# ENVOY : Exploration and Navigation Vehicle for geolOgY University of Pennsylvania's Entry in the RASC-AL Robo-Ops Challenge

Arunkumar Byravan, Aylin Caliskan, Jonas Cleveland, Daniel Gilles, Jaimeen Kapadia, Theparit Peerasathien, Bharath Sankaran, Alex Tozzo
Faculty Advisor: Dr. Mark Yim

**Abstract**

This paper documents the approaches and algorithms adopted by University of Pennsylvania's entry for the RASC-AL Robo Ops competition. For this competition we adopt a highly modular design approach for our rover development, to keep the systems independent and reusable. Our algorithms demonstrate a sliding autonomy framework where the rover is capable of autonomous decision making with varying degrees of human intervention. In our framework we also discuss the philosophy behind our modular design and how it enhances the effectiveness of our remote operations capability. Our hierarchial software architecture provides us with a robust and fault tolerant system. We've also incorporated innovative designs to reduce costs and demonstrate proof of concept.

The paper also provides insight into the our outreach activites which have been the primary driving force behind our participation. In our outreach efforts we managed to engage both the academic and local communities. These activities have enabled us to inform and educate a large portion of the community on the benefits of space exploration and robotics.

## I. Introduction

The RASC-AL robo - ops competition is a challenge to space enthusiasts in the academic community to design and develop a teleoperated rover capable of performing a series of challenging tasks while navigating simulated extraterrestrial terrain. The focus of this competition is to also demonstrate the challenges involved in space exploration and the importance of robotics in these endeavours. Some of the most critical challenges that needed to be addressed included

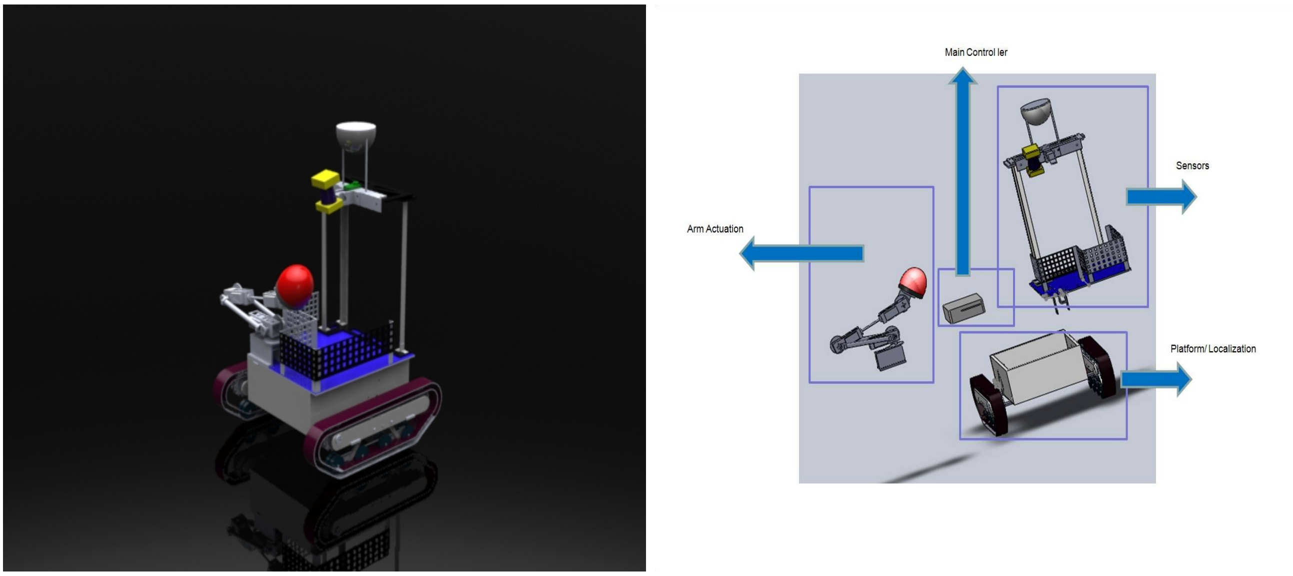
1) Remain within size and mass budget contrainsts
2) Navigate rugged and uneven terrain with appreciable slopes in varying weather conditions
3) Detect and transport various colored rocks.
4) Be remotely controlled from the Home Campus of the University.

Our goal was to develop an inexpensive, lightweight and highly adaptable platform to effectively navigate difficult rock terrain. In the essence of the RASC-AL competition a main objective was to employ a modular design philosophy in which each component is independent and fault-tolerant. This allows different systems such as the sensor suite, gripper/arm actuator, and even suspension to be swappable. We have included the modular components best suited for the objectives of this competition.

Fig. 1. Picture of the platform

## II. VEHICLE PLATFORM

Our rover was built from a tracked vehicle's platform with added controls for drive by wire. Since the RASC-AL competition was planned to be held at the Johnson Space Center rock yard, our platform had to be capable navigating both rugged rocky terrain and low traction sandy terrain. To accomplish this task we needed a proven mobility platform capable of navigating either with ease. The rover is also equipped with a rich sensor suite for perception, planning and localization. The list of all electronics components onboard the platform with their respective power draws are listed in TABLE I. The onboard sensors are interfaced to the a Mac mini, which serves as the primary onboard computer. The Mac mini also communicates with a micro-controller which controls all the onboard actuators. All systems excluding the arm are powered by a 5-cell 12V battery. The arm is independently powered by a 7.5V LiPo battery. The vehicle weighs 52.5lb without payload and generates a max torque of 34.48 lb-ft at a maximum forward velocity of 6.82 miles/hour. The vehicle is also highly modular with easily interchangable components demonstarted in Fig II.




(a) (b)

Fig. 2. Rendered CAD model and Exploded view of components

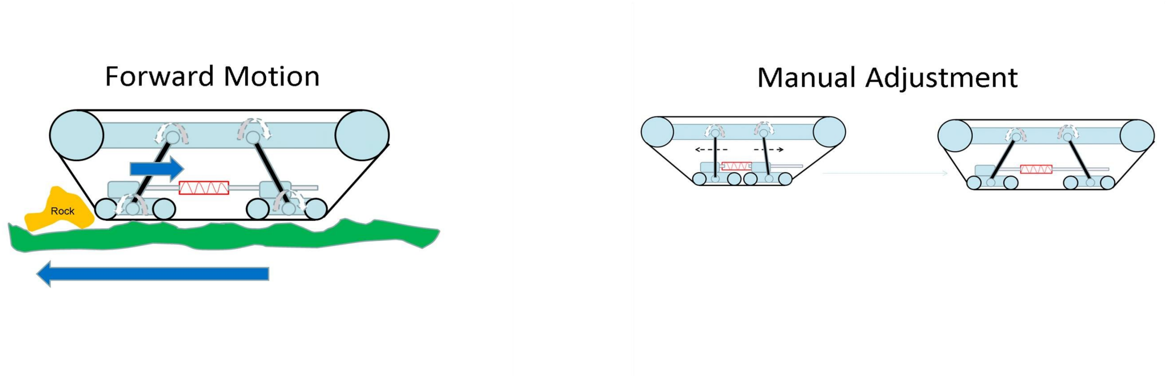
| Device | Power Supply |
|---|---|
| Hokuyo UTM 30Lx | 12 V |
| Mac Mini | 13.5V |
| Encoders | 5V |
| Polulu Microcontroller | 12V |
| Air Pump | 12V |
| Logitech Onboard Cameras(2) | 10V (USB powered) |
| Ublox 6 GPS Engine | 3.6V (USB powered) |
| Verizon Broadband Card | 5V (USB powered) |
| IMU | 2.5V (powered from microcontroller) |

TABLE I
ONBOARD ELECTRONICS LIST

### *A. Base Platform*

Our base platform is a dual tracked mobility platform. We employ a novel, dynamic Hortsmann tank tread suspension which utilizes a shock-absorbing coil spring to maintain speed while driving over difficult terrain, and a manual tensioner for easy assembly.Our platforms lightweight and swappable component design allow it to be used for a variety of missions including planetary exploration, multi-robot tasks, and astronaut assistance. These configurations are illustrated in II-A

The tracked chassis supports a sensor housing built from aerospace grade (6061) aluminum, which also houses the motors powering the sprockets. The gearshaft motors are controlled via the central microcontroller. The motors receive pulse width modulated (pwm) signals from the microcontroller which in turn is interfaced with the central computer through USB.




(a) Forward Motion (b) Manual Adjustment

Fig. 3. Suspension Adjustments

*1) Drive by Wire:* As depicted in II-A.1, the drive-by-wire actuation was performed via the Mac mini. The Mac mini communicates serial commands to the 12-channel mini maestro USB servo controller manufactured by Polulu. This included DC servomotors to actuate the gearshaft motors for the drive train, the arm servos and the air pump for the gripper. To transmit the digital control from the computers to the drive by wire system, we implemented a simple digital to analog converter using a digital programmable integrated circuit (PIC) microcontroller with a resistor-capacitor (RC)-filtered pulse-width-modulation output. The PWM frequency was set to 100Hz with 8bit resolution to enable smooth and accurate control of the actuators. The internal system consisted of a 12V 5cell battery which powered onboard computer and acutation motors. Of the entire perception sensor suite, only the LIDAR needed a direct power source. The GPS, broadband card and cameras were USB powered from the Mac mini. The IMU and encoders were powered from the microcontroller which was in turn powered from the 12V onboard supply. A separate 7.5V battery powers the Arm and it's components.

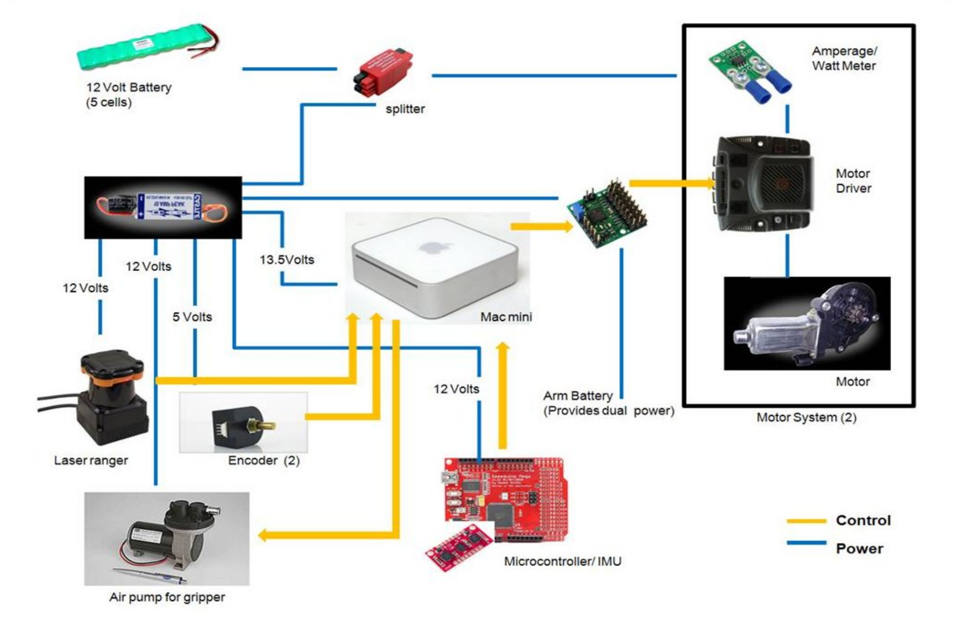



Fig. 4. Drive by Wire System

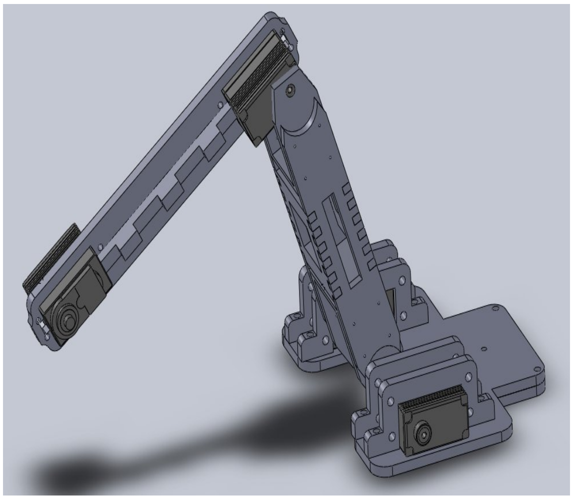
## B. Arm

The rover has a two link arm with four revolute joints. The placement of the arm on the right front side of the rover serves two purposes. The first goal was to avoid obstructing the view of the omni-directional camera and the Hokuyo range finder. The second goal simplifies construction because the base servo motor does not need to be geared down. This configuration might be required in order to reach in front of the rover and turn around to deposit rocks into the storage bin.

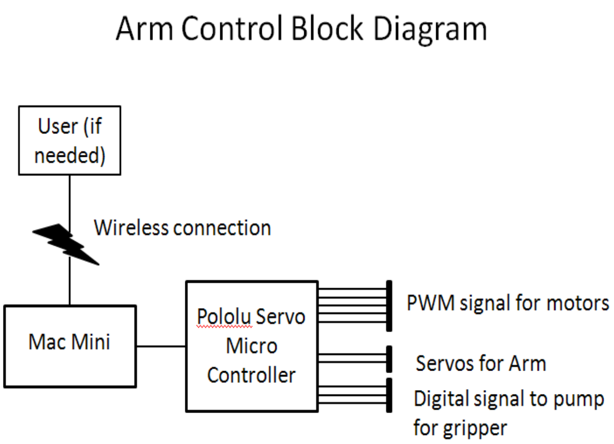
This design resulted in limiting the workspace area to a 120 degree envelope in front of the right corner of the rover. A small camera is installed on the first link of the arm for grasp verification. In our proposal, two options were considered: a commercial-off-the-shelf arm or a custom design. A custom design was preferred as it enhanced flexibility of the footprint of the arm. Other benefits included adjustable link length and a modular gripper attachment unit.
The link lengths were chosen to allow the arm to pick up objects of interest as far as 10-13 inches away from the base of the rover. The base joint allows for a 180-degree rotation about the vertical axis of the rover. The second joint in the kinematic chain consists of two servo motors to increase the lifting moment. Each servo has a torque rating of 24.0 kg/cm. The arm fully extended is 38 cm. This gives a lifting force of about 630 grams at full extension excluding the weight of the arm and the gripper. The arm is constructed from ABS plastic and weighs approximately 80 grams. Each servo motor weighs 65 grams. Table II details the servo motor specifications. Table III documents the torque requirements and calculations.
A Pololu Mini Maestro 12 was used to control the motors of the robot. This microcontroller provides12 analog output

| Motor | **Hitech HS-7955TG** |
|---|---|
| Torque(4.8V/6.0V) | 18/24 kg/cm |
| Size | 39.88x19.81x36.83 (mm) |
| Weight | 64.92 (g) |

TABLE II
SERVO MOTOR SPECIFICATION



(a) Arm CAD Model




(b) Control Block Diagram

Fig. 5. ARM

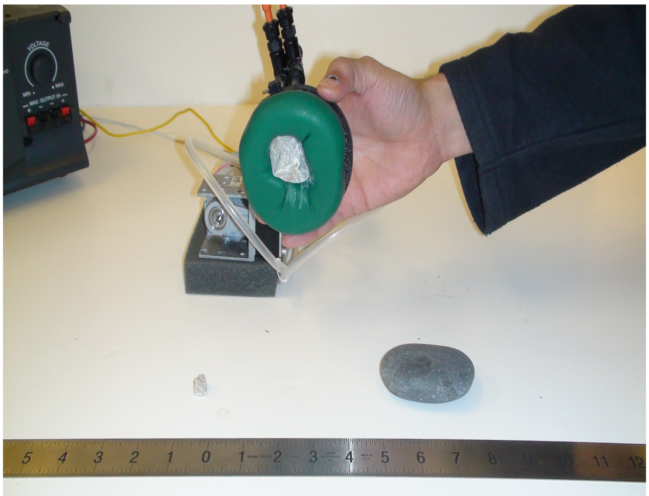
| Item | Mass (kg) | Distance from Base(cm) | Torque Component (kg-cm) |
|---|---|---|---|
| Servo1 | 0.130 | 1 | 0.130 |
| Servo2 | 0.065 | 15 | 0.975 |
| Joint1 | 0.010 | 25 | 0.250 |
| Gripper | 0.460 | 32 | 14.720 |
| Payload | 0.150 | 35 | 5.250 |
| Servo3 | 0.065 | 32 | 2.080 |
| Torque Required at Base (kg-cm) + 25% Margin | | | 23.405 + 5.851 = **29.256** |
| Total Torque Available at Base(kg-cm) | | | **48** |

TABLE III
TORQUE REQUIREMENTS

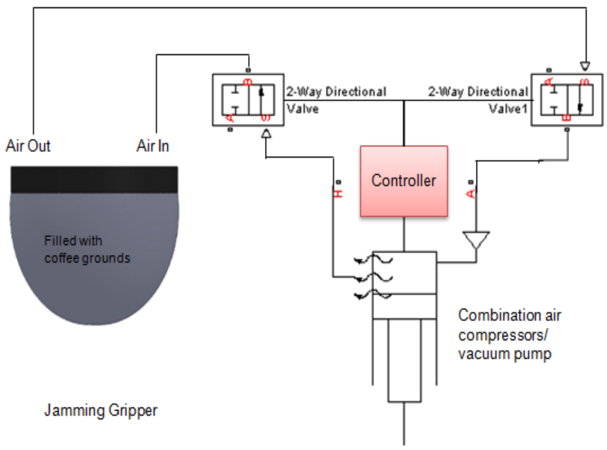
channels. Each channel can be individually set to an analog output or a PWM output. The microcontroller connects to the main computer via USB. These features greatly simplified the motor control aspect of the rover design. The pump and the solenoid valves were also controlled through the analog output. Power is provided separately to each of the arm servo motors through a 7.4 V voltage regulator. Fig II-B illustrates the control architecture of the motors.

### C. Gripper

Lipson and Jaeger [1] demonstrated that an amorphous elastic bag filled with a granular media can be used to grab a large variety of object sizes and shapes. They use a two step process for gripping objects. The gripper is pressed down onto an object so that the elastic bag that forms the bulk of the gripper deforms around the object. A vacuum is formed inside the gripper that causes the granular media to ”jam” against each other to form a rigid shape. This process yields forces that facilitate the grasping of the object.
The jamming gripper allows us to rapidly pick uprocks of unknown surface properties and irregular shapes. This can be acheived without increasing the degrees of freedom of the gripper, reducing hardware and software complexities. Such a design makes the system extremely cost efficient.



(a) Gripper Photograph




(b) Circuit Schematic

Fig. 6. Gripper

*1) Design:* The schematic shown in Fig II-C illustrates the components of the jamming gripper. The jamming gripper consists of a bag filled with granular media. The bidirectional pump is responsible for generating and releasing the vacuum within this bag to jam or free up the media allowing us to pick-up or drop objects. Two way directional valves are used to control the air flow to and from the pump. Filler Media was a crucial factor in determining the performance of the gripper. Based on our tests we concluded that the grains for this media should have high young’s modulus with a low mass density . Coffee grounds meet both of these requirements and were found to be the most effective filler media. The surface properties of the bag that contains the filler media greatly affect the performance of the gripper. We experimented with several materials and found latex to be the best suited for the gripping task. Latex has good surface friction properties and moderate tear resistance.

### D. Sensors

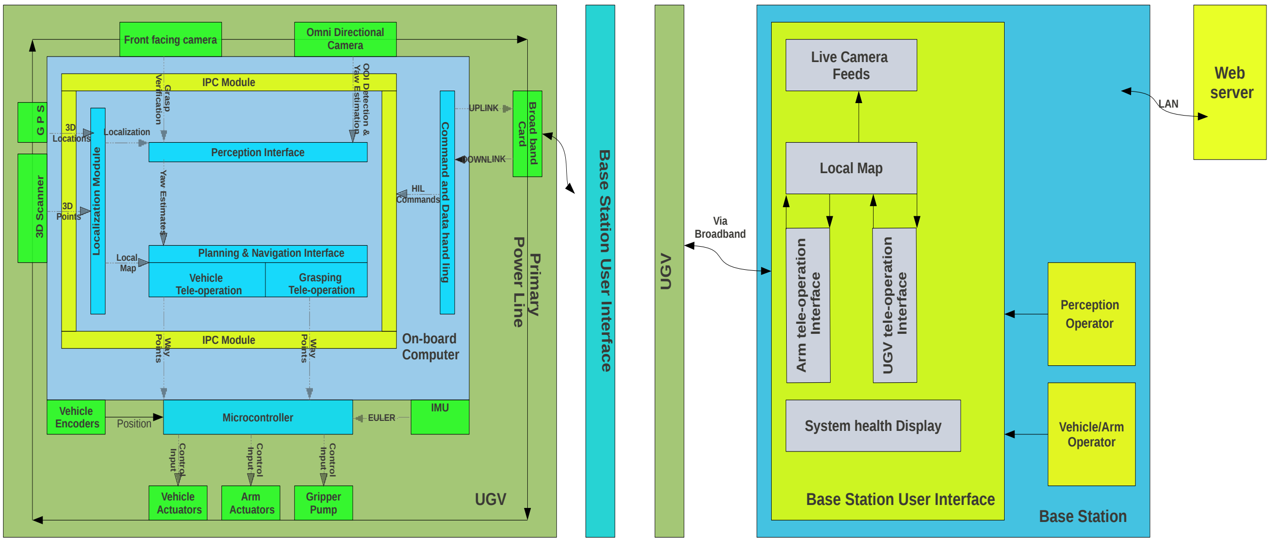
The list of sensors used on the robot are shown in TABLE IV. The sensors were used for Object Of Interest (OOI) detection, ground plane estimation and localization. For detecting the colored rocks we used one omni-directional and one regular camera from our perception stack. The omni-directional camera was also used for navigation while the regular USB camera was used primarily for grasp verification. A forward tilting LIDAR was used for ground plane estimation and obstacle detection. The localization stack makes use of the onboard gps information along with IMU and encoder data for accurate position estimation. The sampling frequencies of the various sensors are illustrated in TABLE IV.

| **Sensor** | **Sampling Frequency** |
|---|---|
| LIDAR (Hokuyo UTM 30Lx) | 30Hz |
| Encoders | 40Hz |
| Omni directional Camera | 30 Hz |
| Front Facing Camera | 30 Hz |
| Ublox 6 GPS Engine | 5Hz |
| IMU | 100Hz |

TABLE IV
SENSOR SAMPLING FREQUENCY

# III. SOFTWARE ARCHITECTURE

The rover software architecture is distributed into two modules: the base station module and UGV module. The UGV module consists of three stacks:
1) Planning & Control Stack 2) Perception Stack 3) Localization Stack The Planning & Control stack operates on the local costmap generated by the forward tilting LIDAR. The Controller is a rollout trajectory follower inspired from the LAGR program [2]. This subsystem operates within a limited horizon due to the lack of features for long term planning.
The Perception module runs a Bayesian color classifier at 30Hz for Object of Interest (OOI) detection. The perceptual module also interfaces with the planning and control structure for higher level decision making. The Localization module fuses information from the various sensors to accurately estimate pose & location information in the global(gps) frame. These modules are hierarchially interfaced using a finite state machine which accurately transitions between the various states of the CONOPS architecture. The transition functions are defined over the current state of each stack and the various sensor inputs. All of the systems software stacks are running on separate threads and they communicate via the IPC.




(a) UGV

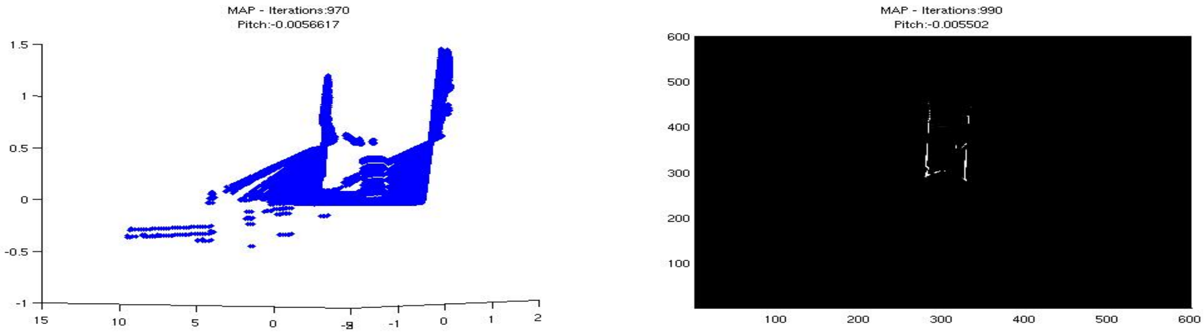
(b) Base Station

Fig. 7. Software and System Architecture

## A. State Machine

The State Machine defines all the autonomous states of the robot. A sample of the states and their corresponding actions are described below:

1) Initial state : The robot remains stationary in this state and waits till the GPS gets a good fix on its position.
2) Predetermined Motion state  Once the GPS has a fix on the robot's position, it publishes a message over the internal process communication. Once the control module receives this message it executes a simple linear control for a time period of 1 second. This motion is executed to estimate the robot's orientation in the GPS frame.
3) Wait state: Once the robot determines its orientation, it waits until it is triggered by an external command. The base station operator then identifies regions for exploration based on the feedback from the perception stack. The operator then identifies a goal position for the rover.
4) Scan/Costmap state: Once the robot gets the goal position, it reorients towards the goal's general direction. It then starts scanning using the tilting LIDAR to generate a local costmap. An example of the 3D points from the LIDAR scans and a map generated is shown in Fig III-A.
5) Plan state: Once a costmap is generated, the robot plans on this costmap. The goal is a point in the direction of the "global" goal, i.e the user annotated point.
6) Motion state: In this state, the robot executes the local plan by the means of a trajectory following controller. The controller is capable of online obstacle detection and avoidance.
7) Manual state: In this state, the robot is under manual control which includes control of the drive motors, the arm's servo motors, and the pump. This the rover's OOI grasping state.




(a) 3D Point Cloud from tilting LIDAR

(b) 2D Cost Map

Fig. 8. Mapping

The above mentioned states are the high level overview of all possible sub-states that the robot can achieve. The operator can force state transitions in the state machine. Finally, there are three possible ways in which the goal position can be set for the rover. In decreasing order of priority, they are:.

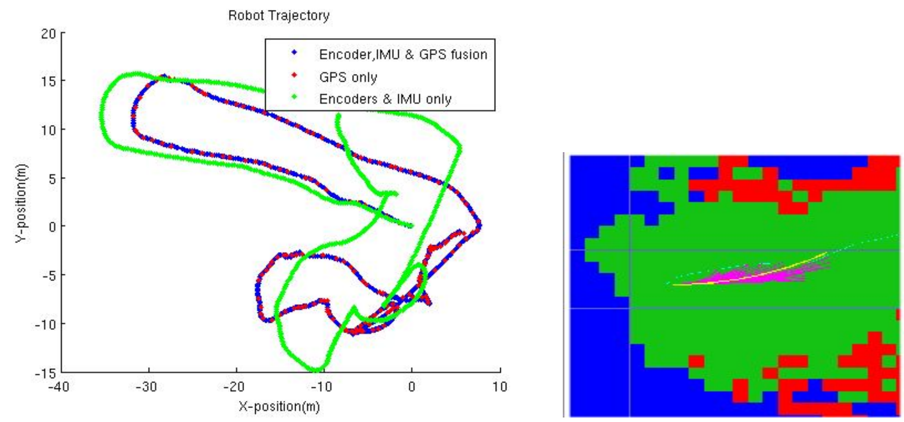
1) Manual assignment : Operator specifies the goal
2) Autonomous perceptual control: Goal identified by the color classifer.
3) Exploration Mode : Rover tries to explore its environment.

### B. Localization and Control

*1) Localization:* One of the key components enabling the robot to be semi autonomous is the localization module. It fuses information from the U-Blox EVK-6H GPS module, the IMU and the wheel encoders. Due to the difference in sampling frequency between the sensors we use two independent UKFs.

The first UKF estimates the robot's orientation. The input state is a three dimensional vector comprising, the yaw,pitch & roll of the robot. The process update model, uses the IMU rate gyro information while the measurement update uses the direction of the g-vector from the IMU. The Pitch and Roll estimates from this UKF are very accurate while the Yaw estimate tend to drift a lot. Therefore, we neglect the yaw estimates from this setup. All the computation is done on the microcontroller and then published to the system via the IPC.

The second UKF estimates the robot's 2D pose. The state consists of the x,y and yaw (Heading) of the robot. There are two different process updates in this model. For the X and Y position process update, we use the wheel encoders whereas the process update for yaw utilizes the yaw rate measurement from the IMU. The measurement update is received from the GPS, where the X & Y positions are with respect to the starting position of the robot. The heading is calculated using the change in x and y positions. Plots showing results from an outdoor trial using the UKF are shown below in Fig III-B.1. This localization module runs in a separate thread that keeps on publishing position updates to the system through the IPC.




(a) Sensor fusion for localization

(b) Trajectory Follower Fig from [2]

*2) Path Following:* The path following module is responsible for calculating linear and angular velocities for steering the rover to follow the desired trajectory as accurately as possible. A regular Astar planner is used to plan in the local occupancy grid. Once this plan in computed the trajectory follower executes the control commands to follow this planned

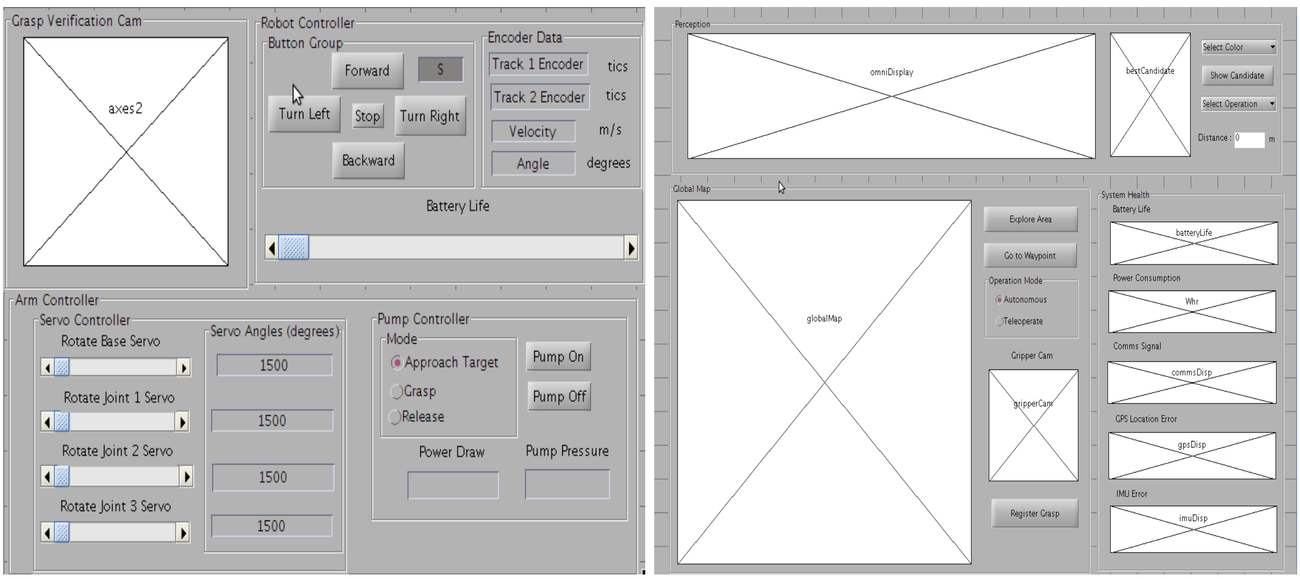
path as closely as possible.
The trajectory follower used on the rover is the rollout based follower initially devised for the LAGR program . In this controller a set of control commands from the rover's state space are sampled. These samples are then propagated to a limited time horizon, (1 sec in our case). Then the controller which minimizes a cost function is picked as the best controller for the robot's current pose. The cost function that is minimized is a linear combination of features and is given as

$$C(t) = \alpha Obs + \beta Gdist + \gamma Pdist + \delta \frac{1}{v^2}$$

where *Obs* is the obstacle cost (taking into account the robot's actual footprint). *Gdist* and *Pdist* are the distance from the final point in the simulated control, to the goal and planned path respectively. The final term in the cost function is to ensure that the maximum velocity which executes the control is selected when minimizing the cost function.

### C. Teleoperation

Teleoperation of the robot is a key component in the system design. The team will be communicating with the robot over the broadband connection. The rover needs to follow the commands sent and relay information back to the base station. The teleoperation module of Team ENVOY's rover has two parts. The first part is the lower level robot control. This includes the control of the arm and manipulator as well as the rover itself. The second part consists of the higher level planning module.
Apart from it's semi autnomous behaviour the robot is very similar to a remote controlled vehicle. Initial tests were actually performed with a standard radio frequency controller. The Logitech Rumblepad 2 gamepad was chosen as a final teleoperation interface primarily due to its USB interface. The controller is mapped to directly drive all the motors on the robot. *Another gained advantage was our ability to improve our outreach efforts due to the popularity of game controllers amongst children.* A second interface for the low level control is the Matlab GUI shown below. The GUI has all of the functionality of the joystick in addition to a grasp verification view, velocity feedback, and estimated battery life. This provides a more detailed outlook on the robot compared to the joystick at the cost of some usability (e.g. multiple button presses for simultaneous motions of the arm and platform). The second level of teleoperation is the high level planner. Shown below Fig III-C is the perception GUI. The perception GUI displays some critical information to the user that overlaps with the control GUI such as battery life and grasp verification. Specific to the perception GUI are the omni-vision display, the global map, the best candidate object within view, sensor errors and system health. These elements are crucial to the path planning module and give us an indication of the robot's current status. The global map can also receive a user commanded input directing the rover to explore a specific region indicated by clicking a point within the global map. We can also set up a series of waypoints for the rover to follow such as for navigating around particularly difficult terrain.




(c) Control GUI

(d) Perception GUI

Fig. 9. Graphical User Interfaces

### D. Perception

The perception stack consists of two components, one a bayesian color classifier for Object Of Interest (OOI, i.e rocks) detection, the second component is a grasp verification interface. The OOI detection module uses a simple Gaussian Mixture Model for color classification. The GMM classifier was trained on hand labeled data using an expectation maximization framework. On every input frame from the omni-directional camera, the algorithm runs the color classification routine to extract OOI's from the current frame. This output is then parsed through a post processing routine which evaluates shape statistics on connected components to produce the final detection results. These detection outputs are then relayed via the user interface for evaluation by the human operator. Once the human operator evaluates the detection results, this output is parsed to the state machine to update the system state which execute a sequences of actions to move towards the target. The entire OOI detection component is run on the rectified and unrolled Omni camera images. The omni image unrolling sequence is illustrated in Fig III-D.

(a) (b) (c)

Fig. 10. Omni Image, Rectified Image, Unrolled Image

The second component of the stack consists of the grasp verification module. This is just a simple user relayed interface, where the Operator can verify from the front facing camera mounted on the arm if the grasping operation was successful.

### E. Communications

*1) Inter Process Communications:* There are two types of communication modules in the ENVOY robot. The first type is an InterProcess Communication (IPC) module. IPC module is used for data parsing between internal processes running on the robot. The ENVOY robot has two types of internal processes, Device process and the Main-IPC process. Each actuator and sensor on the rover has its corresponding device process for receiving information (example: GPS drivers, camera drivers) and sending controlling signals (example: wheel motors and arm servos process). The Main-IPC process is the central node communicating with all device processes. When the device process is ready to send data, it pushes the data into the output pipe along with its corresponding device name. The Main-IPC process iteratively checks the output pipe. When the output pipe is not empty, Main-IPC process pops the data from it and converts it to a UDP packet according to the device name. This packet is then streamed over the internet. On the other hand, Main-IPC process also acts as a server. When UDP packets are received by the rover via the broadband network, the Main-IPC process examines the packet and sends the data to corresponding device processes.

*2) Broadband Communications:* Another aspect of the robot communication is internet module. The internet module establishes a communication link between the rover and the home campus pc station. It receives the ready-to-send UDP packets from the Main-IPC process and sends them over internet via the Verizon 4G broadband network and vice versa. The diagram below illustrates the structure of the robot communication modules.

*3) PC Station:* The PC station is setup to send the control commands to the ENVOY rover and receive the robots information. This module resides in Control GUI and the ENVOYs Internet module. It receives the control commands from the GUI and sends them over the internet in the UDP format. It simultaneouly receives incoming UDP packets from the rover, which it unpacks and relays to the corresponding section of the GUI. In addition to this data exchange, the video stream from ENVOYs front camera will be sent to the webserver for broadcasting a live video stream via Local Area Network (LAN). The diagram below shows the structure of the PC station communications.

*4) Webserver:* Our webserver consists of two dependent modules. The first is a HTML web server run using Apache Tomcat and a HTTP web server, for generating a dynamic HTML web page. The second module is the live video stream. The Video-stream module is the component that receives the live feed from PC station and transmits it over the internet

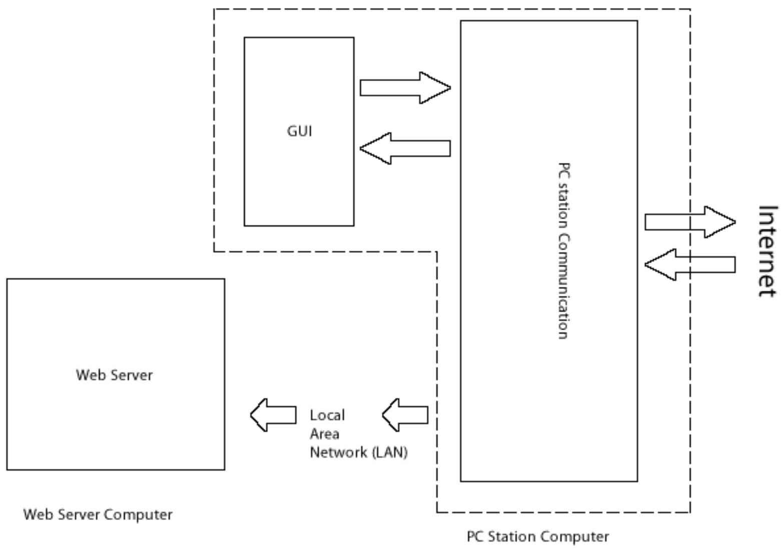



Fig. 11. Broadband Communication Protocol

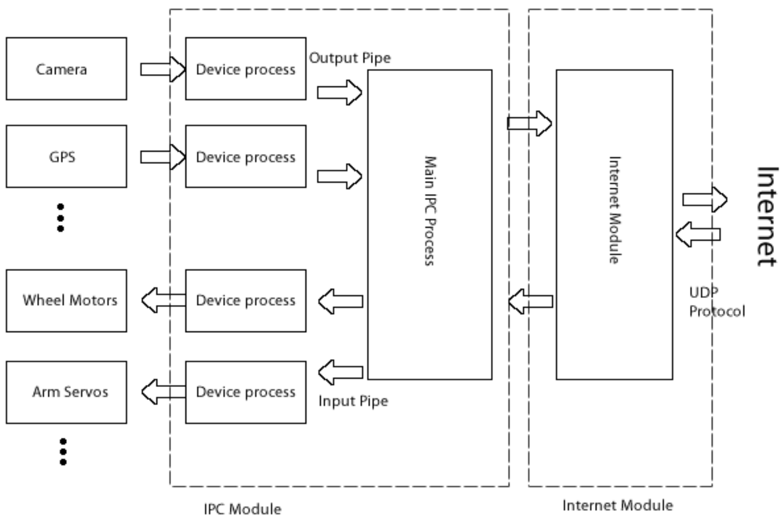



Fig. 12. PC Station Communication Protocol

using a TCP protocol. Any client can receive this information via a web browser embedded with a JAVA applet in the html page. The diagram below shows the structure of the Web server communications.

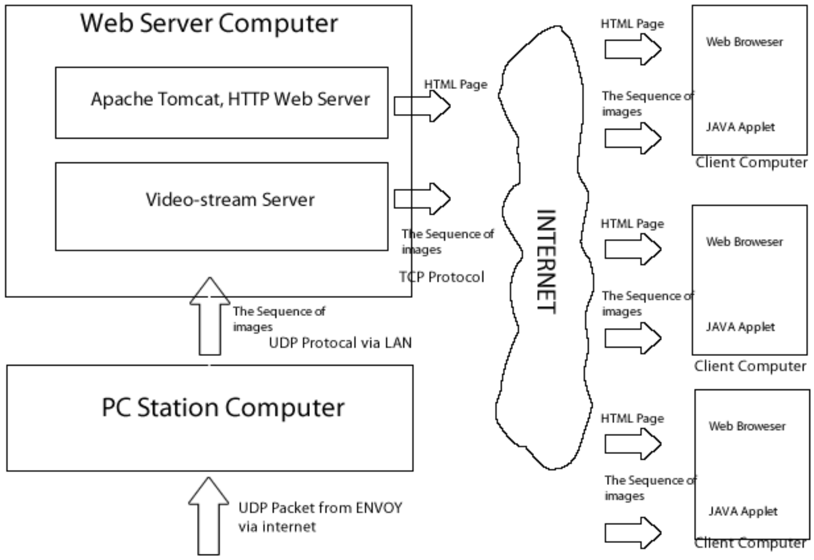



Fig. 13. Webserver Communication Protocol

## IV. Outreach

The primary goal of our team at GRASP has been to connect the community surrounding UPenn with the space-exploration community and utilize our resources to ignite children's excitement regarding space and robotics. After receiving notification of our RASCAL proposal's acceptance, we have embarked on weekly outreach events targeted at the Philadelphia School District as well as the larger community. Altogether, members of the team were involved in more than 300 hours of outreach activity and communicated with over 1,000 children and parents in the Philadelphia region. Our outreach events and activities were built to engage students, spark there interest in science, and increase their knowlegdge of NASA's history and importance.
Here is list of our most significant events:

### *A. Career expectations program from the Philadelphia School District:*

Locations: Welsh valley Middle School, Bala Cynwood Middle School
Description: We spoke to students concerning the need for space exploration and hosted a game where students were able to share their knowledge of NASA's rover history. Students were made aware of the practical concerns in space travel such as weight, cost, and fuel. Students also learned about possible career focuses in space exploration and engineering fields.

### *B. National Robotics Week*

Locations: GRASP Laboratory, University of Pennsylvania
Description: We organized an open house at the GRASP Lab, where students and members of the community were given a tour of the laboratory and a demonstration of some of the lab's most exciting projects. In the course of the day, we hosted a game where students planned a rescue mission for astronauts using our rover. We presented to students about the RASC-AL challenge and the significance of robotics in space exploration. At the conclusion of the event, students were given the opportunity to play with the gripper and drive our rover via remote. Over 100 middle school students, were bussed to our lab for this event.

### *C. Philadelphia Science Festival*

Location: Outside Philadelphia Science Museum on Ben Franklin Parkway
We hosted a tent at the Philadelphia Science festival. We used this opportunity to inform visitors about the importance of science in early school education and the role of robotics in space exploration. Here, the general public were informed about upcoming events at the GRASP Laboratory and were able to sign up for a tour of the lab and GRASP's summer camp. Visitors had the opportunity to drive the rover around via remote, and received literature describing the RASCAL competition and our rover's design.

(a) (b)

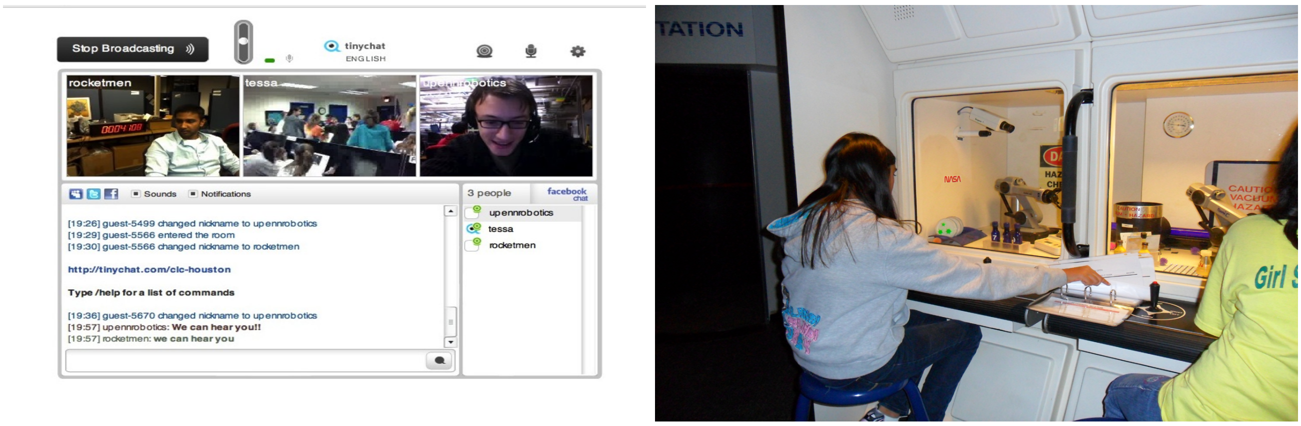
Fig. 14. National robotics week open house


(a) (b)

Fig. 15. Philly Science Festival

### D. *Challenger Learning Center*

Location: Challenger Learning Center, Needville Texas and UPenn via Webcast
Here students were tasked with a mock mission to rescue astronauts on a mars mission. Members of our team supported CLC by being the offsite engineering team for the rescue mission. This was a fun activity for students while simultaneouly learning about mission control duties.




(a) (b)

Fig. 16. Challenger Learning Center

### E. The Immersive Kinematics Art Show

Location: Sinergy Art Gallery, North Philadelphia
Description: Our platform's modular design enables our components to be easily swapped with other tool suites. Through collaboration with our advisor, our platform will be exhibited at a Philadelphia art show on May 7. The top of our robot which houses our arm and perception sensor suite will be replaced with cameras and an interactive LED display. Patrons will learn about robotics and its many applications.

## V. Conclusions

In this paper we presented some of the technical details of ENVOY the rover built by University of Pennsylvania's team for the RASC-AL robo-ops competition. After quantifying the performance criteria for the RASC-AL challenge we built the hardware and software to meet these stringent criteria,. We also demonstrated that robust systems can be developed from low cost hardware while not comprimising on efficiency. An array of sensors such as GPS, LIDAR, IMU and an Omnidirectional Camera were utilized for enabling short range sensing and control. The software modules were written to be robust and fault tolerant when integrating sensor information for perception, localization and control. We also incorporated a HIL interface into our design to demonstrate the importance of sliding autonomy in space operations.

## VI. Acknowledgements

We would like to express our sincere gratitude to Alex Kushuylev for his invaluable advice and hardware assistance without which we would have not been able to procure the sensors and computing required for our system. We also like to thank Peter Szczesniak for helping us with our hardware fabrication. Other individuals whose support and assistance has been crucial to the success of this project and our outreach activities include, Dr. Kostas Daniilidis, Dr. Daniel Lee, Rebecca Stein, Carolyn Mckinney, Elaine Wisser, Tracy Knauss, Peggy Halford and many others.

## References


[1] E. Brown, N. Rodenberg, J. Amend, A. Mozeika, E. Steltz, M. R. Zakin, H. Lipson, and H. M. Jaegar, *Universal Robotic Gripper based on the Jamming of Granular Material*.

[2] B. P. Gerkey and K. Konolige, *Planning and Control in Unstructured Terrain*.

[3] *Little Ben: The Ben Franklin Racing Team's entry in the 2007 DARPA Urban Challenge*, 2008.

[4] M. Likhachev and D. Ferguson, *Planning Long Dynamically-Feasible Maneuvers for Autonomous Vehicles*, 2003.

[5] P. Vernaza and D. D. Lee, *Rao-Blackwellized particle filtering for 6-DOF estimation of attitude and position via GPS and inertial sensors*, 2006.

[6] P. Vernaza, B. Taskar, and D. D. Lee, *Online, self-supervised terrain classification via discriminatively trained submodular Markov random fields*, 2008.

[7] D. Fox, W. Burgard, and S. Thrun, *The Dynamic Window Approach to Collision Avoidance*.

[8] J.-P. Tardif, Y. Pavlidis, and K. Daniilidis, *Monocular visual odometry in urban environments using an omnidirectional camera.*, 2008.

[9] J.Butzke, K.Daniilidis, V.Kumar, A.Kushuleyev, D.D.Lee, M.Likachev, C.Phillips, and M.Phillips, *Magic 2010 Technical Paper, University of Pennsylvania*, 2010.

[10] T. Sebastian, *Probabilistic robotics*, New York, NY, USA, March 2002.