was designed for obstacle detection. Five individual IR sensors are mounted in a pattern that gives Cornelius 125 degree forward field of obstacle detection and one IR sensor is in the back that gives Cornelius 25 degree rear field of obstacle detection. Cornelius is implementing lane detection rather than white line avoidance to better navigate through sand traps and to stay in bounds when encountering dashed line segments. Another innovation is designing our software from the ground up to be JAUS compliant in order to modularize the code, simplify the threading code, and effectively adapt to future competition requirements related to JAUS. Last, for testing and transportation, we designed a JAUS message-based remote control.

II. Team Organization and Planning

As a team, we planned our design strategy by first assessing what specifications the robot would require. Each member of our team researched components that would fill these requirements. We kept each other up to date by using a forum that automatically emailed the discussions, and serves as a running documentation for our work. For an organized synopsis of the design, we maintained a Wiki to keep all important facts and figures necessary for the education of future teams and to ensure that we did not duplicate individual member's research work. Our software strategy was to assign all high level functionality and JAUS compliance to one or two members and low level drivers and algorithms to other members. This plan forced us to clearly define our interfaces before writing code to ensure compatibility at integration time. In order to test code before hardware integration completion, we decided to integrate an OpenGL simulator program written by a former student to test image processing routines and navigation algorithms. Last, we created a GANT chart to map out the hardware design for the purpose of ordering parts on time and keeping the integration process moving as components began to arrive.

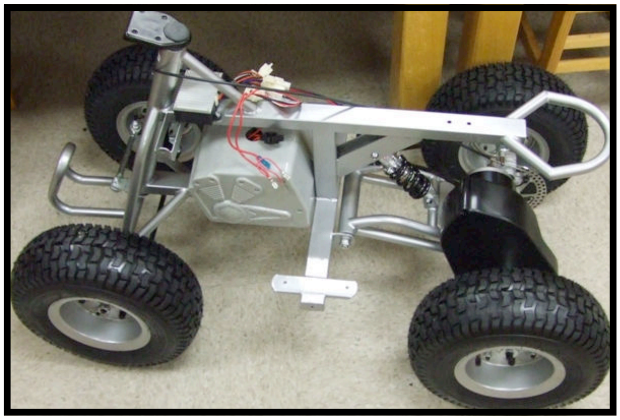
III. Hardware Design

Google Sketch-up was used significantly during our design process in determining the amount of space needed to fit all of the different components on the robot. Each team member was tasked to create a robot component in Google Sketch-up in order to have a visual of where everything would be placed. Google Sketch-up allowed us to determine the need for an extra "shelf" in order to create space for the laptop, since there was no room after the placement of the payload. This is just one of many examples where pre-planning in Sketch-up made a difference in our design.

Chassis

One option for the type of steering that can be used for the IGVC competition was Ackerman steering. We decided to use Ackerman steering due to the challenge of not being able to turn with a zero degree radius. Ackerman steering also provides us with the exact same challenge that the U.S. military faces when trying to automate a machine such as a hummer. The chassis of our robot is a stripped down version of an electric ATV. We particularly liked this model due to the fact that it includes an electric drive motor along with batteries that are designed to drive the motor.

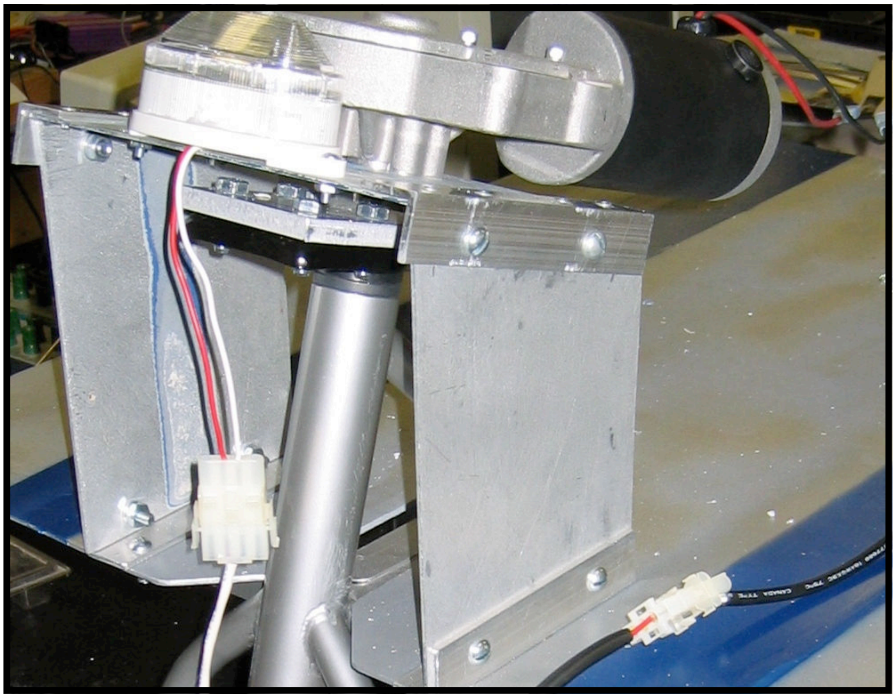
Manufacture picture of electric ATV



Electric ATV stripped down to frame

Steering

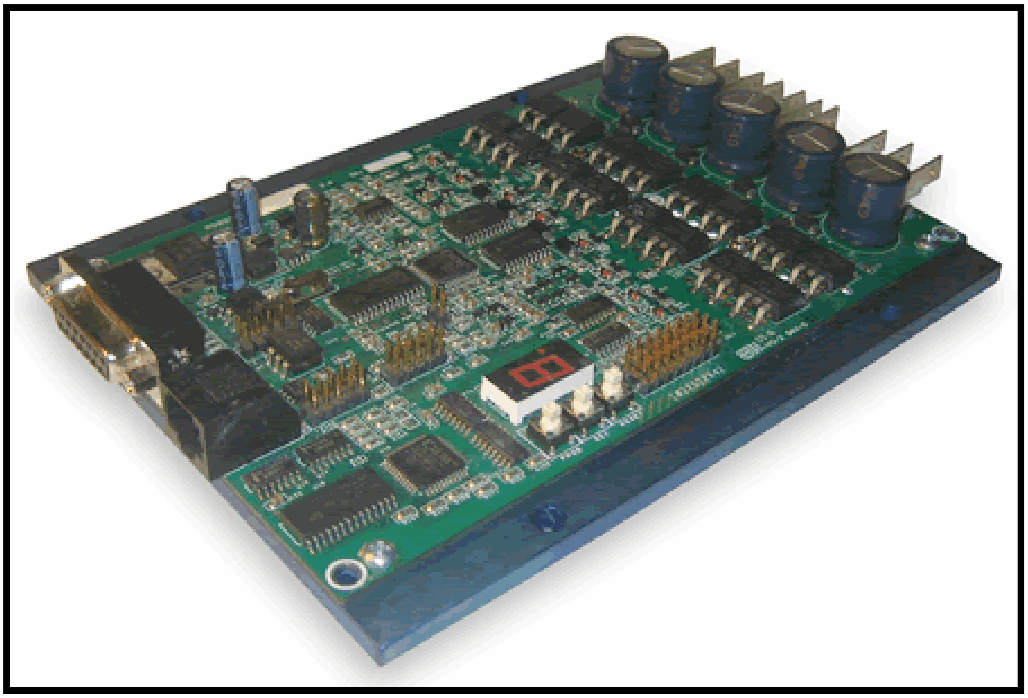
Mounting the steering motor was a major obstacle in the construction of Cornelius. In order to get a good position for the motor, the handle bars were taken off and a bracket was made to give the motor proper height and angle in relation to the steering shaft. A metal plate was also machined and bolted onto the steering shaft with a key slot. The key allows for protection of the motor and supporting brackets if the steering turns too far or too fast.



Steering Motor, Bracket, and Slotted Plate

Motor Controller

The motor controller was selected to be able to perform several key features for the robot. It has the capability of running one channel in closed loop speed mode while running the other channel in closed loop position mode. We choose to utilize the speed control with closed loop feedback for our drive and position for the steering. The controller has inputs for feedback devices like an encoder or a potentiometer. We used an encoder for the speed sensor and a potentiometer for feedback on the steering. The controller, a RoboTeq AX3500, outputs an analog signal through the pot and then reads the voltage on an input in the controller. The change in voltage is recognized as a change in position. The potentiometer changes resistance as it turns and is turned by a bolt that turns with the steering. The motor controller is capable of 60 A on both outputs which is more than enough to drive our robot.



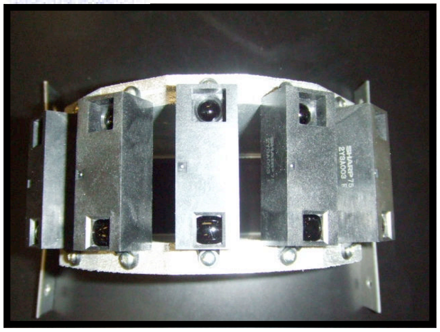
Motor Controller

IV. Software Design

In the past, our robot software has implemented white line avoidance, a strategy that is certain to fail when faced with a course laden with sand traps and dashed lines. We decided to implement a line following algorithm based on the Hough transform. In order to effectively use the Hough transform, preprocessing of camera images needs special attention. We decided to use the HSV color system exclusively to help eliminate problems with luminance associated with different weather conditions and time of day, and to allow us to work with hues directly, a task which is more difficult in the RGB system. We then divide the hues into “bins” of the Münsell system to simplify later processing. To detect lines, we first eliminate large regions of hues with the target bin and keep only those that fit into a threshold of line widths. Last, we use the actual Hough transform to detect line slopes, from which we infer which direction will keep the robot in bounds. We decided to use the Intel Open Computer Vision Library to save time developing common image processing algorithms and to boost the run time performance.

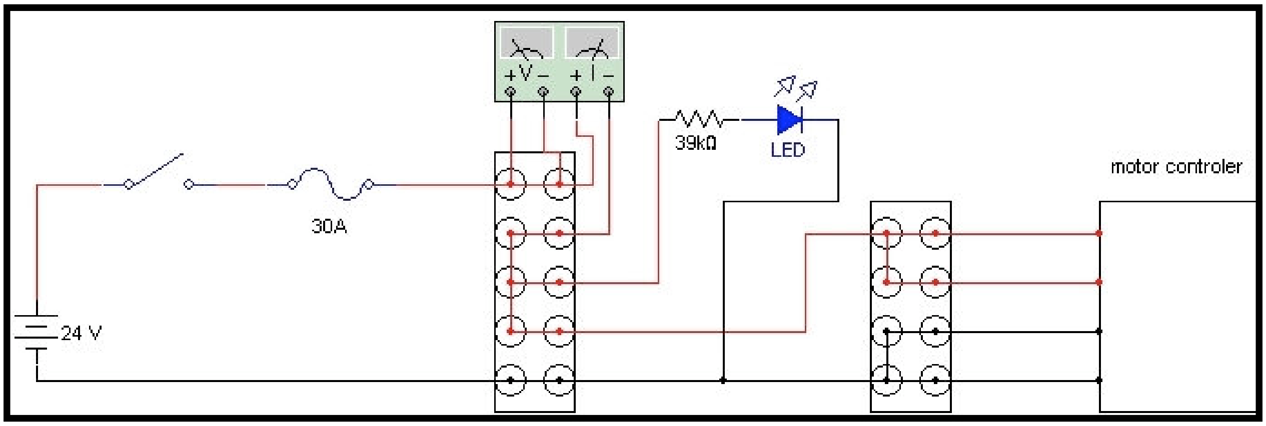
Our process of learning JAUS as a software team involved reading through the specification from the JAUS site and reviewing the implementation of past team’s code. We decided to use JAUS messages and components as the primary means of program organization. This will allow future teams to add or modify components without having to make program-wide changes to software; only the affected component code will need modification. To implement JAUS in a time-effective way with fewer bugs, we decided to use the OpenJAUS implementation of the standard as a framework for our architecture.

The object detection strategy involves scanning an array of six IR sensors connected to a custom made interface and combining the distance information with unnatural color detection from the camera. The interface to the IR array is controlled usin Atmel ATMega48 microcontroller mounted on a modified development board. The software on the chip uses a two-way protocol over a serial cable interface to configure which sensors are active and to relay distance information to the master program. The rear IR sensor is used to detect how much space the robot will have to back up if necessary. Simple object detection with the camera will identify unnatural traps. The process of integrating the output of the camera-based obstacle detection and the IR detection will give preference to the IR, zoning the area in front of the robot into 25 degree arcs.

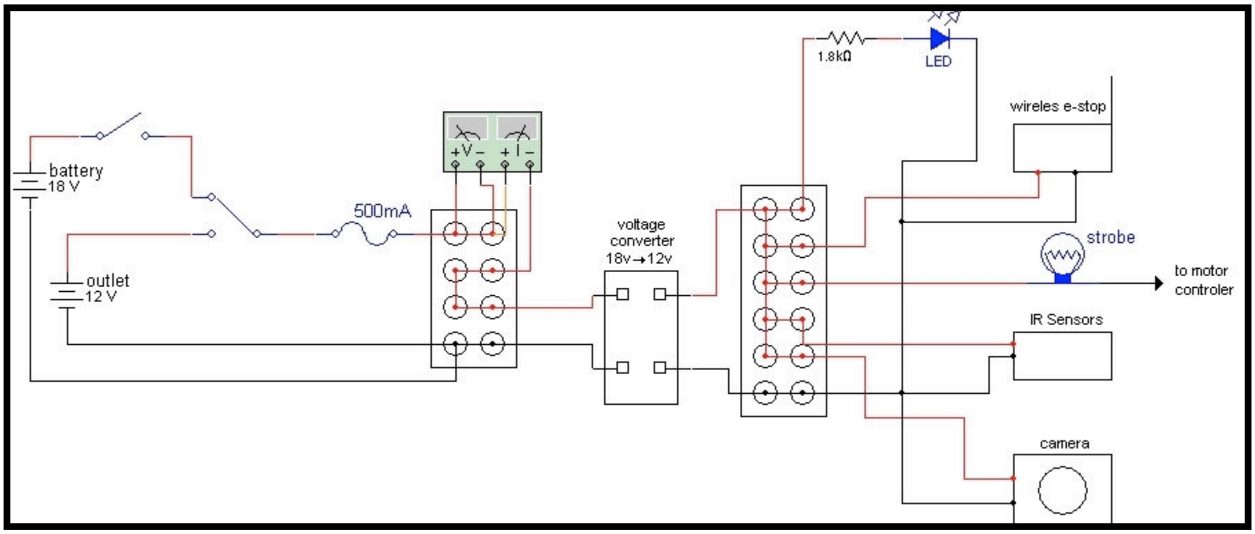


Infrared Sensors

V. Electronics Design



Drive Schematic



Electronics Schematic

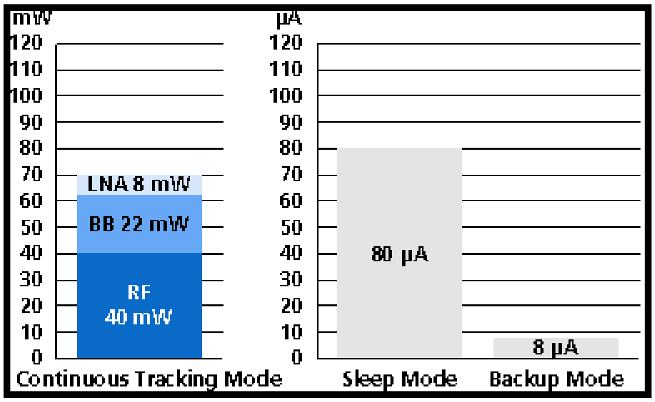
GPS

Since Cornelius is built on a $3,000 budget, we could not afford to spend a great amount of money on a differential GPS. Cornelius is equipped with a U-Blox Antaris 4 GPS. The U-Blox Antaris 4 provides greater accuracy than a standard GPS receiver but can be bought for a fraction of what a differential GPS would cost. The U-Blox Antaris 4 comes with features that allows us to pick up GPS signal within 1/2 second and is designed to be able to receive satellite signal in spots where it might be difficult with other receivers.

The U-blox Antaris 4 is also very efficient in its power usage. In the graph below, you can see that the GPS can be set to different modes that can save power. Sleep mode can be used with the GPS which uses 80uA and backup mode can be used which only uses 8uA.

Finally the U-blox Antaris 4 is more accurate than most GPS receivers. This GPS features a 16 channel receiver compared to the standard 12 channels found in many other receivers. More channels means that the GPS should be able to keep within its 2.5 meters of specified accuracy.

U-blox Logo

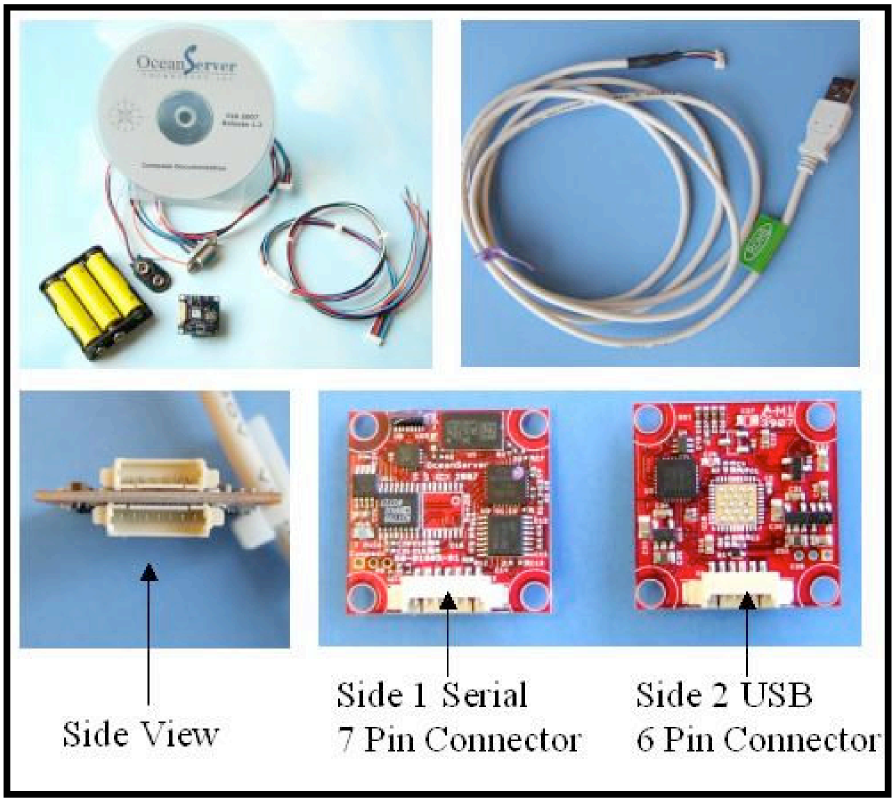


Power Consumption Graph

U-blox Antaris 4 Kit

Compass

Cornelius is equipped with an Ocean Server OS5000-US compass. The OS5000-US is a full three axis compass with tilt compensation. The compass uses Honeywell anisotropic magnetoresitive sensors to detect the earth's magnetic field. Solid state accelerometers from STM provide pitch and roll information. The data is sent to a 50 MIPS processor which determines bearing, pitch, and roll. Ocean Server claims the heading is accurate to 1 degree on level terrain. The OS5000-US can be accessed using either USB or serial connections. Power consumption is very low. The compass uses 20 mA at 5V when operating in RS-232 mode. The OS5000-US is one of the smallest three axis compasses in the world. The entire compass module is packaged on a single 1"x1" circuit board.



Compass

E-stop System

The emergency stop system on Cornelius is fairly straightforward. A keyless entry module was adapted for the wireless component of the emergency stop. The JC Whitney ZX478506T is the particular component selected. The manufacturer's specifications give this wireless module a range of 200 feet. The unit is powered with 12V. The relay outputs of the lock and unlock button were configured to allow either button to perform an emergency stop. A simple momentary pushbutton is located on the rear of Cornelius. These outputs feed directly into the motor controller to perform a hardware based emergency stop.

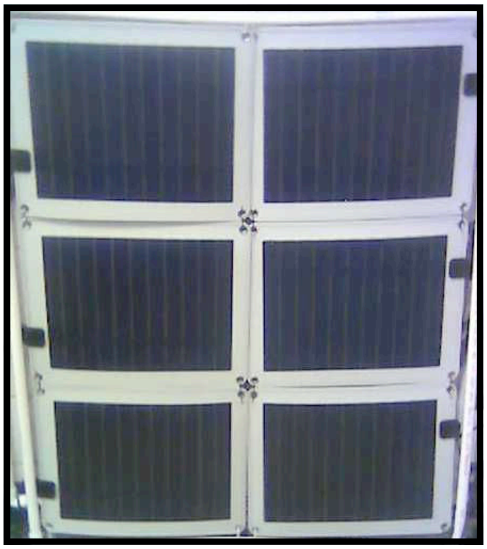
E-stop

Wireless E-stop

Solar Panel Charger

The Solar charging station consists of two groups of 3 Uni-Solar FLX-11 solar panels—one group for the 24v drive battery and the other group for the spare drive battery. Each group is connected together in parallel to increase the charging current. A charge regulator is used to connect the panels to the batteries in order to prevent overcharging and to keep the batteries at full capacity while plugged in.

The panels themselves are durable, flexible, and weather resistant—designed to be used in harsh conditions. Every panel contains bypass diodes to allow the panels to work when partially covered and blocking diodes to prevent the battery from discharging back into the panel. In ideal conditions, each panel outputs 12 volts, 10.3 watts of power and, typically, about 4 amp hours per day. With 3 panels connected in parallel, the charger can output 12 amp hours in a day.



Solar Panels

With more solar panels, our robot would have the potential to be completely independent of outside power and even completely autonomous—powering on at a given time, completing it’s task, self docking at the solar charging station, and powering down again—independent of any human interaction.

VI. Performance Estimates

Speed

Manufacturer claims speeds "up to 8 MPH"

350 Watt Motor: 14.5 Amps for 30 Minutes

Estimated Run time: 40 Minutes

Predicted Weight: 145 lbs

Climbing 15% Grade (8.53 degrees) at 5MPH

5 mph (2.23 m/s) 145 lb. robot. requires 21.5 lb. of force

including 4 lb. of friction.

Steering Motor

Max 174 RPM from motor

Under 20 in/lb max 172 RPM

Geared with a 48.5:1 ratio

Output under load becomes 3.5 rev/min = .0591 rev/sec

.0591 rev/sec = 21.28˚/sec

Lock-to-Lock is approximately 100˚

Time it takes to turn 100˚ = .2778 sec

So the fastest our steering motor will be able to turn directions will be .2778 sec which is more than adequate for our robot.

The max current: 10.05A at 20 in/lb

Steering torque should stay under 60 in/lb

60 in/lb max current = 14.52A

*This is a very rough estimate.

IR

Manufacturer data on our IR sensors gives 3 meters maximum range, with between 20 and 25 ms total response time. Response time will vary with the number of beams scanned, with a total of about 0.5 seconds to scan all 30 beams.

Compass

The manual says the compass has accuracy to within 1 degree of the heading in an undisturbed magnetic environment. If we mount the compass greater than 11 inches from ferrous material we should have plus or minus 0.2 degrees error in our reading. So the heading indicated by the compass should be within 1.2 degrees of our actual heading. Add some play for other errors and it should be accurate to 2 degrees of our true heading. We can access the data up to 40Hz.

GPS

Position 2.5m CEP 5.0m SEP

CEP = Circular Error Probability: The radius of a horizontal circle, centered at the antenna's true position, containing 50% of the fixes.

SEP = Spherical Error Probability. The radius of the sphere, centered at the true position, contains 50% of the fixes.

VII. Bill of Materials

Description	Details	Price	Cost to team
Razor Dirt Quad Electric ATV	http://www.razor-help.com/dirtQuad.html	$300	$300
Motor controller	AX3500 RoboTeq	$395	$395
Steering motor	NPC-41250	$155	$155
Keyless entry	JC Whitney ZX478506T	$30	$30
GPS	UBlox AEK-4H	$199	$199
Compass	Ocean Server OS5000-US	$299	$299
Computer	IBM ThinkPad R31 (used)	$350	$0
1394/USB PCMCIA card		$20	$0
Solar battery charger	Brunton SolarController	$49.94	$49.94
Solar panels	Unisolar FLX-11	$900	$0
Thinkpad external charger		$40	$40
Thinkpad batteries		$130	$130
Strobe	Action Electronics HAA110W	$11	$11
Potentiometer	P3 America R23P-RCWT	$20	$20
USB to Serial	Edgeport4	$260	$0
Rear encoder	Model 225q www.encoderoutlet.com	$252	$252
18V battery	Ryobi One+ batteries, charger, flashlight	$100	$100
Steering pot	R23P-RCWT	$20	$20
Sheet metal	Aluminum 1/8"	$100	$0
Motor mount	Aluminum 3/8"	$200	$30
Misc hardware		$100	$100
IR SENSOR	Sharp GP2Y3A003K0F x6 at $52.50 each	$315	$315
IR interface	AVR board and misc parts	$50	$50
DC/DC converters	TRCElectronics SD-15A-5 and SD-15A-12	$40	$40
Panel Meters	Futurlec.com	$40	$40
Misc elect		$100	$100
Spare Batteries	Powersonic PS-1270 12V 7.0 Ah (x2)	$40	$40
	TOTAL	**$4,476**	**$2,676**