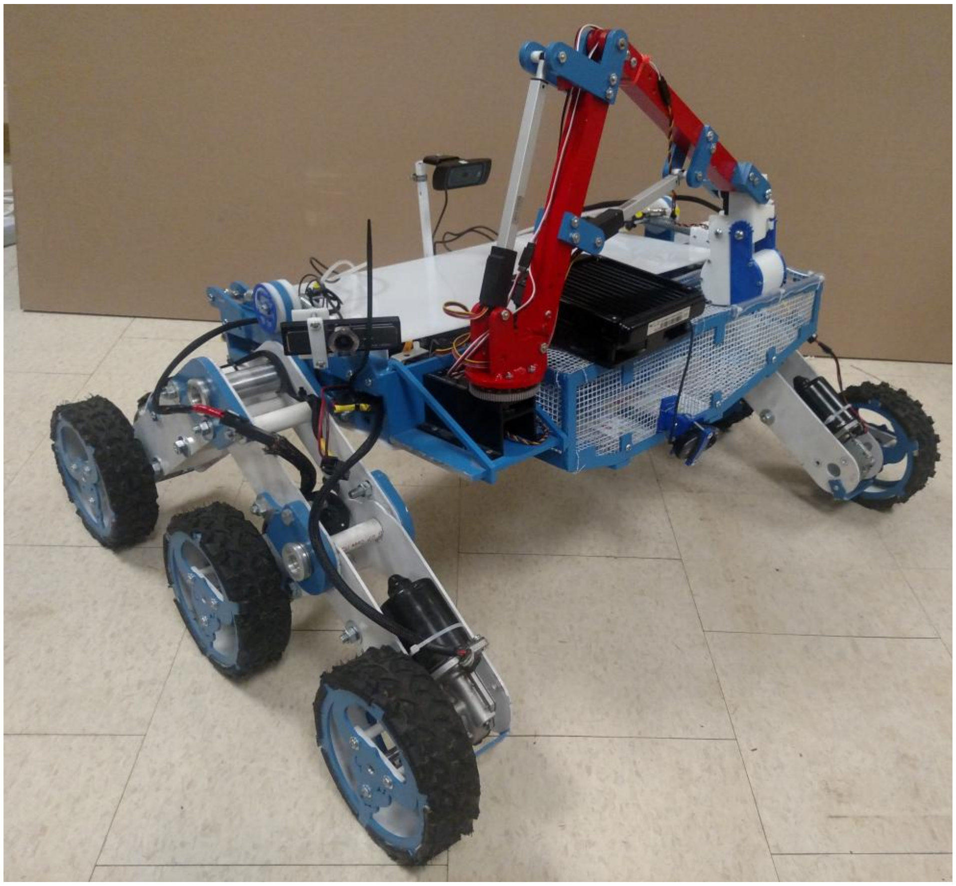
2014 - 2015 RASC-AL EXPLORATION ROBO-OPS COMPETITION

Final Technical Report

University at Buffalo, *The State University of New York*



Faculty Advisor:
Dr. Kevin Burke
Department of Electrical Engineering

Team Lead:
Peter Casey

Systems Leads:
Livio Forte
Mitchell Rathbun
Kaustubh Vartak
Tomasz Pietruszka

Team Members:
Akash Deshpande
Deep Parekh
Shishir Ambastha
Smriti Jha
Tao Wei
Vinayak Karuppasamy
Vishesh Seshadri
Katie James
Akshay Gupta
Andrew Frauens
Hao Li
Michael Bill
Mustafa Celebi

University at Buffalo
The State University of New York

1 Introduction

The University at Buffalo Space Bulls Robo-Ops team has redesigned a new planetary rover prototype for competition at the NASA Johnson Space Center. This year's team includes 4 veteran members from the 2014 competition which in combination with a breadth of engineers and scientists have completely redesigned, built and tested a new rover. The rover, Astraeus II, is made of a single chassis section with a unique rocker bogie suspension and 6 custom bull wheels powered by 6 independent DC motors. The suspension has removed the standard bulky differential arm and has utilized hydraulics to both mimic the differential motion and dampen the vibrational response. This design will increase the agility of the rover while maintaining stability while climbing steep slopes and the various terrain conditions. A linear actuator driven 2 segment arm with an end effector will be used for rock retrieval with the ability to move the wrist while encasing the captured sample. A front mounted, partially covered rock collection chamber will be used for holding the samples and keep samples from bouncing out. 4 high definition, wide angle webcams will be used with manual zoom capabilities for driver vision, blob detection and audio streaming. A closed loop control system will autonomously override the driver when necessary prior to a catastrophic event. The electrical system contains a high energy density battery with built in protection for a sustainable platform and lightweight performance.

2 System Description

2.1 Drive System

Capabilities of the Astraeus II are improved over last year's model. The newly designed rocker bogie inspired suspension gives us excellent ability to traverse varied terrain. Utilizing just one of last year's frame sections without a center pin, the rover can better manage uneven terrain and climb taller obstacles. The arm has also been redesigned and now makes use of linear actuators. The arm has more reach and can retrieve samples from a large area with ease.

2.1.1 Chassis

The rover utilizes a modified rocker-bogie suspension system mounted on one of the two frame sections from last years rover. It was determined that one frame would be sufficient for all of the electrical and communication components, but an additional rock collection area would have to be added for the arm and retrieved samples. By reusing one section from last year's rover two rovers were created; one of which would be used for prototyping and the early stages of development, while the other would be for the final competition rover.

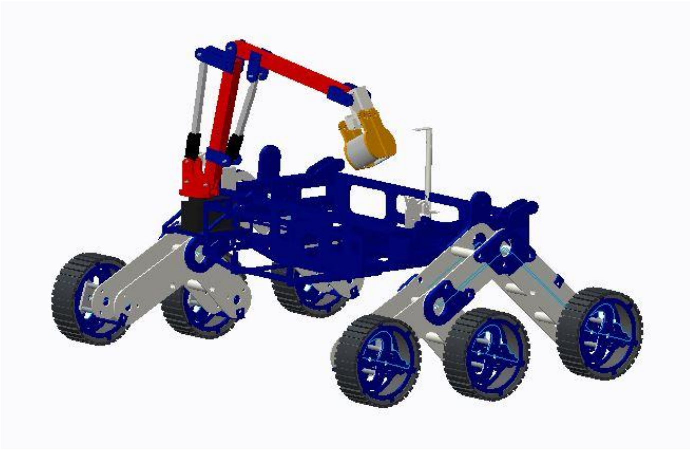


Figure 1- 3D Rover Rendering

The frame is constructed out of 6061-T6 aluminum, which was selected for its high strength to weight ratio and ease of machinability. This allowed modifications to the frame with the components necessary to mount the rocker-bogie suspension and a new rock collection area. The new size of the frame measures 18in x 24in x 5in (0.46m x 0.61m x 0.13m) and weights 9.26lb (4.20kg). The rock collection area was constructed using tig-welded lasercut aluminum plate, which allowed for rapid fabrication as all components were cut from prepared CAD files. The rock storage area is lined with hardware cloth with the intended purpose of allowing any sand to pass through. The top of the storage area is also partially covered in the same hardware cloth, which will allow for samples to be placed in at one end but make it difficult to fall out.

A modified rocker-bogie suspension is used to maximize traction by evenly distributing the rover's weight to all of the six wheels, minimizing sinking in soft and uneven ground. The rocker and bogie sections were designed with an angle of 110 degrees in order to fit within our maximum size specifications, while giving us the ability to traversing obstacles taller than the suspension height. Initially, the suspension was to be constructed of carbon fiber tubes connected together with aluminum joints. After much thought, It was deemed too costly and time intensive to machine, therefore, the team turned to laser cutting the silhouette of the suspension on to aluminum plates. Each side would be connected together, using aluminum tubing and fasteners, in strategic locations that would reduce both bending and torsional deflection. The rocker and bogie sections would be connected together, and to each side of the chassis, by the use of pins and bearings to ensure free motion. The rocker is limited to articulate ±20 degrees via the use of a Spring and Damper system (SMD) as shown below in Table 1.

In order to reduce vibrations, the SMD utilizes a dual action cylinder as both the damper and hydraulic differential arm. By changing the fluid inside the cylinder, the suspension behavior can be fine tuned. The hydraulic differential arm allows the suspension to remain rigid while climbing planes, yet able to articulate separately over uneven surfaces. This will allow the Astraeus II to climb obstacles up to 40 cm and planes as steep as 63 degrees [Table 1], which would prevent the catastrophe that had occurred with the Astraeus I.

Table 1: Suspension Articulation Table

Obstacle Size (cm)	Suspension Angle	Chassis Angle	Rover Angle
0.00	0.00	0.00	0.00
4.00	5.42	0.36	0.36
8.00	11.08	0.71	0.71
12.00	16.92	1.43	1.43
16.00	19.89	2.85	22.74
20.00	19.89	5.70	25.59
24.00	19.89	11.40	31.29
28.00	19.89	22.80	42.69
32.00	19.89	29.20	49.09
36.00	19.89	36.09	55.98
40.00	19.89	43.35	63.24

The limiting factor of the rated payload are the pins that mount the suspension assembly to the chassis. The FEA of the pin can be seen in figure 1.

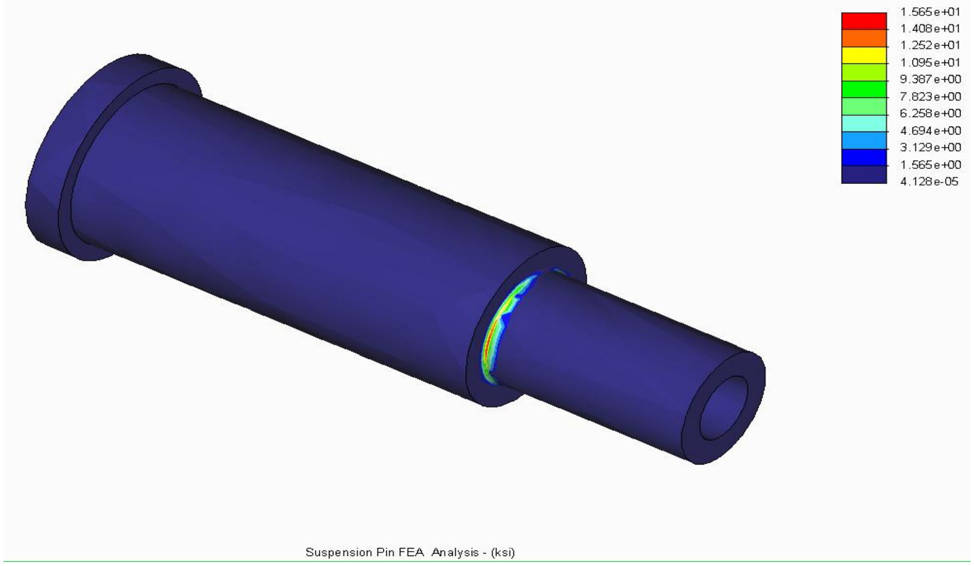


Figure 2 - FEA Suspension Pin

Propulsion is provided by six DC motors that are mounted inside of the suspension plates. The motors provide enough torque to allow the rover to move at a brisk pace, and traverse large obstacles. The rover utilized differential style steering, more commonly referred to as tank style steering. This configuration was chosen due to the simplicity of the mechanical and controls design, as well as the ability for the rover to have a zero turn radius. Additionally, differential steering lends itself to a lightweight design, as there are no motors or other hardware required to provide the axis and movement for each wheel's rotation.

The wheels have been designed to be lightweight but strong enough to handle the impulse loading seen when the motors rapidly change direction when trying to make a turn. Due to the physical limitations of the rocker bogie, and the center-to-center distance of the motors, we went with a 6in wheel. Two Space Bull hub plates sandwich a 2 inch wide section of PVC pipe through corresponding keys and keyseats located equidistantly around the edges of the pipe. In order to resist shearing from the impulse loading, steel keyed plates are attached to the inner surfaces of the aluminum hub plates. This was all bolted together, through corresponding holes in the Space Bull hub and keyed plates, with spacers to prevent shifting of the components. The tread utilized is standard mountain bike tire, which provides substantial traction on any terrain. The tread is secured to the PVC pipe with epoxy and friction fit between the aluminum plates.

2.1.2 Sensory System

The rover uses four cameras to navigate and survey the field. Two Logitech C310 USB webcams serve as driving cameras. One camera sits in the front of the chassis for an unobstructed view of the ground and the other one sits at a height on a small tower a few centimeters behind the first camera. As a wide angle camera, it allows for a larger field of view thus enabling the operator to see a wider section of the terrain. In addition to the navigation cameras, we employ two Genius WideCam F100s to aid rock sample retrieval in regions out of the main camera's field of view. The cameras are screwed onto either sides of the rover for an investigation of the field for rocks. The F100 allows manual focus and 120-degree ultra wide angle view, both of which are advantageous for blob-detection. The C310 at the front aids the positioning of the arm over the rocks once they have been found.

2.1.3 Power System

Astraeus II is powered by a 25.9V lithium-ion polymer battery with a peak output of 30A and a 21Ah capacity. Precautions have been taken with a closed loop sensor system to be implemented into the rover to keep within the limits of the power source while maintaining overall system performance. PicoPSU is a compact power supply with built in voltage regulars used to power and protect the onboard computer and all additional peripherals.

2.2 Sample Retrieval System

After last year's arm had trouble lifting a payload at full extension, the main goal of this year's design was to create an arm that would be able to lift the heaviest rock at full extension. The arm was first designed to be machined from aluminum. Although it is heavier than the plastic used for last year's arm we were confident that utilizing the mechanical advantages of a linear actuator we could achieve a fluid motion.

We began the analysis of the arm with the same length as last years arm, but quickly realized that we would be able to achieve a longer length. After multiple iterations and testing using ANSYS we achieved an overall length of 22 inches. This length still gives us a factor of safety of 2, but it was determined to be the optimal length because it still allows us to reach close to the rover. Like last year's arm we have two linkages and 4 degrees of freedom. This allows our arm to rotate 180 degrees and reach a maximum of 18 inches out, we are also able to reach 1 inch away from the front of the rover. This is suitable because it allows 1 camera to cover the entire useable rock collection area.

The manipulator is a 2 semi-cylinder scoop that closes around rock samples. The manipulator itself and the wrist mount to the arm are 3D printed and attached together with screws. The 3D printing allowed for small changes to be made quickly, adapting the design to accommodate more varied rock samples and to fit the arm properly. Thin aluminum flashing is fixed to the bottom of the scoop to improve retrieving samples off smooth surfaces. The open and closing of the scoop halves and the tilt of wrist gives the manipulator subassembly 2 degrees of freedom. The subassembly weighs X lbs and measures roughly Y by Z inches. The manipulator is designed to accommodate the varying rock sizes found in the rock yard. The dimensions of the manipulator can hold a rock sample that is 58mm by 81mm in the closed position. The high torque servo that controls the closing action has 611 oz-in of torque, to clamp on to irregular shaped samples or samples larger than the manipulator's closed volume.

3 Control and Communication Systems

3.1 Control and Communication Systems Overview

Two computers, one on the rover and one at our home base, were used in the communication system. The onboard computer runs Ubuntu on an AMD quad-core processor. The home computer also runs Ubuntu with a quad-core AMD processor. A graphical interface was displayed with a main window for the camera and vision system. We also included a display for the servo and actuator values being sent to the rover, a display for the temperature and sensor values being sent back from the rover, and multiple controls that allowed the user to select which camera to display. The FPS, resolution, and screen width of the camera can also be controlled. Furthermore, the User Interface (UI) was responsible for getting commands from Xbox controllers and sending them over a TCP connection to the rover computer. Once the commands were received by the rover, they were parsed and sent to either the camera system, or the control system that consisted of our drive motors and arm. The rover computer sent data back to the home computer over the same TCP connection, data which was then displayed on the UI.

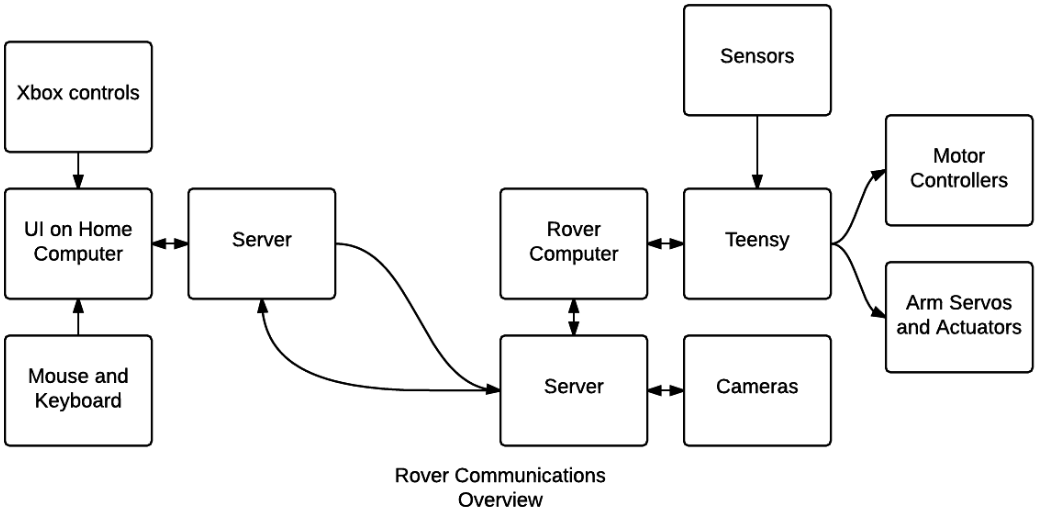


Figure 3 - Communication System

3.2 Control System

The rover was controlled by two XBOX 360 controllers, as opposed to the single controller used for the competition last year. This decision was made to increase the overall control we had over the rover. For the drive motors it was decided that having each joystick mapped to the opposing sides of the rover would offer analog control while being intuitive and user friendly. Also, having a full controller dedicated to the arm allowed for each axis of motion to be represented by a single joystick or button combination. This simplified the arm's control process and allowed for greater precision when picking up rocks.

As for the implementation, the controls were picked up through the UI. This was done by running a Python script that made use of the Pygame libraries to update the input value of each button and axis of the controllers. These values were then sent to the rover using our communication system.

The rover computer made use of ROS, or Robot Operating System, which is a framework that offers libraries and tools to simplify the process of controlling robotic systems. Doing so made communication between the various hardware we had to interface with efficient and modular, as code is broken down into separate nodes. A node in ROS is basically an executable file connected to ROS. Communicating between nodes is made simple by ROS through the use of topics. Each node has the ability to publish or subscribe to a given topic. We took full advantage of this pattern by running multiple nodes concurrently and relying on topics for communication between different areas of the code.

When the command string reached the rover computer, it was published to a topic. A single node subscribed to this topic and parsed each string using python's built in string libraries. After determining the proper destination of the string, it was published to the corresponding topic. Each of these topics were in turn subscribed to by a node dedicated to a specific component of the rover's control system. Strings either represented a control command, a camera command, or a reset command. Below is a diagram of all the nodes and topics working together.

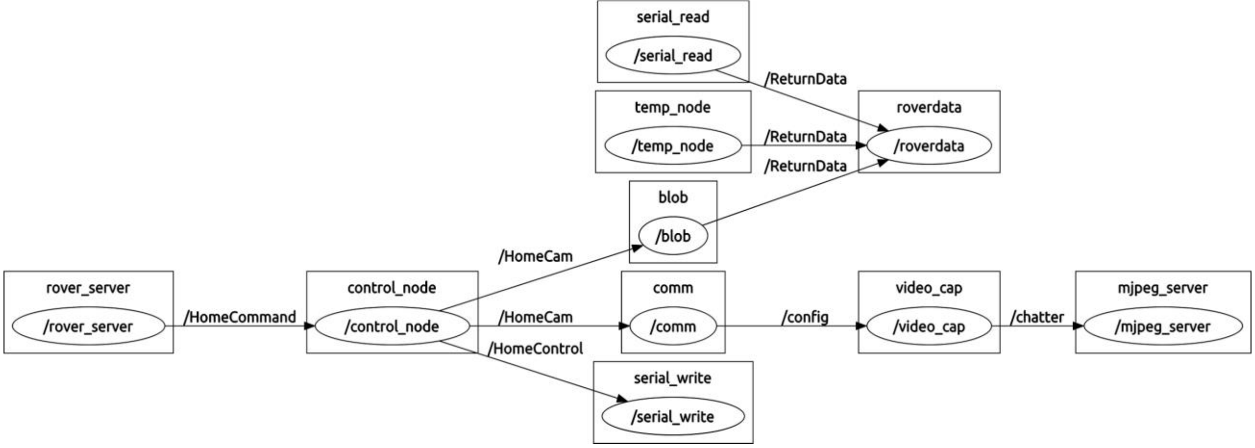


Figure 4 - Communication Paths

The control commands simply pass a list of values to be written to the arm and motors. A node is dedicated to send this string along to the onboard teensy. This is done with serial communication and the computer serial port. The teensy then parses the string and writes to the corresponding servos and actuators. The modularity of our control code allows for much easier debugging and problem detection, along with easy modification of movement speeds.

The camera commands consisted of four separate values, with one for the specific camera we wanted feedback from, and one for FPS, resolution, and screen width respectively.

The reset command was particularly difficult to implement. The purpose of such a button was to remotely restart the computer without having to shut down all of the systems. Doing so would reset all connections and initial values, and served as a method to prevent a shaky connection or signal from ending a run. This was implemented by having a button on the UI set a flag. Once this flag was detected by the rover computer, all of the ROS nodes were killed. In our launch file, all of the nodes were made to be respawnable. This meant that once they were killed, each node would restart, allowing for the reset button to have its desired effect.

The control scheme was also implemented locally, as was required for the competition. To do this, it was decided that the ROS joy libraries would be implemented as opposed to the Pygame libraries used in the long distance control. This was because the Pygame libraries could not be used since the rover computer was essentially operating in ‘headless’ mode and Pygame is a gaming library. The ROS joy libraries provided a unique challenge, as a node in the joy library can get commands from multiple controllers, but can only write to one topic. Since the two Xbox controllers were needed for different areas of control, it was eventually decided to run two joy nodes and map them to different topics.

3.3 Communications

Communication between the rover and the base computer is being done through TCP sockets. Every command and data value is made into a string and sent between the two computers. The camera and drive information received at the rover is then published to its respective ROS topics to switch between cameras and move the rover. We are also sending back feedback data from the rover, such as CPU temperature, which gets displayed in the GUI at the base computer. The CPU temperature is actually used as a sort of ping that allows the team to check the connection with the rover computer. This is done by simply concatenating the time to the end of temperature string that is being sent, thus displaying that time when it reaches the rover UI.

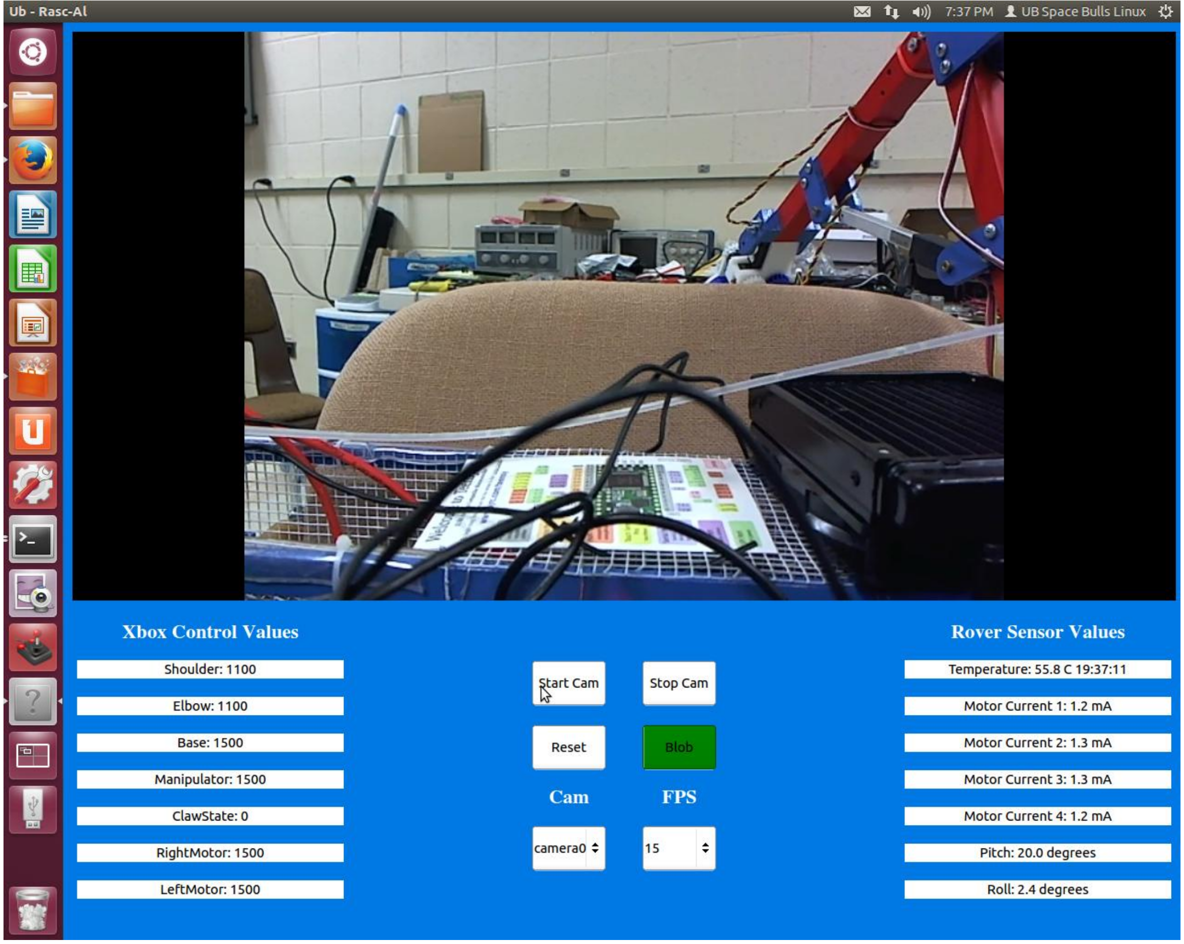
A Sierra Wireless AirLink GX440 router was chosen due to its compact design, I/O flexibility, and compatibility with Verizon’s 4G LTE network. The Sierra Wireless AirLink GX440 has a wide input voltage range and is powered directly from the battery. It is connected directly to one of two Ethernet ports on the onboard computer. The router is configured with a static IP address so that the base computer can connect to the hardcoded IP address. This connection is essential for long range communication and control.

3.4 User Interface

The UI was implemented using the PyQt libraries. While the visual design was fairly simple, the UI was responsible for multiple tasks. The most important was displaying the rover camera view on our base computer. This window was chosen to be fairly large, as it gave us a better view of the rover’s surroundings. It also helped with the precision needed when picking up rocks. The UI also displayed the values being written to the rover’s control system, which allowed for the user to have a better idea of the rover’s positioning when dealing with delay and/or a poor camera feed. Furthermore, values were read back from the rover, such as the temperature, which was displayed alongside the time that the string was sent.

The Xbox controllers were integrated with the UI to control the rover effectively. For this to occur successfully, our TCP communication code also had to be integrated so that we could seamlessly send commands to the rover. Command strings were also sent to select cameras along with given characteristics of the camera stream.

The blob detection was also implemented alongside the rest of the UI. This was done by reading an HTTP stream of the blob detection output and displaying it in a window when the driver presses the corresponding button. To signal the driver regarding the detection of blobs, the button would display as green if a blob was detected onboard the rover, and its normal color white if nothing was detected. We chose the aforementioned approach because it established blob detection as an aid to the driver and didn't preempt the main camera and controls when it wasn't needed.



Xbox Control Values

Shoulder: 1100
Elbow: 1100
Base: 1500
Manipulator: 1500
ClawState: 0
RightMotor: 1500
LeftMotor: 1500

Cam: camera0
FPS: 15

Rover Sensor Values

Temperature: 55.8 C 19:37:11
Motor Current 1: 1.2 mA
Motor Current 2: 1.3 mA
Motor Current 3: 1.3 mA
Motor Current 4: 1.2 mA
Pitch: 20.0 degrees
Roll: 2.4 degrees

Figure 5 - User Interface

3.5 Camera and Vision System

The four cameras are mounted randomly with integer IDs 0-3 when the rover starts. However, we need to know which ID corresponds to which camera. Last year, this problem was solved by re-configuring the four cameras when the rover was started. This year, we resolve this problem more elegantly by establishing soft links to the cameras via Linux udev rules.

The two Logitech C310 USB webcams are identified by their serial numbers. The two Genius WideCam F100 cameras have exactly the same serial numbers, so we identify them by the USB port numbers to which they are connected. The cameras are then remapped with integer IDs from 7 to 4. In this way, we can always tell that camera 7 is the first Logitech C310 USB camera, camera 6 is the second Logitech C310 USB camera, etc.

The video capture component captures image streams with the OpenCV library. It is also subscribed to a configuration topic, which allows us to switch between the four cameras and also adjust the frame rates for the video dynamically. The image streams are then prepared with a MJPEG server, compressed with H.264 encoding, and finally transferred to the home computer with the Gstreamer library.

The two cameras used for blob detection help the operator locate the colored rocks by means of image processing. The OpenCV library is used to analyze the streams from both images. First, the camera frames are converted to HSV color space. We then define hue, saturation and value ranges for each color. If a set of pixels belong within this range, a blob is formed around them. Each blob is represented by a bounding box which are color coded in order to be easy to distinguish visually. The streams from both cameras with the blobs are merged into a single image and published to a ROS topic. MJPEG Server streams the ROS topic images over HTTP, which can then be viewed in the UI at the control center.

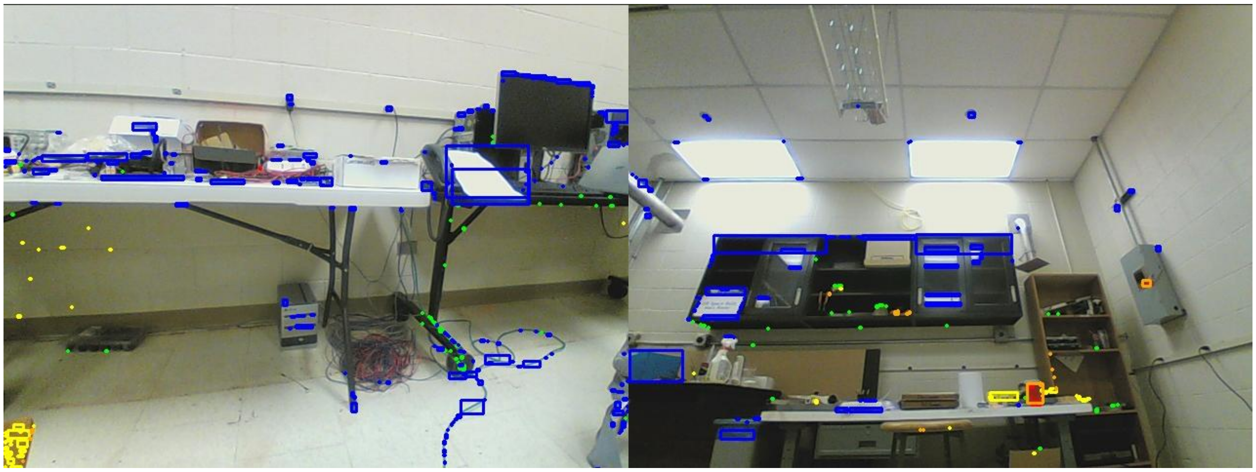


Figure 6 - Blob Detection

3.6 Closed loop control system

The control system is driven by a Teensy 3.1 micro-controller which is interfaced with the computer through a serial connection. The choice of controller provides ample room for processing sensors and controls due to the high speed of the controller (96 Mhz). The microcontroller provides ample I/O pins for communication with the several motor and actuator controllers. Communication with the Teensy occurs in a packet setup wherein the computer sends several groups of comma separated values which send set the speed for the drive motors as well as the positions for all the actuators and servos. Each packet has a checksum included with it for error correction. If an error is caught, the packet is simply ignored. The microcontroller then applies a low pass filter on all these values ensuring quick changes in the movements or position do not occur. This helps with any delayed or missing packets in the link to the Rover, and prevents undue stress on any component due to any sudden changes in the commanded speed/position.

The drive motors of the rover are driven by two RoboClaw 30A dual-channel controllers. The microcontroller communicates with the two controllers through a bidirectional packet serial system. This setup enables the microcontroller to set a precise speed. Additionally since this communication link is bidirectional, the microcontroller can query the individual motor current draw as well as various diagnostic information such as the controller temperature, and the supply voltage of the controller. This provides vital information about the state of the rover and its drive system which is then relayed back to the operator. The linear actuators on the arm are controlled through dedicated control boards. Like the servos in the wrist and claw, these control boards are controlled through microsecond timings. This enables the entire arm system to be run under one subsystem.

The rover this year has a focus on system monitoring and remote feedback. Data is continuously collected and relayed back to the base station to keep the operator informed about the state of the rover. An onboard accelerometer and gyroscope package will be utilized along with a complementary filter in order to gain the angle of the rover's chassis to the ground. All motor currents, as well as the voltages of the various voltage bars are also monitored. Temperature sensors placed in critical areas get the temperature of the rover. All these sensors contribute to a snapshot of the rover's health which is then relayed to the operator so that they may better respond to the rover's needs. Additionally, a fail fast and safe system is instituted such that if the controller determines that any of the values are over a safe value steps will be taken to mitigate a potential disaster. For example, if a motor current stays high for too long, the microcontroller will stop moving that motor to avoid overheating the controller, or burning out the coils.

4 Technical Specifications

The Astreus II more efficiently distributes stresses from the chassis to the suspension than its predecessor. This allows for the rover to handle a max payload of 90 kg with a safety factor of 2.24. The rover itself weighs 35.4 kg which is heavier than the Astreus I which only weighed 32 kg. Being powered at all six wheels via high torque DC motors, to take full advantage of the rocker bogie system affords us excellent traction and maximum speed of 0.64 meters per second. This maximum speed is governed in software to allow for precise control when climbing out of the craters. The rocker bogie suspension and the low center of gravity of the Astreus II allows the traversing capability of rocks as large as 40 cm or angles as high as 63 degrees. The run time for this rover have been tested to be over 3 hours.

5 Mission Control Center Operational Plan

Mission control center will have three primary drivers. One driver will be controlling the rover navigation while working in conjunction with the arm and manipulator driver. The third and final driver will be the supporting eyes and ears for the other two. This person is going to watch the blob detection cameras for samples and at the sensor system parameters that are streamed to home base. The supporting driver will also be the primary decision maker stating where to drive and what rocks to collect while making sure the drivers have ample time to traverse all terrains and return to the start point. Practice for these roles has taken place throughout the testing time to date. While at competition these roles will be followed as in competition to work out any problems that may arise. Having people with a narrow role allows for a highly trained person to operate single tasks well versus many tasks relatively poorly.

With a lot of test time the rover has been tweaked to maintain stability with slight errors in control and a closed loop control system has been implemented to prevent rover and arm drivers from going beyond limits unsafe for stability. Back up drivers for all three roles will be onsite to take the place of another driver in the case a driver has to be absent or be relieved of their duties.

6 Budget

In addition to the grant received for this project sponsorships and donations were sought out. In conjunction with these supports, undergraduates teamed together to work on research grants from the Center for Undergraduate Research and Creative Activities. The budget can be seen in the following tables.

Table 2- Competition Expenses

Competition Expenses		
Vendor	**Item**	**Cost**
NIA/NASSA	(4) Competition Registration Fees	$920.00
Entrprise	(7 day) Rental Van	$700.00
Courtyard by Marriott	(5 nights) Hotel Room	$570.00
Various	Gas and Tolls	$700.00
Al Ross	Team Shirts and Banner	$800.00
Total		***$3,690.00***

Table 3 - Material and Equipment Expenses

Material and Equipment Expenses		
Vendor	**Item**	**Cost**
Amazon	Electrical components, valves, hardware	$ 1,629.14
Robot Marketplace	(2) Motor Controllers	$ 258.80
Metal Supermarkets	Metal stock, rods, plates	$ 175.41
Firgelli	Linear Actuators and control boards	$ 529.34
Adafruit Industries	Electrical sensors and display	$ 100.95
Custom Laser	Laser cutting suspension	$ 602.85
Harbor Freight	Tools, drill bits, saw blades	$ 25.52
Lowes, Home Depot	Hardware, metal, springs	$ 455.89
McMaster-Carr	Specialty screws, bolts	$ 325.73
Target	Bike Tire Treads	$ 130.93
Servo City	Servo parts	$ 27.55
Total		***$4,262.11***

Table 4 - Sponsorship and Donations

Sponsorship and Donations		
Vendor	**Item**	**Cost**
CURCA Grant	Closed-loop Control System for Planetary Rover Design	$1,938.00
CURCA Grant	Kinematic Analysis of a Robotic Arm for Remote Control Applications	$1,716.24
AM Equipment	(10) 218 Series Gearhead Motors, 64mm, 24V	$928.60
3M	(12) Assorted color electrical tape	$79.08
Servo City	Discount on all purchases	$20.00
UB Facilities Plumbing Shop	10' x 6" Schedule 40 PVC	$92.40
Verizon Wireless	4G LTE Service Plan, 3 Routers	$1,568.00
National Instruments	myRio Development Board and LabVIEW License	$1,249.00
Merritt Machinery	Machine work and material	$500.00
Airline Hydraulics Corp	2 UDR-10-2 Air Cylinders	$66.22
Clippard	2 UDR-10-2 Air Cylinders	$66.22
Hitec	6 Servos - Arm	512.94
Firgelli Technologies	Discount on all purchases	$100.00
Sierra Wireless	2 Routers	$1,398.00
Mini-Box.com	(2) picoPSUs with AC power adapter	$158.90
AA Portable Power Corp	Discount on battery and charger pairs	$400.00
Total		***$10,793.60***

7 Public/Stakeholder Engagement

This year, the Space Bulls team has continued active use of a Facebook page to share updates and information to our followers throughout our progress. This avenue allowed the public to get updates from us as they became available. Posts were made frequently throughout the build to keep everyone informed with our progress.

The Buffalo branch of the American Association of University Women (AAUW) hosted its tenth annual Tech Savvy conference this past March. Being a national organization that advocate for equality and education for women and girls, this gave us a great audience for a broader impact in to the lives of many middle school girls to pursue careers in STEM. This year the team hosted a “Rover Races” workshop, which is a NASA classroom activity.

To show our gratitude to the engineering school, the team combined with the UB IEEE Student Branch for open house events throughout the semester. This allowed us to reach current and prospective students and parents for the ongoing work. We used this to show case a multidisciplinary project that included undergraduate research and collaboration with graduate students.

The team also collaborated with the UB IEEE Student Branch at the Buffalo Museum of Science as part of Engineers Week. Here we gave young children and adults the opportunity to use our newly designed arm to pick up candy. This was great for candy lovers and engineers alike who had a good test group for feedback on the ease of control.

Two groups of undergraduate researchers on the Space Bulls team were chosen for the poster presentation at the 2015 Celebration of Student Academic Excellence. This research done by mechanical and electrical engineers was funded by The Center for Undergraduate Research & Creative Activities (CURCA). This was an excellent opportunity for students to take part in research while gearing their interest in robotics alongside the rover project, and providing great presentation experience.

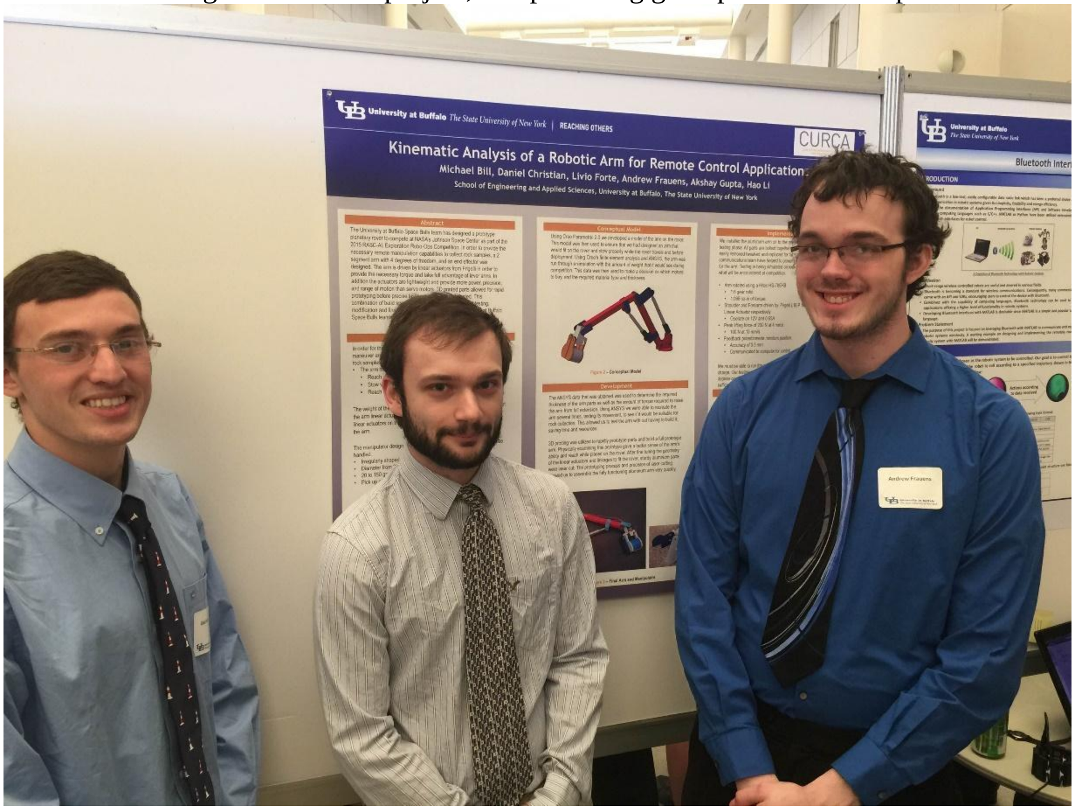


Figure 7 - CURCA Presentation

In connection with our sponsor, Verizon Wireless, the UB Libraries brought VGo, a teleconference robot with multiple educational applications, to campus. Part of the presentation, the team was able to discuss the creation of Astraeus II and were able to network with the UB Community regarding the project. The team was also able to discuss the networking configuration and hardware with some of their engineers to work on improving our current hardware.

This April a few Space Bulls representatives joined the UB President and Western New York legislative delegation as part of a celebration for all of the opportunities and achievements at UB. Students spoke with the delegation regarding the project and great opportunity it has provided them personally and as an engineer.

The Space Bulls combined with the School of Engineering to work with elementary school students discussing fundamentals of science. Seen in the following figure is one of the team members explaining circuits to a young mind.

Figure 8 - Science in Elementary Schools

8 Moving Forward

Moving forward the team will be testing and practicing driving. One item still under potential revision is the manipulator. Throughout the testing the mechanical team members are analyzing different methods to grip rocks picked up. The team is also sending feedback to sponsors of work to date in appreciation of their support.

9 Acknowledgements

The UB Space Bulls team would like to sincerely thank all of our donors, sponsors, and supporters. Through the support of the School of Engineering, our advisors and faculty members we were able to see this opportunity through to the end. Without the generosity of our peers, donors and industry partners we would not be as successful as we are today.

10 References

http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29

"PROJECT EVALUATION CRITERIA AND SCORING APPROACH," NASA/NIA, [Online]. Available: http://nia-cms.nianet.org/RoboOps/Program-Overview/2015/2015-Robo-Ops-Competition-Rules.aspx