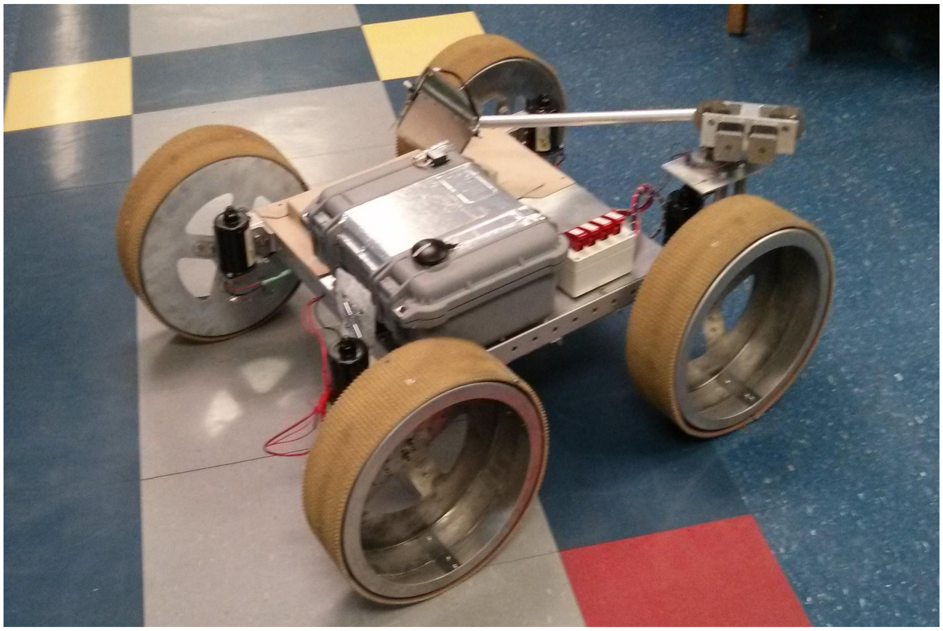
# UMass Lowell Rover Hawks
# Final Report
# 2013 RASC-AL Robo-Ops Competition



***Team Members***

*Undergraduate Students:*

Michael Antonov, Moran Assaf, Yaovi Ayeh, Alexander Chea, Adam DeFelice, Joshua Estrada, Alec Ferguson, Brianna Gainley, Limor Golan, Ryan Hart, Sean Harkins, Carlos Ibarra, David Jelley, Gregory Johnson, Michael Lunderville, Jheddy Melendez, Cameron Morris, Ian Ndicu, Alejandro Salido, Luan Tran and Bradley Wall

*Graduate Students:*

James Dalphond, Eric McCann, Brigit Schroeder and Nat Tuck

*Faculty Advisor:*

Professor Holly Yanco
Computer Science Department
University of Massachusetts Lowell
One University Avenue
Lowell, MA 01854
holly@cs.uml.edu

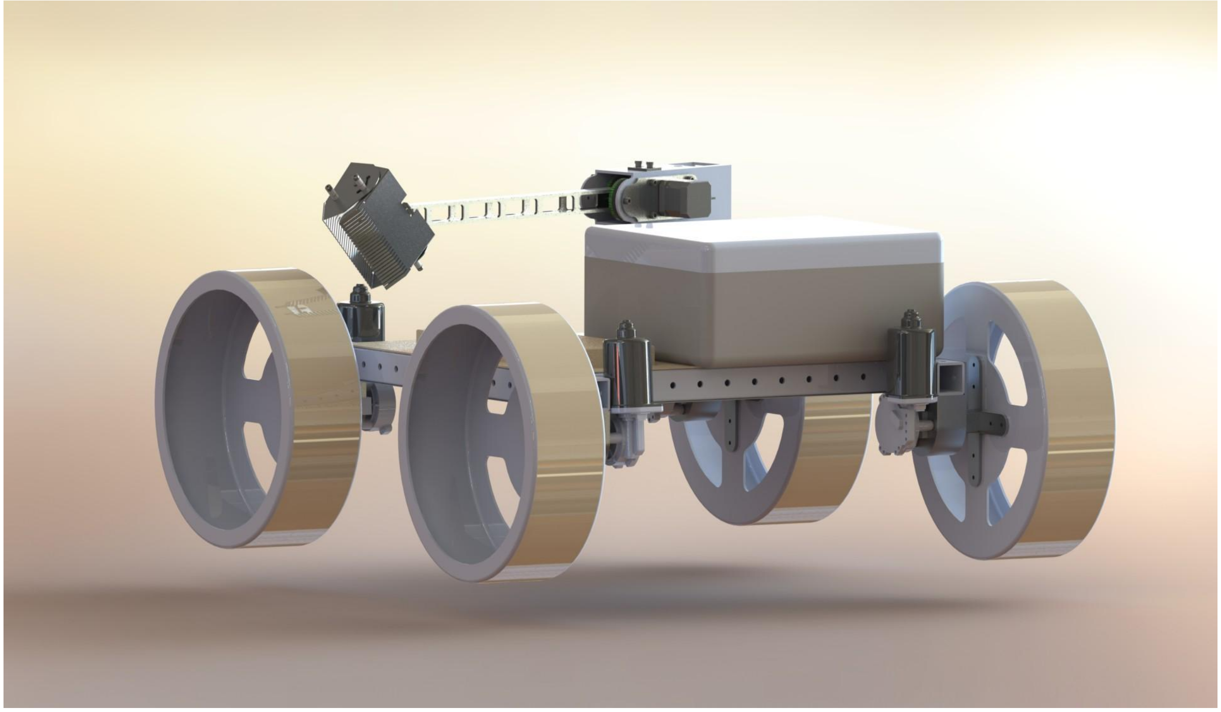
## 1. Introduction

Over the past four months, our team, the UMass Lowell Rover Hawks, has built and tested three rover designs. This document presents the final design.  It also describes the changes that were made to different subsystems between the different rover revisions.

## 2. System Description

Our rover's design is based on a four wheel, skid-steering system. Each of the four wheels is designed to extend past the frame of the robot, giving a much better angle of attack for obstacles such as rocks. Each wheel is independently and directly driven by its own motor. Though we experimented with different types of suspensions during our design process, the final version of the rover does not use a suspension system.

## 3. Technical Specification



**SolidWorks rendering of the Rover Hawk**

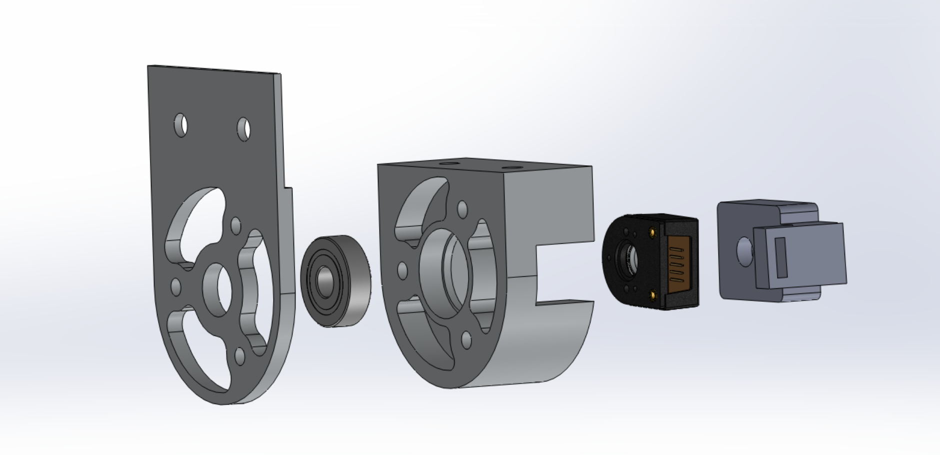
### 3.1 Chassis

The rover's chassis is all aluminum, built out of off the shelf extrusions by 80/20 Inc. The HT series square extrusions consist of precise holes placed every 1.5" (3.8 cm) on all faces. This material is lightweight, rigid, and is inexpensive. Using this material allowed us to test multiple chassis designs very quickly without relying on a machine shop. Many of the components used in the robot are shorter than the tubing itself, meaning that much of the robot's internals fit within the height of chassis. The dimensions of the rover are 32" (0.8 m) wide by 36" (0.9 m) long by 19" (0.5 m) tall. At the time of writing this document, the robot weighs 78 lbs (35.4 kg).

### 3.2 Drivetrain

The motor mounts on the robot were not initially designed with weight in mind. The forces acting on them from the skid steering made it such that these parts needed to be fairly strong. They already contained large voids in the material to house the encoders and bearings. The motor mounts were designed to match the aluminum extrusions used for the chassis, thus allowing us to make use of the ability to quickly reconfigure and test designs. The motors sit inside the chassis while the shaft extends through the encoder, mount, bearing, and bearing plate. Because we are directly driving the wheels and the shafts had to be longer to pass through the optical encoders, the motor mounts were designed to place a high load bearing close to the end of the shaft. The addition of this bearing significantly reduces wear on the internal gearing of the motors.

After the design was tested and proven to work well, we took a second look to see if we could reduce some weight. By cutting out some of the material around the shaft, and leaving ample material around the motors mounts we were able to remove ~0.2 lbs (~.1 kg) from each assembly, saving almost a pound (0.45 kg) on the final rover's weight.



**SolidWorks rendering of the motor mounts with bearing and optical encoder**

The rover's drivetrain consists of 4 - AME 218 windshield wiper motors. These motors are capable of spinning at 116 RPM (no load) at 8.2 foot-lbs of torque. With 13" (0.33m) diameter wheels, the rover is capable of traveling at over 1.7 meters per second (3.8

mph); however, we have limited the speed below that in software to 1 meter per second (2.2 mph).

The motors are controlled by two Roboteq MDC2230C controllers. The left front wheel and rear right wheel are on one controller, and the other wheels are connected to the other. The controllers are given identical commands, but each has its own encoders and separate closed control loop.

### 3.3 Wheels

The wheels for our rover have gone through several iterations over the past few months. Designing and fabricating wheels that were lightweight and strong, yet that would still provide adequate traction, turned out to be one of the most time consuming parts of the project.

In our first design, the wheels were constructed of two aluminum vent pipe caps held together by a PVC axle. While this design worked well, the axle needed to strengthen the faces of the pipe caps was heavy and difficult to assemble.  During testing, we also found that the connection of the motor to the wheel was not as solid as we wanted; we observed some wobbling.

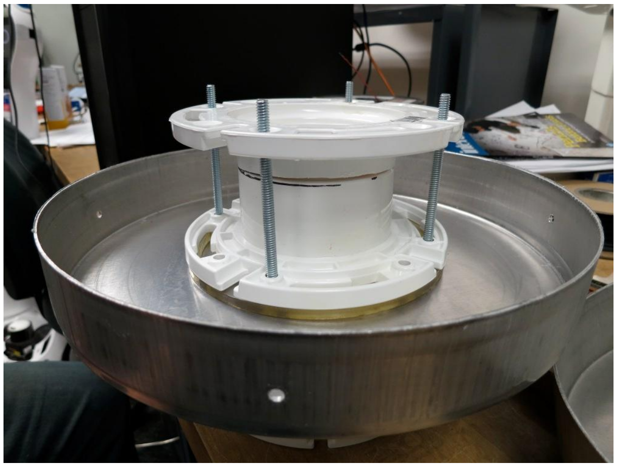


**Photo of the first wheel design, showing one side of the wheel (an aluminum vent pipe cap) with the PVC axle mounted**

The second design of the wheels consisted of two aluminum caps again; however, they were placed in a configuration in which they could be directly connected to each other without the need for an internal axle.  A second hub adapter was designed to spread the forces across the wheel, rather than concentrating it on a single point as in our first design.  While this design included this larger hub adapter, over time, the rover’s all-aluminum wheels would warp, making them only last a few hours of testing if obstacles were hit at higher speeds.

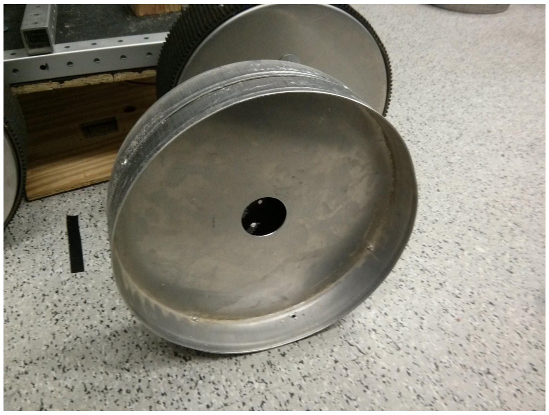


**Photo of the second wheel design**

For the third revision of the wheel design, we took a step back and looked at other options for their material. Aluminum was proving to be too weak. We decided to try steel, but the added weight made the design problem more difficult. We knew we could do an all steel design that could be welded, but this design would be heavy, even with weight reduction. We couldn't go all aluminum because the wheels would become more complicated in order to make them stronger. We decided to make the inner load bearing hubs out of steel and remove as much material as we could without sacrificing structural integrity. The outer wheel hubs, used simply to increase surface area, are not load bearing and were thus made out of aluminum. Again, we removed as much material from this part as possible to reduce weight while again keeping structural integrity in mind. The weight reduction led to the removal of approximately 0.8 pounds (0.36 kg) per wheel half, from the pipe caps.

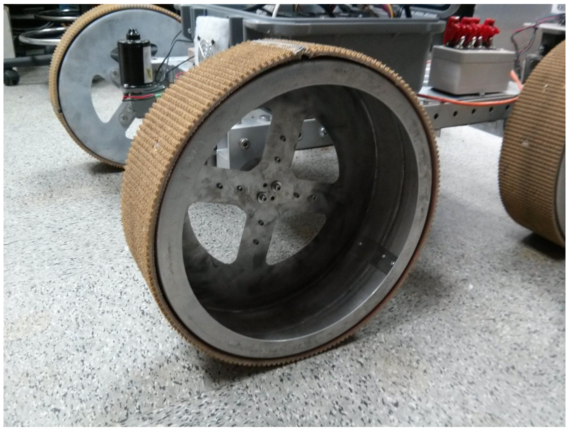


**Photo of the final wheel design**

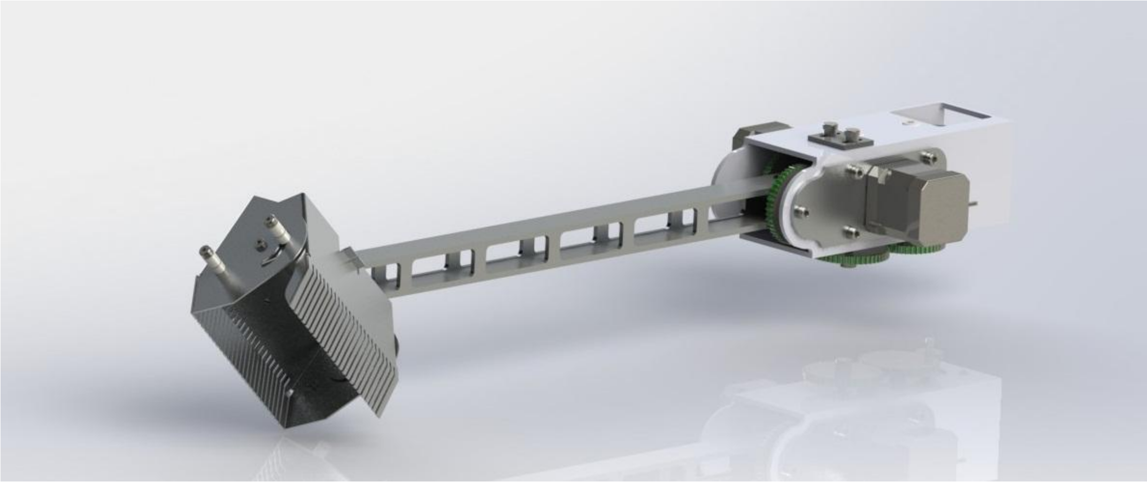
In the third design, the two halves of the wheels are joined by steel tabs riveted to both halves at 90 degrees from each other. They are held together even more by the treads that are riveted to both halves of the wheel independently from the other rivets. The treads for the rover are made from 4” (10.2 cm) wide conveyor belt material designed for inclined conveyor belts.

This third design weighs 5.6 pounds (2.54 kg) per wheel, which is less than the 6 pounds (2.7 kg) per wheel of the first design, while the steel used means this final wheel is three times stronger than that heavier first design.

### 3.4 Manipulator

The rover’s manipulator system consists of a 3 DOF arm and gripper. The arm is able to move 120 degrees up and down, and can rotate 300 degrees. Rotation is limited by the wiring of the stepper motors. The gripper can open from a closed position to a maximum of 4.5” (11.4 cm) wide, giving enough of an opening to pick up the largest rock on the field. Each degree of freedom is driven by a single stepper motor. Each motor is geared down to a ratio that maximizes speed while minimizing current draw.

The shoulder rotation is geared 3:1. The center of gravity of the arm assembly was designed to be the center of rotation of the shoulder. Because of this design decision, only slight gearing was required. The shoulder lifting mechanism is geared 9:1 due to the fact that it will be lifting a weight at a distance of 20 inches (0.51 m) from its center of rotation. The gripper actuation is cable driven by a stepper motor that is geared down 3:1. The gripper is normally closed using springs and is opened by pulling the cable. This design reduces the amount of power needed to use the gripper, as it will spend more time closed than open.



**SolidWorks model of the Rover Hawk’s manipulator**

### 3.5 Power

The power system is divided into two subsystems, logic and motor power. Both systems supply 12v power to their respective systems. The power system is built from three BT-70791A lithium-ion batteries. Each battery contains two 12v 17Ah cells, providing six total cells. Three cells power each of the subsystems and are wired in parallel. In this configuration, each subsystem has a 51Ah 12v power supply. The positive terminals are run through a 20A Schottky diode to ensure that the batteries are not charged either from each other or through the regenerative braking capability of the motor controllers.

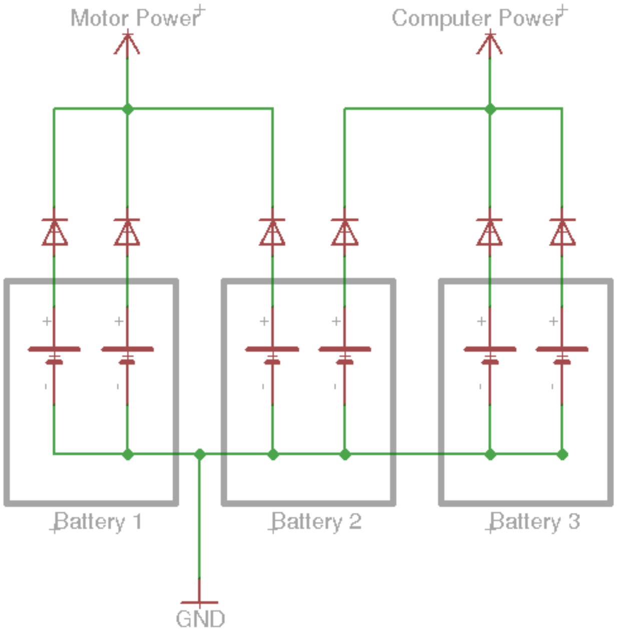


**Diagram of the rover's two power systems**

Because one battery is split between the two subsystems, and the cells are in parallel, we can remove one battery entirely from the rover. Removing one battery reduces the system's current sourcing capabilities but also reduces the overall weight of the robot by 3 lbs (1.36 kg). The table below shows the estimated run times for the robot in the worst case. For the drivetrain, the worst case is defined as constantly moving.  The worst case for the computer/arm subsystem is defined as constantly moving the arm and gripper with the computer running at maximum load. With a battery removed, the system is still capable of running well over the 1 hour minimum.

| Subsystem | Max Load | Runtime (2 batteries) | Runtime (3 batteries) |
|---|---|---|---|
| Computer/Arm | 12A | ~2.8 hours | ~4.25 hours |
| Drive System | 5A | ~6 hours | ~10 hours |

### 3.6 Computer

The computer that we built for the rover was meant to maximize power while still keeping battery life in mind. The computer consists of a Mini-ITX desktop motherboard with a 3.5GHz Quad-Core i7 that draws 95w. It contains 16GB of RAM and a 128GB solid-state drive. The computer is capable of running our rock detection system on multiple cameras and all of the video compression needed to pass the video over the 4G network.

The computer is housed inside a Pelican case where it is safe from the environment that it will be operating in. The Pelican case was coated in a metallic material that will reflect sunlight helping to keep operating temperatures lower. To cool the computer, we have two filtered fans set up to pull air directly across the fins of the CPU's heat sink. The fan on the front side is an induction fan, while the fan on the rear of the case is an exhaust fan.

The computer is powered using an automotive-grade computer power supply from Mini-Box. These power supplies were specially designed to run off of a battery setup with an input voltage between 12-24 volts.

### 3.7 Sensors

The robot was designed to contain several sensors that will make controlling the robot easier. Though we are not building a map, the GPS will give us an idea of where we are on the field by placing a point on a Google map where the robot currently believes it is. In addition to the GPS, we incorporated an IMU to give us a better idea of the pose of the robot, which direction it is facing, and its stance so the robot’s operator can tell if the rover are on a rock or an uneven surface.

In order to have closed loop speed control, we installed a high accuracy optical encoder on each motor’s output shaft. These encoders allow our motor controllers to operate in a PID loop that will keep the motors moving at the speed that they were commanded to go. Our motor controllers also allow us to see how much current is being drawn at any given time, allowing us to have a better idea of the rover's battery life.

### 3.8 Communications

The cellular device on board the robot is an AT&T 4G hotspot, with two omnidirectional antennas affixed to the robot. Rather than take a chance with possibly spotty Linux support for a USB dongle, or an even less supported mini-PCIe embedded module, we took advantage of the motherboard's wireless drivers, allowing us to prevent potential issues with compatibility and automatically connecting to the cellular network.

Our user interface and robot drivers communicate with each other using the ROS communications stack. The user interface uses Eric McCann's ROS# library, a component of which generates .NET 4.0 message handling code tested with the ROS TCP protocol on versions ranging from ROS Diamondback to ROS Fuerte.

As the user interface is implemented in the Windows Presentation Foundation using ROS#, which is an unofficial work in progress, certain design decisions were made for ROS# compatibility. ROS# does not yet support the Theora image transport, which uses MJPEG compression to transmit images with lower bandwidth utilization than the same images being sent using JPEG compression. To minimize bandwidth utilization over the cellular data connection, an intermediate server, co-located with the user interface, receives the MJPEG messages from the robot, and converts them to JPEG for consumption by the user interface.

Another such design decision relates to the life cycle of nodes in the C# user interface. The underlying callback queues and sockets of ROS nodes in ROS# require forcibly preventing garbage collection to allow them to function properly at runtime. The deinitialization process is thus hit or miss. To prevent the need to deinitialize and reinitialize the image receivers in the user interface (UI), disabling of unselected camera streams is accomplished on the robot by inserting a selective republisher, or multiplexer, node between the cameras and the UI. The UI tells this node from which of the multiple cameras to send images, allowing all of the user interface's image topics to exist for the life of the window, without bandwidth implications.

### 3.9 User Interface

Our user interface is implemented in C# (.NET 4.0). The actual window layout is handled by the Microsoft Windows Presentation Foundation. ROS# handles bidirectional communication with the robot. ROS topics form the robot directly affect change in the window, and input in the window is turned into messages to be published to the robot.

The UI team's first focus was aiding the driver's completion of the exploration and collection task. Consequently, information included in the user interface is either relevant to navigation or tasks. As the driver's visual focus is limited to one element on the screen, we elected to reduce bandwidth used by camera images by limiting the number of simultaneous feeds displayable on the UI to two. The two dedicated regions in which images are displayed also serve as the means by which the driver selects the cameras from which to display images.

The rock detectors running on the robot, described below, are detecting rocks on the video streams from all of the cameras, regardless of whether or not the driver is looking at them. To take advantage of this vision processing (described below), mechanisms have been added to alert the driver that he or she may want to switch one of their camera regions to show the camera on which a rock has been detected.

In addition to selectable camera feeds from the robot, user-controllable tools such as a "rock counter" and stopwatch are also available to aid the driver in estimating progress. Additional tools at the driver's disposal include linear and rotational velocity scalar sliders so the driver can tune the robot's maximum speed to be safe for specific driving conditions, camera parameter tuning sliders, and a button with which to observe and change the robot's motor controllers' emergency stop status.

We have also added the Automatic Direction Reversal (ADR) feature. ADR mode essentially inverts the joystick's axes, such that the robot can be driven backwards as if it were driving forwards. As both the forward, backward, and auxiliary cameras are all selectable in both the primary and secondary camera areas, it made sense to tie ADR mode to the state in which the selected primary camera is the rear camera.

### 3.10 Cameras, Mast and Vision System

The Rover’s vision system is comprised of forward- and rear-facing Logitech C910 Pro webcams, which are used for navigation, rock detection, and arm control. In order to get a better view from our cameras, a camera mast was designed to lift the cameras above the 0.5 meter (19.6”) height limit. The mast is built out of a Dynamixel RX28 servo motor attached to a fiberglass rod. The front and rear facing cameras attached to the mast can then be lifted an additional half meter for a better view.

The camera drivers are based upon ROS's uvc_camera package, wrapping video4linux drivers in ROS boilerplate. The version on the robot has been modified to allow runtime control of the camera settings (e.g. brightness, contrast, exposure, etc.) in a fashion conducive to control from a ROS# user interface.  Given the variation of outdoor lighting conditions the robot might encounter, being able change camera settings on the fly from the UI can be essential for detection to function properly. For example, large ambient light changes affect the amount of light backscatter into a camera's lens, increasing the possibility of glare and washout in an image. Being able to dynamically change the exposure and white balance helps to mitigate these effects. The cameras were also fit with polarized filters to reduce the effects of glare created by bright sun.

The rock detection algorithm uses both color and shape information for detection. Input images are converted to the HSV color space, and filtered using a set of thresholds determined by a color calibration routine.   The image data from the hue (H) channel is a value that represents a unique shade in the "true" color spectrum and the saturation (S) channel is a measure of how pure or saturated the color is. Both of these values are light invariant, which makes them ideal for outdoor color detection.  Once an initial set of blobs is detected from color filtering, shape descriptors are used to filter out any blob detections that do meet the general appearance of a round to oblong rock. This step is especially helpful in reducing the number of false detections.

The rover's rock detection system is persistent, active on all cameras even when they are not being streamed to the interface. If detections are found in an unstreamed camera's image, a notification is sent to the interface to alert the driver of potential targets for acquisition. When detected in a visible stream, rocks are enclosed by a similarly colored bounding box in the camera streams chosen by the user. To achieve a reasonable balance between hardware constraints, image processing performance and optimal image streaming rate, a 480p video resolution with 16:9 aspect ratio was found to best meet all these needs.

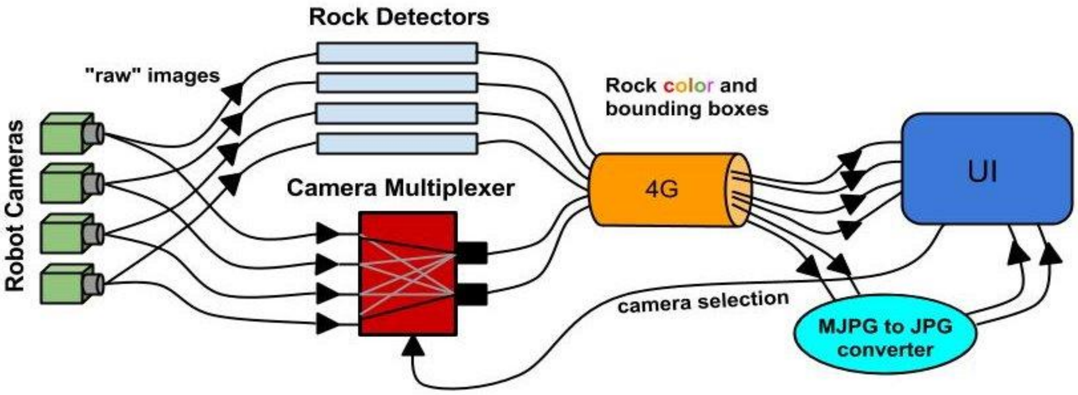


**Diagram of the vision, camera, communication and UI systems**

## 4. Testing Strategy

Over the course of the project, we have been thoroughly testing the capabilities of our rover's design on multiple terrains and obstacles. Using UMass Lowell's New England Robotics Validation and Experimentation (NERVE) Center, we have been able to test the rover on several terrains in a controlled environment, including 30 degree inclines, sand, gravel, and uneven terrains. We have also housed the Rover Hawk at the NERVE Center for driving practice for our operators, who controlled the rover from the robotics lab three miles away.

## 5. Overall Strategy for the Competition

Throughout our design process, we have been very aware that there is a strategic advantage to be gained by having a lightweight rover. All of our design decisions, particularly on the final version of the rover, were made with this fact in mind. We worked to cut weight from the rover, as described throughout the report above. Being light allows a team to go later in the day, and going later permits watching the other teams' live streams for sample position reconnaissance.

During the competition, our plan is to maximize rock collection points and hopefully find the alien life form. As both are solely dependent on the objectives being seen, our user interface and vision system have been designed to assist us with this task.

## 6. Budget

| | |
|---|---|
| Rover parts, machining and computer: | $ 8,500 |
| Team t-shirts, pins and stickers: | $ 1,000 |
| Registration for competition: | $ 1,200 |
| Robot shipping (estimated): | $ 800 |
| Travel (estimated): | $ 5,000 |
| | |
| Total: | $16,500 |

These costs have been paid with the $10,000 grant as well as Yanco's discretionary funds.

In addition to the funds we spent, we had four companies provide sponsorship through the donation or loan of equipment. Harvest Automation donated 4 batteries worth $300 each, for a total donation $1200.  Ideas Inc. donated the fabrication of the wheel halves for our final wheel design, a value of $400.  Scanning Devices loaned us a battery charger worth $1,000.  Dassault Systemes donated 10 SolidWorks seats.

**7. Public Outreach**

We have demonstrated our rover and discussed its development at several public outreach events:

- In February, we showed a very early prototype of the rover at the opening of the New England Robotics Validation and Experimentation (NERVE) Center, an event with over 200 attendees, including executives from over 25 robotics companies and academics from many local universities.

- In April, we demonstrated the Rover Hawk as well as its vision system at an open house at NERVE as part of National Robotics week.  This weekend event drew over 120 children, parents, educators and roboticists.

- In May, we exhibited our rover at the New England Botfest, a robot science fair held in conjunction with the New England Botball competition.  Over 300 people attended the event; over 50% of the attendees were students in middle and high school.

- Also in May, we presented the rover to the Boston and New Hampshire Chapters of the IEEE Robotics and Automation Society during their meeting held at the NERVE Center.

Through these events as well as lab tours to visitors, including open houses for admitted students, over the past five months, we have presented our rover to over 1,000 people in person.  We also have introduced over 1,500 people to our rover and the RASC-AL Robo-Ops competition through our mid-project review video, which was featured on IEEE Spectrum's Automaton Blog on their Video Friday post on March 22.

In addition to these outreach efforts that resulted in us reaching over 2,500 people, Prof. Yanco spoke about our participation in the competition when she was interviewed for NPR's Morning Edition on WBUR in Boston in February.  We have also been blogging our progress on the rover development at http://roverhawks.blogspot.com and have a Facebook page, /UMLroverhawks.