2012 Alabama Lunabotics Systems Engineering Paper

School:
University of Alabama

Team Name:
NASACAR

Team Members:
Justin Baker, Jessica Colburn, Justin Headley, Adam Melton, Dalen Mullenix, Andrew Price, Logan Ream, David Sandel, Mitchell Spryn, Stephanie Troy, Jason Watts, Matt Westberry

Faculty Advisor:
Dr. Kenneth Ricks

ABSTRACT

Excavation will hold a key role for future lunar missions. NASA has stated that "advances in lunar regolith mining have the potential to significantly contribute to our nation's space vision and NASA space exploration operations." [1]. The Lunabotics Mining Competition is an event hosted by NASA that is meant to encourage "the development of innovative lunar excavation concepts from universities which may result in clever ideas and solutions which could be applied to an actual lunar excavation device or payload." [2]. Teams entering the competition must "design and build a remote controlled or autonomous excavator, called a lunabot, that can collect and deposit a minimum of 10 kilograms of lunar simulant within 10 minutes." [2].

While excavation will play an important part in lunar missions, there will still be many other tasks that would benefit from robotic assistance. An excavator might not be as well suited for these tasks as other types of robots might be. For example a lightweight rover would do well with reconnaissance, and a mobile gripper arm would be fit for manipulation, while an excavator would be comparatively clumsy and slow in both cases. Even within the realm of excavation it would be beneficial to have different types of excavators for different tasks, as there are on Earth.

The Alabama Lunabotics Team at the University of Alabama has made it their goal to not only design and build a robot that could compete in the Lunabotics Mining Competition, but would also be a multipurpose tool for future NASA missions. The 2010-2011 resulting robot was named the Modular Omnidirectional Lunar Excavator (MOLE). Using the Systems Engineering process and building off of two years of Lunabotics experience, the 2011-2012 Alabama Lunabotics team (Team NASACAR) has improved the MOLE 1.0 design and optimized it for the 2012 Lunabotics Competition rules [1]. A CAD model of MOLE 2.0 can be seen below in Fig. 1.

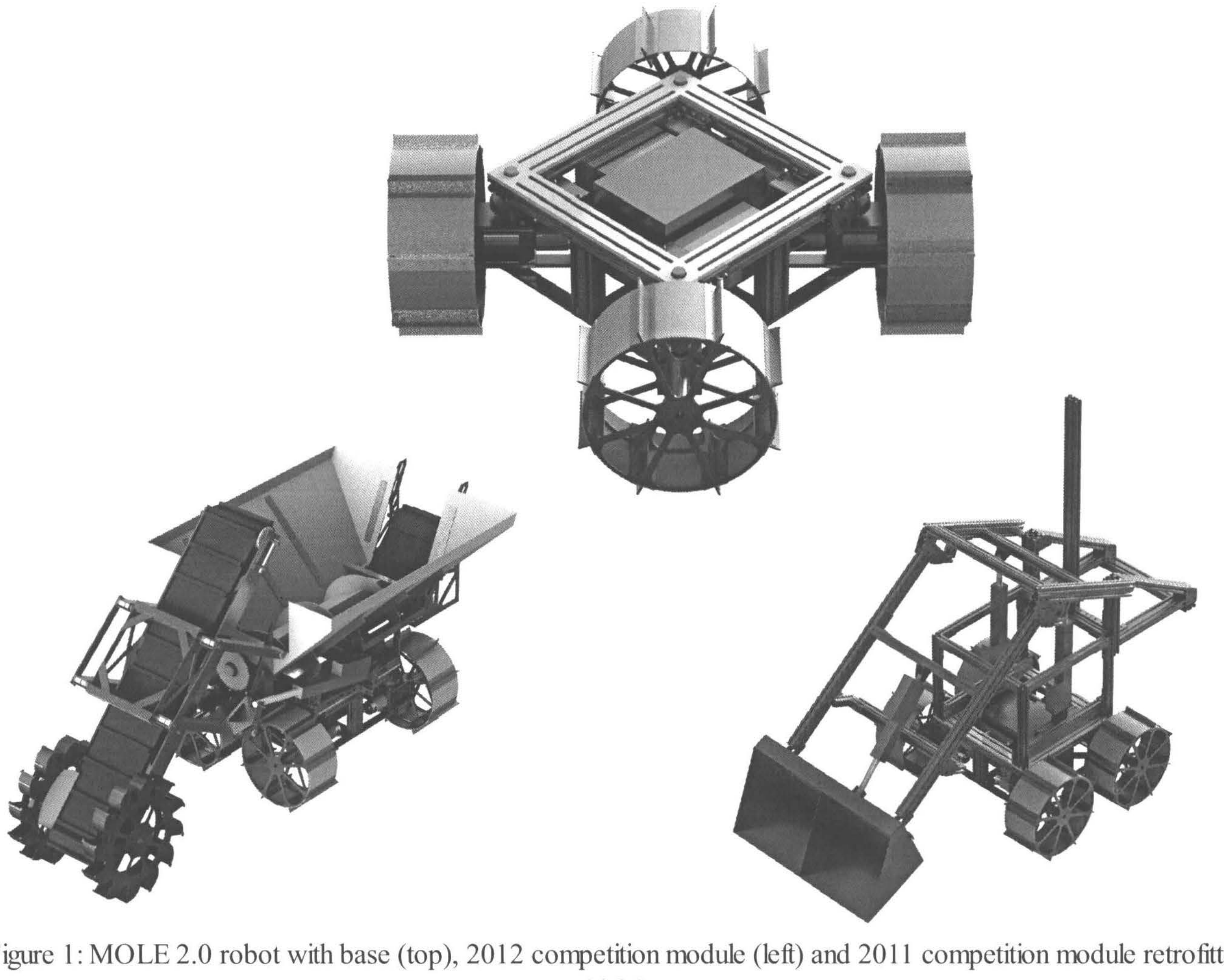


Figure 1: MOLE 2.0 robot with base (top), 2012 competition module (left) and 2011 competition module retrofitted (right).

Table of Contents

INTRODUCTION

Team NASACAR used the Lunar Engineering Handbook, written by David Beale, as a guide to the application of a Systems Engineering approach [3]. The team defined mission objectives, derived system requirements from these objectives, divided the project into clear subsystems, and in general followed the phases of the Systems Engineering life cycle as seen in the Vee Chart in Fig 2. Within each phase the 11 Systems Engineering functions (as seen in Fig. 3) were implemented.

The primary goal of this project was to apply a Systems Engineering approach to the design and implementation of a modular lunar regolith excavator and to demonstrate its capabilities during the 2012 NASA Lunabotics Mining Competition. The secondary goal of the project was to design the robot so that it could be easily expanded in the future to be applied toward functions other than excavating that are also important for lunar missions. Other objectives included gaining experience with efficient hardware design and software-architecture planning, learning the processes involved in working with a multidisciplinary team, obtaining valuable engineering skills for future careers, and spreading enthusiasm for Science, Technology, Engineering, and Mathematics to K-12 students. This paper documents the Systems Engineering process followed by Team NASACAR during the project.

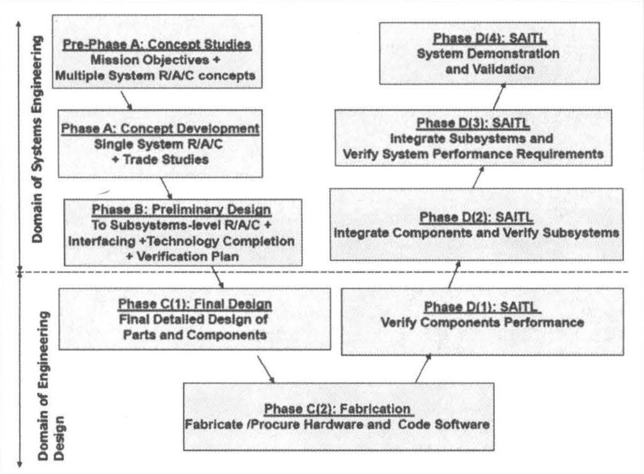


Figure 2: Systems Engineering Life Cycle Vee Chart

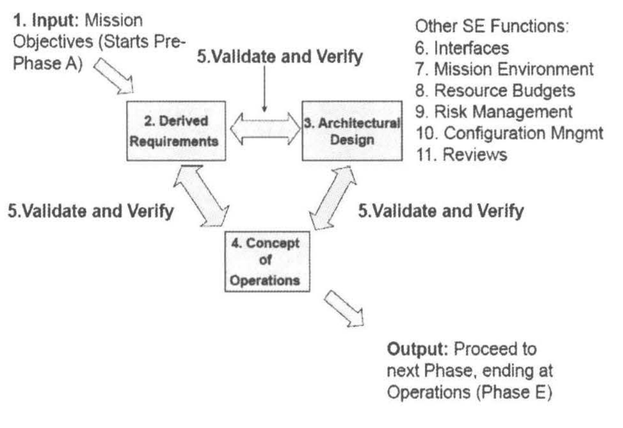


Figure 3: The 11 Systems Engineering Functions

SYSTEMS ENGINEERING

Phase A: Concept Development

Due to the directed nature of the project and limited time, Pre-Phase A and Phase A were combined so that a single system concept could be more thoroughly developed. Below is the Mission Objective along with the system level Requirements, Architectural Design, and Concept of Operations (R/A/C).

-Mission Objective:

The Mission Objective for Team NASACAR is to use the MOLE 1.0 robot as a foundation for creating a modular robot (MOLE 2.0) capable of performing multiple functions viable to a lunar mission as well competing in the 2012 NASA Lunabotics Competition.

-System Requirements:

The main system requirements, shown in Table 1, were split into two categories: requirements derived from summarizing the Lunabotics Rulebook [1] and the Mission Objective requirements not pertaining to the competition. The requirements are labeled according to their requirement type: **F**unctional, **P**erformance, **I**nterface, **V**erification, and **O**ther.

Table 1: System Requirements

Lunabotics Requirements
F: The robot must collect, transport, lift, and deposit the lunar regolith simulant.
F: The robot must be either autonomous or teleoperated.
P: The robot must collect at least 10kg of regolith in 10 minutes.
P: The robot must lift the simulant at least 0.5 meters above ground level.
V: The robot will meet all system requirements by April 30, 2012.
O: The total project costs must be less than $20000.
O: The total mass of the robot with an excavation module must be within 80kg.
O: The robot must fit into a .75m width X 1.5m length X 0.75m height footprint.
Remaining Mission Objective Requirements
I: The robot must be modular. That is, it must be capable of supporting multiple modules that perform different functions and are easy to install/remove.

-Architectural Design:

The team determined that the MOLE 2.0 design would be broken down into three main subsystems: Base, Module, and Data Processing. The Base and Module subsystems were also broken down into three identical subsystems: Mechanical, Power, and Controls. The Data Processing subsystem was composed of two main subsystems: Front End and Back End. The Architectural Design Product Hierarchy can be seen in Fig. 4. This layout differed significantly from the MOLE 1.0 architecture in two key aspects.

The first major difference was that the Base and Module subsystems contained their own independent power and controls subsystems. In the previous design, an Electronics Module subsystem was used as a central hub for all the power and controls hardware. This change was brought about to help simplify electrical and controls interfaces and to allow for more efficient placement of power and controls hardware.

The second major architectural difference was that the robot software was abstracted into its own Data Processing subsystem. This allowed for a clearer perspective of the software layout as a whole.

-Concept of Operations:

The Concept of Operations (ConOps) for the robot will be largely determined by which modules are being used. This paper focuses on the ConOps of the excavating modules for the competition. The system level ConOps derived straight from the competition rules. The only system level choice left to be made was whether the robot would be autonomous or teleoperated.

Due to the large amount of points being given for both full and partial autonomy, the team carefully considered these options, weighing the pros and cons and simulating different point outcomes. After several discussions the team concluded that a fully autonomous robot, while possible, would likely be out of the scope of this year's project; however partial autonomy (autonomous traversal of the obstacle area) should be within reach. In order to achieve this goal and to keep open the possibility of full automation the team set a subset of members to focus on researching methods of automation. The resulting system level ConOps can be seen in Fig. 5.

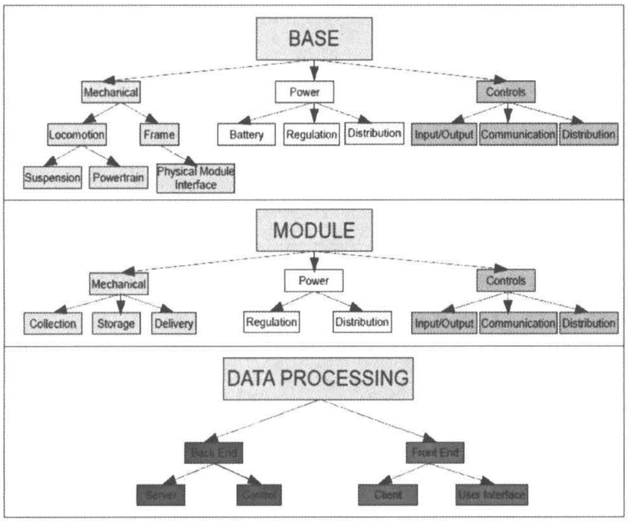


Figure 4: Architectural Design Product Hierarchy

-Base:

Throughout the design process of Team NASACAR's 2012 competition Base, several improvements and concepts were considered. While many other forms of locomotion were studied in the design of MOLE 1.0, a unanimous decision was made to keep the locomotion method from the previous design which consisted of four wheels with the ability to sweep into other driving positions. This feature gave the MOLE 1.0 robot its omnidirectional characteristic, but its main advantage was that it provided the robot with the ability to perform a "neutral turn" (where the center of the turn coincides with the center of the wheel base) with almost no wheel slippage. Wheel slippage is a key factor on the moon since the top layers of the regolith have very low densities. If too much slippage occurred, especially under a load, then the robot would be in danger of bottoming out and being unable to

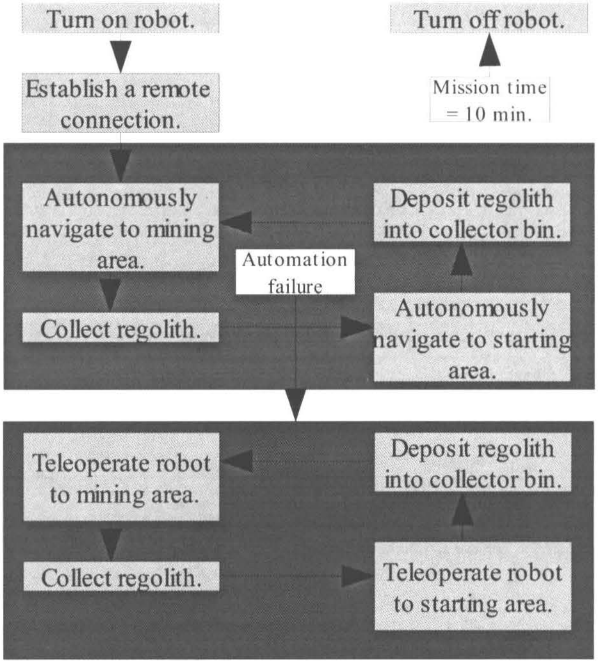


Figure 5: System Level Concept of Operations

move. The same turning ideology is currently being employed in many of NASA's robots including the Mars Rovers, the Centaur 2, and the Space Exploration Vehicles (SEVs). These can be seen in Fig. 6.

Looking back at the MOLE 1.0 Base design, several weak points were found and cataloged and improvements were ranked by priority. The top priority was to increase the torque and power available to the wheels so that the Base could handle larger and more varied modules. On top of this, the Base dimensions needed to be optimized to the 2012 competition rules so that the new goals could be met while still leaving as many options for Module design as possible. Once these criteria were decided upon, a list of requirements was constructed. The Base mechanical requirements can be seen in Table 2.

-Module:

The cost of transportation is a major constraint on colonization and exploration of the moon and other planets. It is much more cost effective to have a system that can be easily modified for a variety of missions once in operation on the lunar surface than to have multiple specialized systems. This section will describe the design process for Team NASACAR's 2012 competition Module. Throughout the process a balance between innovative solutions and proven processes was sought. Taking advantage of insight from our sponsors and team members' experiences with working at earth mining operations, the team explored ways to adapt earth mining processes to the specific challenges faced by NASA on the moon.

Each team member was asked to bring forth

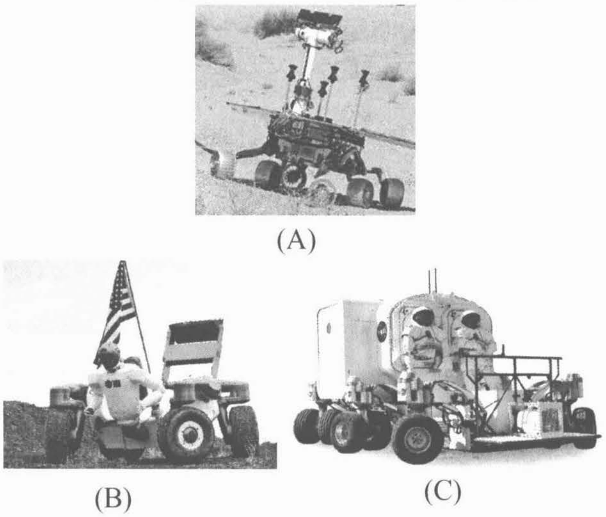

(A)
(B)
(C)

Figure 6: Examples of NASA robots that configure their wheels for zero slippage neutral turns: (A) Mars Rover. (B) Centaur 2. (C) SEV.

Table 2: Base Requirements

Base Mechanical
F: Will use provided Power and Control commands to enact the appropriate physical response.
Locomotion
F: Must provide contact area large enough to support Module and Regolith.
P: Must support full load of Module and Regolith.
P: Must provide locomotion to accommodate as many module designs as possible.
I: Must mount to Base Frame System.
Suspension
F: Will support the Base Frame and keep entire Powertrain in contact with lunar surface.
F: Must be comprised of passive elements.
P: Must support 250kg+ load.
I: Must connect Powertrain to Base Frame.
Powertrain
F: Will provide locomotion of entire Base system.
F: Must run on provided 18V or lower power grid.
P: Must not stall under 250kg+ full load.
I: Must mount to Suspension system.
Frame
F: Will support all Base and Module Systems.
P: Must be able to support 200kg + Lunabot mass.
P: Must have minimal deflection under full load and locomotion.
Physical Module Interface
I: The Module must mount to T-slotted aluminum 8020 via appropriate fasteners.
I: Must be stiff enough so that the module load does not warp the base chassis.
Power
F: Must be disconnected upon drawing >500A of current.
F: Must have a red kill switch that cuts power.
F: The box must be properly protected from lunar environment.
F: The box must be protected from EMF interference.
F: The base power system must provide a way to monitor the power consumption of the battery.
P: Batteries must be able to supply a minimum continuous voltage of 18.5V and a minimum constant current of 125 A.
P: Batteries must weigh <3Kg.
P: The box must be completely stable while running.
P: The base power system must be able to supply a smooth, continuous current to all electromechanical devices.
I: Must have a quick means of connecting to any module.
Controls
F: Must provide wireless communication to the Front End.
F: Must accept feedback from Base wheel positions.
F: Must accept feedback from battery voltage.
F: Must accept feedback from current measuring circuit.
F: Must provide control over Base motors.
P: Must be able to support back end processing requirements.
I: Must communicate with module via interface.

ideas in which lunar soil could be collected and transported. This phase presented a variety of processes, however not all of them were feasible for our team to implement. Also during this time, the Module team leaders were tasked with providing requirements for whichever concept that would move into Phase B. The Module requirements can be seen in Table 3.

After significant discussion and concept visualizations using Solidworks, the team agreed that the optimal excavation approach was to have a digger which fed a center storage hopper and a rear conveyor to offload the material. This was a prevalent method in the teams concepts and one that has been demonstrated in this competition and in real mining processes to be an efficient method of transportation and delivery.

The main decision remaining in this Module concept was the type of digger to be implemented. The submitted concepts were narrowed down to four designs to collect and deposit lunar soil into the onboard storage. With each design thoroughly explained, the team proposed a set of weighted metrics that could be quantitatively applied. Once the designs were examined under each of these metrics, they were ranked competitively against each other and the overall lowest numerical value was chosen as the teams' design.

Table 3: Module Requirements

Offloading
P: Must have a maximum unloading time of 1 min.
P:Must not cause significant moment of inertia changes due to the process alone.
P: Must keep center of mass inside base profile without use of counterweights at all times.
Storage
P: Must keep static profile inside base dimensions.
V: Initial load capacity of 2 times the base mass. To be adjusted pending base testing.
Digging
P: Must have a digging time of 1.5 min.
P: Must not cause center of mass to move outside of base at any time.
F: Must have the ability to collect compacted material.
F: Must have the ability to implement a percussive/resonant process to aid in collection.
Power
F: Must safely supply power to all module electronics and mechatronics.
P: Must be small enough to be mounted to the module itself.
I: Must be able to accept the power from the base power supply.
Controls
F: Must provide video feed and other feedback to the Base module.
F: Must control module motors and actuators via commands received from the Base module.
I: Must receive commands from the Base via a module interface.

The concepts considered were given codenames to help separate the design from the designer. It should be noted that the codenames reflect the "spirit" of the concepts. The "Aardvark" concept was a dual bucket wheel excavator. The "Puppy" concept was a bucket-chain system augmented with an oscillating percussive digger. The "Starfish" concept was a rotating blade/paddle that would feed a conveyor system, and the "Dolphin" concept was a percussive shovel that could be driven forth to feed a conveyor system. The digger decision matrix can be seen in Table 4. As one can see, the Aardvark design was the standout among the group. The Aardvark, Puppy, and Starfish concepts at the time of discussion can be seen in Fig. 7.

-Data Processing:

The purpose of the Data Processing subsystem was to provide the user control over the robot hardware and feedback on the robot status. To achieve this objective, the subsystem was further broken into two sections: the Front End and Back End. The Front End would consist of a user interface to accept user input and display robot status. The Back End would react and

Table 4: Digger Decision Matrix

	Weight	Aardvark	Puppy	Starfish	Dolphin
Moving Parts	0.2	1	4	3	1
Time to Completion	0.5	3	1	1	4
Simplicity	0.7	3	4	2	1
Weight	0.2	3	3	1	2
Penetration	0.8	1	2	4	2
Digging Modes	0.9	1	2	3	3
Rate	1	1	3	2	4
Percussion	0.3	4	2	3	1
		8.3	**11.7**	**11.5**	**11.9**

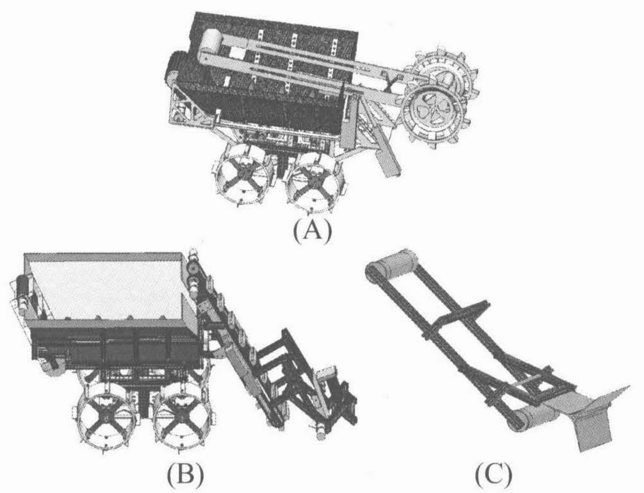

(A)
(B)
(C)

Figure 7: The Aardvark (A), Puppy (B), and Starfish (C) digger concepts.

respond to requests sent from the Front End while running background tasks to keep the robot hardware within its structural limitations. The overall design follows a basic client-server architecture in which the Back End represents the server and the Front End takes the place of the client. The key advantage of using this architecture is that time critical tasks, such as controllers for coordinated hardware movement, can be placed behind the network latency of the wireless link. The client-server interface also provides the option for different client implementations, such as an autonomous Module. The Front End and Back End requirements can be seen in Tables 5 and 6 respectively.

While the points awarded for autonomy in this year's competition were significant, the team decided to focus primarily on creating a robust, effective teleoperated digging platform. Since the dimension and mass requirements for the bot shrank from last year's competition, the team needed to completely redesign the Base. This cut down on the testing time for any autonomous control systems dramatically; thus there would not be enough time to implement full autonomy into the contest entry.

However, the team decided that it would be prudent for a subset of members to research autonomous methods that could be applied to partial autonomy for this year's competition or full autonomy for future competitions. To this end, several promising methods were conceptualized including: IR LED triangulization, Internal Measurement Unit (IMU) with a PID controller, image processing of the Lunarena environment, and simple beacon tracking.

Table 5: Front End Requirements

Primary Controls
F: Must provide access to drive motor direction and velocity.
F: Must provide access to change wheel sweeping position.
F: Must provide access to control module motors and actuators.
Primary Feedback
F: Must provide feedback of motor currents and velocities.
F: Must provide feedback of actuator positions.
F: Must provide feedback of battery status.
Debugging
F: Must log all correspondence with the robot to a file.
P: Must have a high level of fault tolerance, in particular feedback parsing.
P: Must provide connectivity debugging tools.
Other
F: Must have the ability to execute either client or server side motion macros.
F: Must have the ability to show desired (setpoint) values vs actual (reported) values.
F: Must send 500ms period heartbeat signal when control input is inactive.

-System Definition Review (SDR)

At the end of Phase A the team met with the Project Manager for the SDR. The Manager concluded that the system requirements were defined and formed the basis for the proposed conceptual design. He confirmed that the system architecture was adequate, and the ConOps aligned with the mission requirements. After the SDR the team was cleared to advance the project to Phase B. [4]

Phase B: Preliminary Design

In Phase B the subsystem designs went through several iterations and trade studies were performed comparing different components and parts. The goal at the end of Phase B was to have a design that met "all the system requirements with acceptable risk and within cost and schedule constraints and establish[ed] the basis for proceeding with detailed design" [3]. The subsystems evolved concurrently whenever possible. Verification plans for each subsystem were developed.

-Base:

-Mechanical:

In Phase B team members were encouraged to come up with preliminary Base mechanical designs that met the criteria while incorporating the desired improvements over the previous design. Three designs were submitted and compared. The first two options were modifications of the MOLE 1.0 Base design and mostly comprised of retrofitting larger/stronger drive motors while simplifying some of the complex frame. The third option was a complete redesign that

Table 6: Back End Requirements

Server
F: Must accept commands from the client on how to drive the robot.
F: Must drive the robot's hardware based on commands received.
F: Must be able to stop the robot when commanded by the client.
F: Must provide feedback on the robot's status to the client.
F: Should provide the amount of energy consumed to the client
P: Must use less than an average of 5 Mbps during a run.
P: Should react to user commands with no noticeable latency.
P: Should provide low level hardware control to the client.
P: Should provide high level task automation to the client.
I: Must communicate with the client over the wireless link.
I: Should have a well-defined protocol for communication.
O: Should be able to safely handle dropped connections.

incorporated the core mechanics of MOLE 1.0 while simplifying the overall frame and mechanical design, offering several methods of sweeping the wheel positions, and incorporating the new requirements from the outset. The benefit of the first two designs was that they had fewer unknowns in the design, as most of the design had already been tested in the previous competition. While the redesign had very few knowns, it showed a greater potential. After much debate, the decision was made to finalize the complete redesign.

Afterwords, the method in which the wheels were to be swept was considered. The proposed methods involved using actuators (like what was used in the MOLE 1.0 Base), a rack and pinion design, and a worm gear driven design. The worm gear method was ultimately chosen after a closer look revealed that the rack and pinon would be too complicated, and the actuator method would have 50% of the sweeping range due to the reduced space of the smaller Base. Fig. 8 shows the MOLE 1.0 actuator method compared to the MOLE 2.0 worm method.

The final major decision for the Base mechanical subsystem was design of the competition wheels. While the MOLE 1.0 wheels proved to be very robust and effective, they weighed a total of ~12kg. This was obvious area for improvement due to the large penalty for robot mass at the 2012 competition.

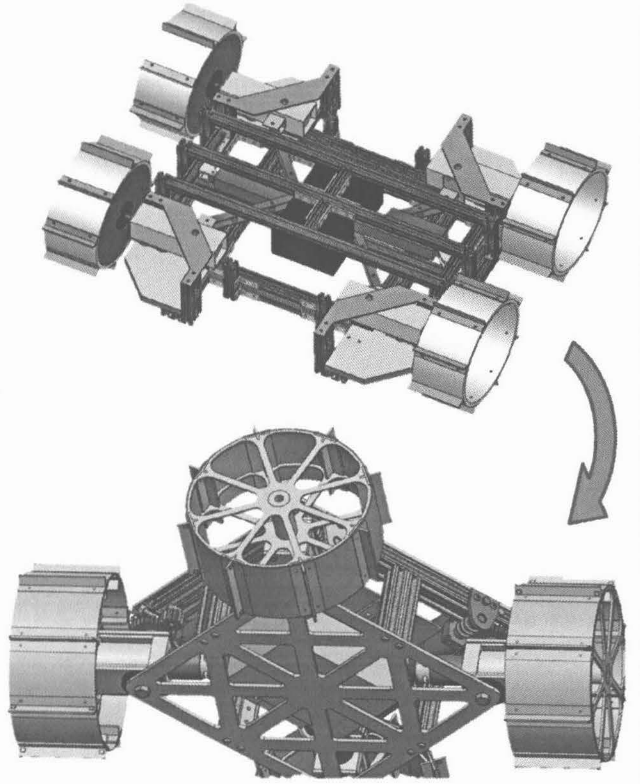


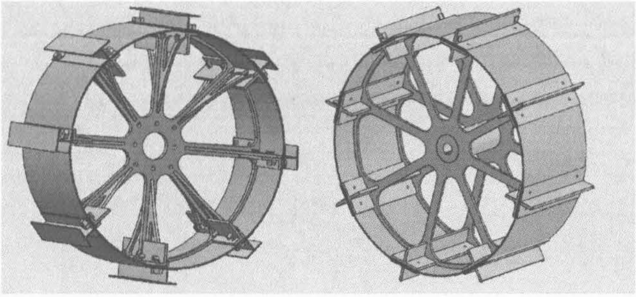
Figure 8: The MOLE 1.0 (top) used linear actuators to sweep the wheels while the MOLE 2.0 design (bottom) implemented a worm gear method.

The task of redesigning the wheels was delegated to a subset of the Base mechanical team, and two competing designs emerged. Both designs consisted of an internal hub structure supporting a fiberglass surface and "paddles" mounted to the surface for traction. Wheel designs one and two can be seen in Fig. 9. After some discussion and comparison of the designs through trade study, the team decided to move forward with option 2. The wheel decision matrix can be seen in Table 7.

Each component of the Base mechanical system would be tested to verify functionality. The mechanical subsystems could be tested as they were assembled. For example the worm gear sweeping assembly could be tested independent of the Base frame or locomotion. Finally a Base subsystem test would be implemented by driving the Base with and without a mass load and with different wheel positions.

-Power:

The Base power system of MOLE 2.0 received a major revamp from the MOLE 1.0 design. This was due mainly to the increased power requirements of the



(A) (B)

Figure 9: Wheel option 1 (A) and 2 (B).

Table 7: Wheel Decision Matrix

Criteria	Wheel 1	Wheel 2	Advantage
Weight	1.2kg/wheel	1.35kg/wheel	Wheel 1
Complexity	Complex due to custom spokes and mounting of cleats.	Very simple, very easy to build.	Wheel 2
Cost	More expensive than Wheel 2 due to additional machining charges.	Inexpensive, mostly prefab parts.	Wheel 2
Structural Integrity	Strong enough to hold expected loads.	Strong enough to hold expected loads, but more robust than Wheel 1.	Wheel 2

larger motors. The general topology of the power system can be seen in Fig. 10.

Noting the success of MOLE 1.0's electronics shielding system, the team decided to use a similar enclosure. This would consist of an aluminum box with a removable lid. The main advantage of using an aluminum enclosure was due to its heat sinking characteristics. The heat generated by the power components would be quickly transferred to the box, which in turn would tranfer the heat to the aluminum Base frame where it would dissipate.

For MOLE 1.0 the team used Sealed Lead Acid (SLA) batteries to power the robot. These batteries have the advantages of being cheap and easy to use. However, these batteries could not supply the voltage and current that the MOLE 2.0 required, and they were extremely heavy. After examining alternatives, the team settled on Lithium-Polymer (LiPo) batteries. These batteries are able to create a much greater voltage and current source than SLA batteries, but at a cost: they are much more prone to explosions and fire. In order to use these batteries, individual overcurrent protection would be required for each battery.

Part of the power requirements was to be able to measure the robot's power consumption. This can be done by monitoring both the current flowing out of the batteries and the battery voltage. The simplest way to measure current is through a shunt resistor circuit which simply provides a measureable voltage reading across a low resistance calibrated resistor. By measuring the current and plotting it against the battery voltage over time, the total power consumed can be determined.

The Base power subsystem would be tested incrementally. Each component would be powered individually to make sure no issues arrised. The components would then be integrated into the electronics box and the whole subsystem would be checked for shortages. The power subsystem would then be checked during Base subsystem tests and full system tests to ensure that heat dissipation was adequate and batteries were stable.

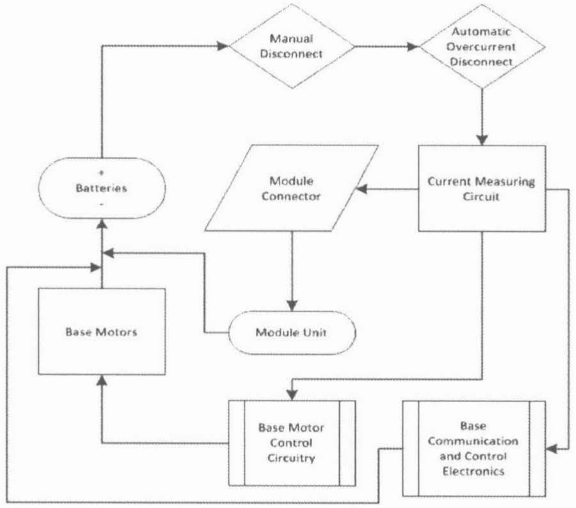


Figure 10: Base power system topology.

-Controls:

The basic controls requirements for the MOLE 2.0 Base subsystem were very similar to the MOLE 1.0 system. The main area that the team saw room for improvement was in the Base/Module controls interface. In the MOLE 1.0 system, each feedback line provided its own individual connector. To interface a Module many connections had to be made. However the decision to separate the Base electronics from the Module electronics provided the opportunity to greatly simplify this interface into one connector.

Besides the Module interface change and the offloading of the Module controls hardware, the Base controls hardware design was nearly identical to the MOLE 1.0 Electronics module. This included a router for wireless communication, a Single Board Computer (SBC) for processing, motor controllers for driving and sweeping, and feedback electronics to control the wheel positions and read the battery and shunt voltages. A Base controls flowchart can be seen in Fig 11.

Each component of the Base controls subsystem would be tested independently as the were acquired to verify functionality. The Back End software would be installed on the main controller to ensure the hardware met processing requirements. The motors and motor controllers would then be assembled as an independent

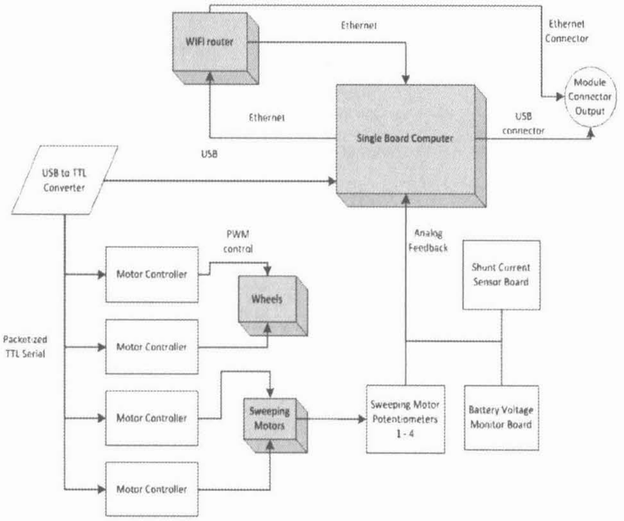


Figure 11: Base controls flowchart.

system to verify control capabilities. The wireless network would then be configured and connected to the main controller to verify remote capabilities. Finally the full Base controller subsystem would be assembled and integrated to the Base subsystem to verify complete functionality.

-Module:

-Mechanical:

The mechanical design of the regolith excavation subsystem consisted of three additional subsystems: Offloading, Storage, and Digging. Each of these three mechanical subsystems had its own independent restrictions and design process. These design processes are described below.

Offloading:

The offloading sub-system consisted of a belt, its supporting structure, tensioning system, and drive system. The main design restraint on this subsystem was the angle of the offloading conveyor relative to the height of the cleats on the belt. Normally, cleated belts are run on flat pulleys due to the tension changes that occur when a cleat runs over a crowned pulley. Unfortunately, these belts must be run at slower speeds due to the guides that are required to track a flat belt. These slow speeds and friction were deemed unacceptable for this design, so a compromise was made. The drive system would have to be able to support the torque changes needed to run a cleated belt over a crowned pulley. In order to limit the amount of torque required, a ½ inch cleat height was chosen. Using design guidelines in [5], the team decided that a belt angle of 45 degrees would optimize offloading. Once initial framework was designed, it was a concern that the optimal angle did not allow for much height over the lunabin. With this in mind, the framework was designed to support five other height/angle combinations to ensure the system could effectively offload the regolith.

To test this design, graduated amounts of simulant will be added to the belt at different angles to measure the maximum transfer rate that can be achieved.

Storage:

At previous Lunabotics competitions, the most effective means of storage seemed to be a central hopper with a single conveyor belt at the bottom to assist offloading. While the conveyor belt has been proven viable for transferring stored regolith to the offloading system, the team decided to design around another industrial corner piece: the screw conveyor. The reasons for this choice were threefold: size, simplicity, and mass. The system would need only one motor and two mounting points and lightweight polymer screw conveyors were available commercially. The only downside to this design was that a screw conveyor inherently would not be able to move 100% of the material from the storage bin to the offloading belt. This was accepted as a cost for the simplicity of the design and work began to move forward.

The subsystem will be tested by running the motors at varying speeds with no off loading belt attached to see how efficient the conveyor and storage bin is by itself. The offloading conveyor will then be added and the test will be re-run with the offloading belt running at varying speeds to see if the two subsystems are truly independent of each other.

Digging:

One of the major design constraints on this subsystem was the conveyor belt angle. It was decided in the beginning of Phase B that the same belt design would be used in all belts to save design costs and to standardize spare parts. With the chosen belt, the optimal angle was below 45 degrees. If the digger were actuated directly into the ground then the belt would have to be over 65 degrees in order to fit inside the starting dimensions. While the bucket wheel excavator could operate at this angle, it would not be as efficient. It was therefore decided to actuate the excavator and belt assembly on a four-bar linkage setup. This way a much longer belt could be used and stored inside the storage hopper before the run. This longer belt allowed the operating angle to be lowered considerably. The four-bar linkage was then put under the following restrictions: 8 inch stroke actuators would be used to allow a 15 inch bucket wheel to reach a depth of at least 7 inches below the surface; the system would be most sensitive in its compacted state; the system could not physically invert; and the system would be designed to have as little lateral movement as feasible in any state. The final design constraint was that the bucket wheel would be designed so that it would be most efficient when it was at a depth of two inches. This would allow for the first and second passes to collect the most regolith as possible. Mechanically, the bucket wheel was designed to dig up to two inches per pass, but expectations are to run at one inch of depth per pass to prevent failure during competition.

The bucket wheel will be tested by powering the motors at different rotational speeds, transverse speeds, and depths to see at which the maximum digging rate can be achieved. Once this is determined a full subsystem test can be qualitatively examined to determine overall efficiency. The bucket wheel can then be modified to attempt to increase in-system efficiency.

-Power:

The team decided in Phase A that the Module power system should be supplied directly from the Base power system. The Module electronics box would be constructed similar to the Base electronics box, using aluminum to sink any heat generated into the Module frame to be dissipated.

The Module power system will be tested incrementally. First each component will be tested to verify functionality. Then the components will be integrated and tested to verify there are no shortages. Finally the components will be tested as a full system to verify that the components don't overheat under expected loads.

-Controls:

The Module controls subsystem was relatively simple to design since all the processing was done on the Base. Commands received via the Base/Module interface would simply be routed to their corresponding device. One of the main design changes between the MOLE 1.0 and MOLE 2.0 Module controls hardware was the way the vision/camera system would be operated. In MOLE 1.0 each camera plugged directly into a USB hub where it was then routed to the SBC. The SBC controlled which camera was activated and would stream its data on a single port. This resulted in a ~5 second delay when switching between cameras since the cameras had to be activated/deactivated. To rectify this issue the team decided to use IP cameras so that the streamed data would bypass the SBC and route directly to an IP address. This way all the cameras could be on at the same time and switching between them would rely solely on the Front End. This would also allow for multiple views to be seen at the same time. A Module controls flowchart can be seen in Fig. 12.

The Module controls subsystem would be tested in the same manner as the Base controls subsystem. The main difference would be the additional testing of the IP cameras and the verification of the Base/Module controls interface.

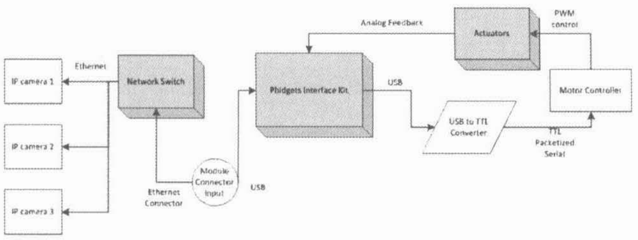


Figure 12: Module controls flowchart.

-Data Processing:

-Automation:

Implementations of the automation methods conceived in Phase A were detailed and compared against each other to see which was best suited for this application.

To implement the IR triangulation method, a circle of IR LEDs would first be mounted to the top of the lunabot. To track this LED array, two IR phototransistors mounted on servos would be placed on the lunabin. A microcontroller would then read off the servos' angles and triangulate the lunabot's position. A simple diagram can be seen in Fig. 13. Although sound in theory, this method would be difficult to implement effectively due to the required accuracy of the servo readings.

The IMU + PID method would be relatively easy to implement since it would only require wheel encoders and an IMU device. With these components working in tandem the lunabot could follow a series of straight line paths. However while this approach would work on a smooth, flat surface, it would run into a key implementation problem: there would be no way to deal with sudden impulses to the instruments such as collisions with obstacles, therefore an unacceptable amount of error would accumulate.

There were several different implementation options for the image processing method. One option was to use a Hough transform to detect obstacle shapes. Rocks would have a downward shape, craters a

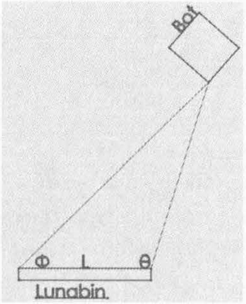


Figure 13: Global positioning through IR traingulation.

different shape, and walls a third. The downfall of this method was the variability of the obstacles to be detected since round rocks would be detected differently from jagged rocks, etc. Another option was to use a simple convolution matrix for edge detection. This would be simple to process and would be a quick way to detect any obstacle. A picture of this operation can be seen in Fig. 14. This method looked promising, but still required much more time to develop. A recurring problem with any image processing scheme is the similarity of the terrain. When viewing the arena through a camera, everything is the same color: gray. This makes it difficult to detect distinct features for even human observers, let alone computerized algorithms. Images and data taken from this year's competition will be very beneficial as test images for future image processing schemes.

There were two implementation options considered for simple beacon tracking, both of which used LEDs placed on the Lunabin and onboard IP cameras. In one option the LEDs would be configured as three array clusters that would form the vertices of a triangle. The beacon could be detected by the cameras using a simple color detection algorithm. Based on the size, rotation, and skew of the array in the camera view the robot could determine its location in the Lunarena relative to the Lunabin. The second beacon option consisted of a single LED array placed at the center of the Lunabin. This option could be used solely for partial automation, as the robot would simply center the beacon in the camera view to keep its orientation as it traversed the obstacle area. Since this was the simplest method of providing partial autonomy the team decided to move forward with its implementation.

-Front End:

For this year's client software, design specification centered around three major objectives. First and foremost, the software needed to provide an intuitive user interface during the competition runs. Within this primary concern were three constituent regions of interest. Obviously, any client software needed to implement direct access to control functions exposed by the robot's server. In addition, the client software needed to be able to combine common functionality into "macros" to automate common tasks. Finally, previous years' experience suggested that the program needed to provide extensive tools for logging vital signs, anticipating exceptions, and debugging communication issues.

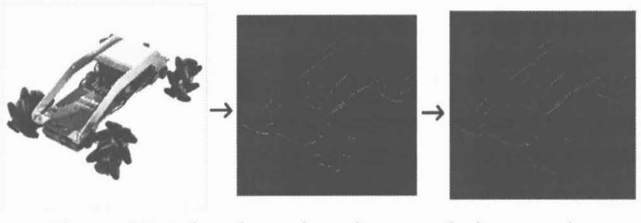


Figure 14: Edge detection via convolution matrix.

As a second objective, this year's client needed to center around a flexible and extensible core framework. Primarily, this would allow easy implementation of an autonomous controller for next year's platform. As an added benefit, a flexible framework would allow the creation of backwards-compatible classes to control last year's legacy system.

Finally, the team wanted to maintain the ability to host an educational website for the robot. This year the team constructed an outreach page that allows students to operate the robot via the internet, and the team wanted to expand this capability for the future. The confluence of these factors, in addition to the easy availability of the Visual Studio IDE, recommended a solution developed for desktop and web in C#.

Since software is often the last testable phase of the design, the software leads decided to take advantage of Visual Studio's automated testing and profiling toolset. A separate testing project was created that could perform unit testing on the individual classes and methods, as well as run through the high level functionality of the robot. Integrated testing and validation with hardware would be performed in Phase C to verify functionality with specific components. Front End functionality with the full system will be validated and verified in Phase D.

-Back End:

The Back End was designed for two main tasks: providing a layer of abstraction around the low level hardware control and reacting to requests sent by the client (e.g. the Front End). This would require a standard TCP/IP server through which the client could send requests and a multitude of control loops monitoring and controlling various aspects of the robot hardware. Because the robot was designed to be modular, a secondary requirement was that the Back End should be easily extendable by the introduction of additional modules.

Ultimately, the Back End architecture chosen closely followed that of the previous year's lunabot due to the success of its operation, the similarity in system requirements, and the potential for code reuse. Commands received through the server would be passed to a object-oriented-like hierarchy with a general robot object at the root. The robot object would consist of various modules which would, in turn, consist of peripherals and electronics. A command passed to the tree would be parsed at each node and passed to the correct sub-component until reaching its target where it would be executed. A response would be directed back up the tree and finally sent through the server back to the client.

The object-oriented architecture would allow independent testing of each component of the system. In Phase C each hardware component could be individually integrated to verify compatibility with the server. The Back End would be tested with the full system in Phase D.

-Preliminary Design Review (PDR)

At the end of Phase B the team met with the Project Manager for the PDR. During the review the Manager expressed his concern for the proposed schedule based on the complexity of the subsystem design choices as well as the risk involved with the technologies chosen. However, he confirmed that the designs met the system requirements, the technical interfaces were consistent with the overall technical maturity, and the risk mitigation plans were adequate. After the PDR the team was cleared to move on to Phase C. [4]

Phase C: Final Design and Fabrication

In Phase C each subsystem was finalized down to its components and parts, the Module interfaces were specified, and the coding for the software was adapted to meet the final design specifications. At the completion of the subsystem designs, materials and components were procured and fabrication began.

-Base:

-Mechanical:

The core of the Base redesign was the implementation of larger, stronger permanent magnet dc motors to drive the wheels. The new motors, combined with their gearboxes, could supply up to 80 ft-lbs of torque, compared to the 35 ft-lbs provided by the MOLE 1.0 motors and gearboxes. Each of the Base motors were sealed with Cuben Fiber fabric to prevent regolith simulant from damaging them.

The corner supports of the newly optimized frame used the stiffness of the new motor gearboxes as a structural component, while incorporating the worm gear system of rotating each drive motor assembly about a shaft to perform the required sweeping maneuvers. On top of this, the wheels designed in Phase B were pulled in much closer to the new frame, thereby reducing the operating track width to within the minimum width dimensions of the competition rules. The rest of the frame was simplified down to straight beams and perpendicular supports so as to maximize the available load capacity and minimize frame deflection under load. All of this combined to reduce the static and dynamic moment that the frame and drive motors needed to support for any given load. The central core of the frame was left open to house the proper power systems and electronics while still protecting them from physical harm. Finally the design met the met the Module interface requirements by incorporating T-slotted Aluminum 8020 extrusion into the upper frame for Module mounting purposes. A size and design comparison of the MOLE 1.0 and MOLE 2.0 Base subsystems can be seen in Fig 15.

-Power:

For the Base power system the team chose to use four 18.5V 5Ah LiPo batteries in parallel. To isolate the power source from the system in case of a short or current overdraw, a 150A circuit breaker was installed between each battery and the current-limiting solenoid that implemented the manual shut-off circuit. 120A Anderson Powerpole connectors provided an easy way to interface each battery with the rest of the system.

To power the Base electronics, 5V switching regulators were used for the Base router and 12V switching regulators were used for the main controller. A 10A fuse was installed between the shunt and the regulators to protect the electronics from power surges.

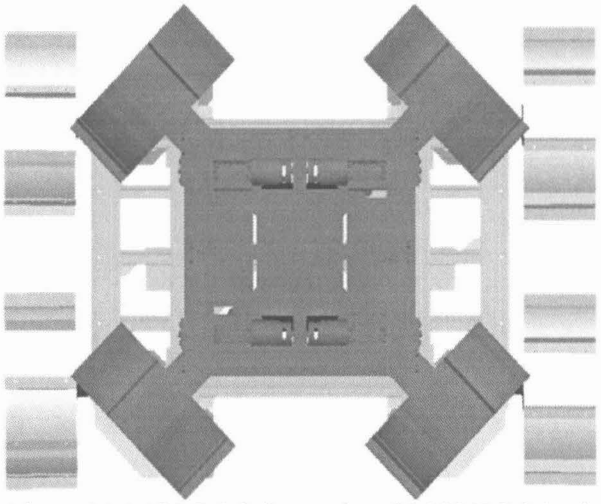


Figure 15: MOLE 1.0 (bottom) and MOLE 2.0 (top) base size/design comparison.

Two 60A motor controllers were used to power the wheel motors and two 25A motor controllers were used to power the sweeping motors. The motor controllers were fed power directly from the output of the shunt resistor. A 120A Anderson Powerpole connector was attached to wires extending from the electronics box to provide the power interface to the Module. The final Base power schematic can be seen in Fig 16.

Initially, the team planned to use a common differential amplifier to measure the current coming from the battery. The voltage drop would be amplified by an op amp, outputting a safe voltage for the onboard computer. While this would be the simplest solution, the system's accuracy would depend on the values of the many resistors. Accordingly, another solution was searched for. Eventually the team settled on combining the differential voltage feature of an op amp and the variable beta gain of a transistor in creating an amplifier that took in a voltage drop and output a current. This system would use fewer parts, and would be more accurate. Even better, a commercially available chip was found that met specifications, making installation much easier.

-Controls:

While the controls layout of the MOLE 2.0 Base was similar to MOLE 1.0, all of the components were either changed or upgraded. The team decided to continue to use a Phidget SBC as the main processing unit, however it was upgraded to the newer SBC2 because of its faster processing speed, better operating system (Debian vs. a custom distro), and larger amount of USB ports (6 vs 4). The router was changed from a Linksys WRT54G running a custom installation of DDWRT to a smaller, less power consuming Buffalo WHR-HP-G300N with DDWRT factory installed.

The motor controllers in MOLE 1.0 were all Phidget HC motor controllers that connected directly to the SBC via USB. This provided a very convenient way of accessing them with the Back End software using the Phidgets API. However these motor controllers could not supply the current required by MOLE 2.0's upgraded motors (they were only rated for 14A while the stall current for the motors was 130A) and were only rated for a 12V system. The motor controllers chosen this year were Sabertooth 25A and 60A controllers which could support from 6V to 30V. Since these controllers did not directly support USB, a single USB to Serial chip was installed between the SBC and the motor controllers. This way each controller could be accessed via Serial addressing.

In the MOLE 1.0 system the wheel sweeping positions were measured via analog feedback provided by the linear actuators. In MOLE 2.0 the team decided to construct a worm gear driven sweeping system. This meant that a custom wheel position feedback system was required. To implement this a potentiometer was attached to a small nylon gear and mounted to the Base frame so that the nylon gear meshed with the worm gear. The potentiometers were powered via the 5V regulator and the feedback fed directly into the analog inputs on the SBC. This configuration can be seen in Fig 17.

As mentioned in Phase B, one of the major design improvements from MOLE 1.0 to MOLE 2.0 was the consolidation of the Base/Module power and controls interface into one connector. The team chose to

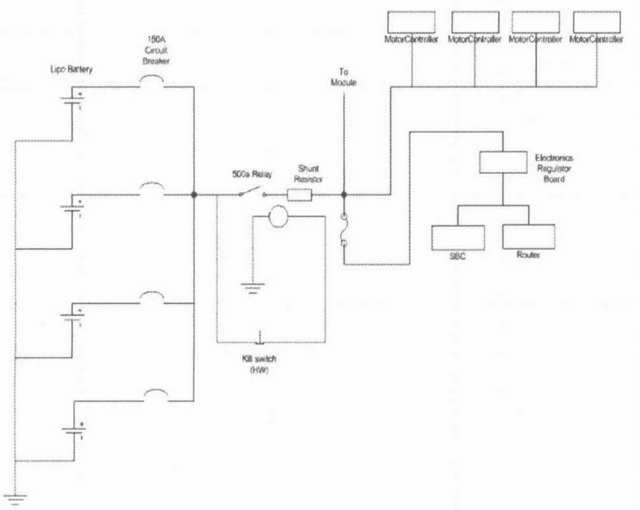


Figure 16: Base power schematic.

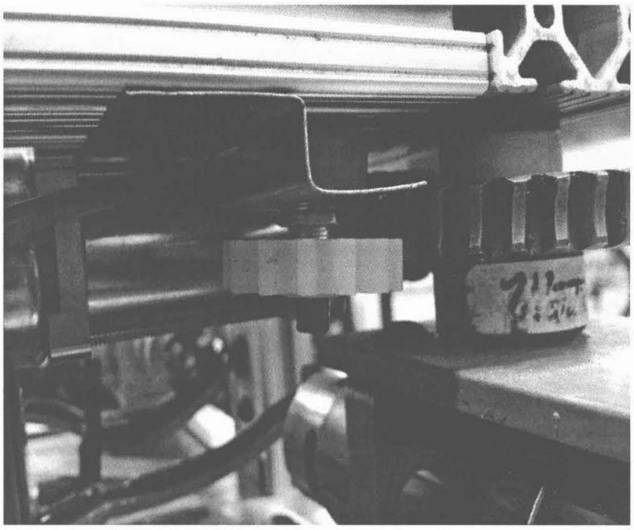


Figure 17: Wheel position feedback system.

use a standard VGA connector since it provided enough pins while being a low cost and simple solution. The connector would provide direct access to the router via ethernet and to the SBC via USB. The remaining pins would handle the kill switch signal and some status LED signals from the digital output of the SBC. A picture of this interface compared to the MOLE 1.0 interface can be seen in Fig. 18.

-Module:

-Mechanical:

Due to the competition mass restrictions and the apparent size of the design of this year's robot, along with the need for as much rigidity as possible, all supporting structure was machined out of 6061 aluminum. This alloy provided a moderately hard structure that would still be easy to weld. It was a much lighter option than steel and could still easily be machined accurately.

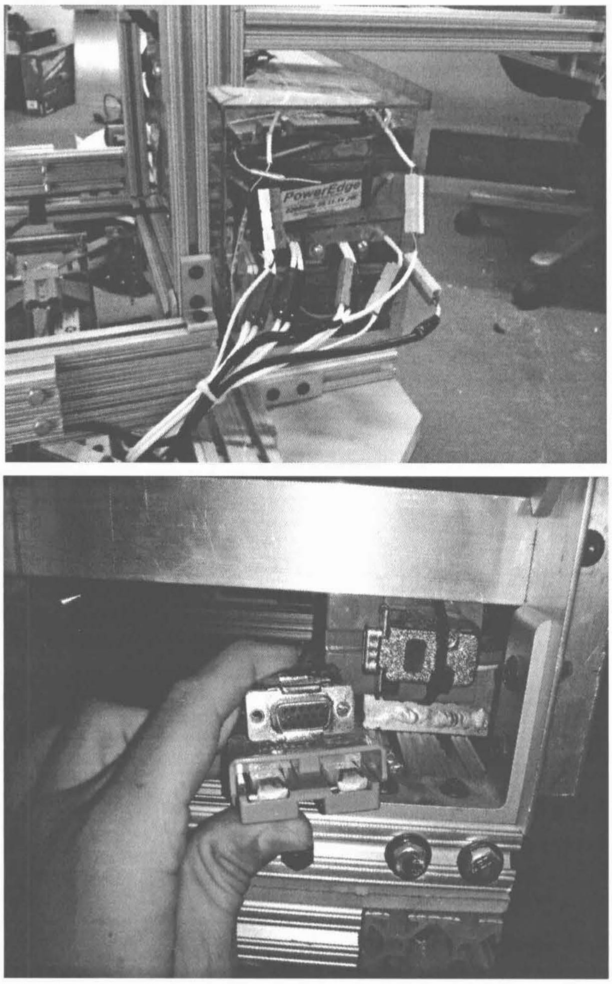


Figure 18: MOLE 1.0 (top) and MOLE 2.0 (bottom) Base/Module power and controls interface.

The Base/Module physical interface required that the Module bolt directly to the Base 80/20 frame. While this would in effect be modular, the team decided to design a quick-release system that would allow the Module to be installed and uninstalled within seconds. This system consisted of four quick-release pins that would attach the Module frame to two 8020 bars that were bolted onto the Base.

Offloading:

The support structure for the rear conveyor comprised of four pieces of machined aluminum. The bottom platform would be made of 1/4 inch plate, while the other three pieces would be machined out of 3/16 plate. Two upper supports would be welded to the bottom plate that supported an alum rod on which the belt pulleys would be mounted. These plates had six slots in which the upper pulley could be mounted and tensioned. The pulleys themselves were made out of four inch polypropylene rod. Each pulley had a 1/8 inch crown and were designed to be as light as possible. The drive pulley would be pinned to the shaft and the sprocket while riding on roller bearings press-fitted into the upper supports. The idler pulley would have roller bearings pressed into both ends.

The Module framework tied all vertical loading into the Base by design. In keeping with the quick-release system, this force transfer was designed to be accomplished by separate triangulation pieces that were not required to be attached directly to the Base. The triangulation and support was accomplished without impeding movement of wheels. Tensioning of the belt was accomplished by allowing the cross-shaft to float on a threaded rod. Using sets of standard nuts, tracking, tensioning, and location of the idler pulley could be obtained. Finally, the upper supports included support for the storage bin. The offloading mechanical design can be seen in Fig 19.

Storage:

Originally, the team was going to create a mold and vacuum form a custom fiberglass hopper. With this in mind, the support structure was designed to clamp onto the fiberglass while distributing the force across a large strip. When the time came to begin the fiberglass production process, a few design changes had been made that made the fiberglass production unfeasible. The team then decided to use 1000 denier Cordura

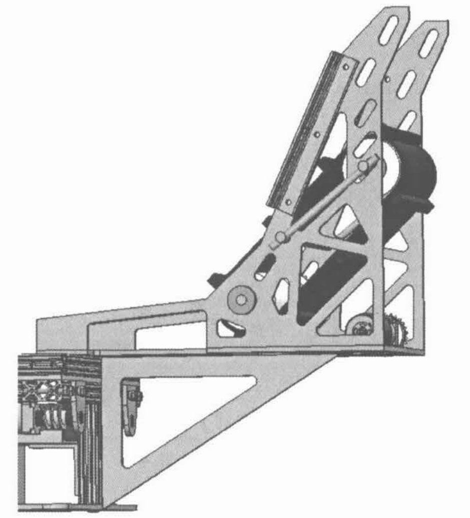


Figure 19: Module offloading mechanical system.

fabric to serve as the main storage unit. This fabric was lightweight at only 14 oz per yard on a 60 inch ream. It was abrasion resistant, tear resistant, had excellent UV qualities, and was fairly inexpensive. Due to the framework design, the only modification that was needed was for an aluminum plate to be formed around the screw conveyor and attached to the frame supports. The pinching design of the fiberglass frame is exactly what was needed for a full fabric hopper.

The screw conveyor is a commercially available polymer conveyor. This conveyor consisted of three sections of polymer screw flights placed on a 2 inch steel square tube. This tube was then attached to solid, hardened steel connectors. This assembly had a mass of over 25 kg. The team disassembled the conveyor and replaced all of the steel components with aluminum in order to reduce the weight to less than 4 kg. This conveyor was supported by a ¼ inch plate and a 2 inch x 1 inch aluminum tube riding on oil impregnated brass bushings.

Digging:

The digging structure is supported by a similar design as the offloading structure with the main difference being the addition of a 1/4" plate that serves as both lateral support and support for the screw conveyor. The four bar linkage is made out of primarily 2 inch x 1inch tubing. The primary concern for the team with the four bar was the presence of slack in the system. With this in mind, focus was placed to add in as much rigidity into the framework and linkages so that a consistent cutting angle could be obtained. The digger support structure can be seen in Fig. 20.

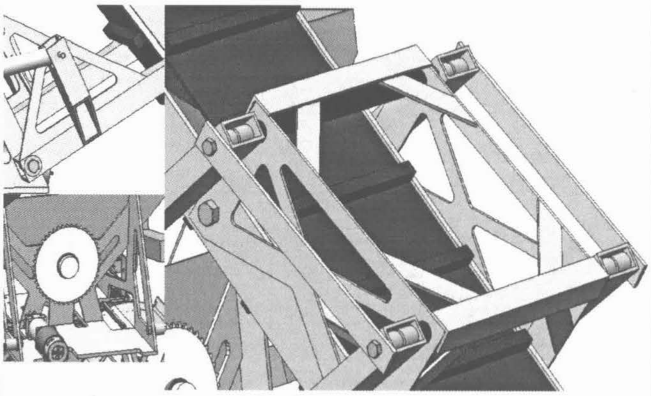


Figure 20: Module digging structural support.

The final length of the belt was 42 inches. The linkage was actuated with dual 150lb actuators with a stroke of 8 inches. The bucket wheels were constructed out of mild steel plate and 4 inch steel thin walled tubing. The tubing was cut into 2 inch sections, then cut in half, creating a curved piece that was welded to the side walls. The total length of these pieces was then trimmed to specifications. In order to excavate in a transverse method, the bucket wheels needed to have some form of "tooth" on the side of the wheel. For this the team decided to build up a weld bead, then grind the bead into a point, creating cutting edges outside the edge of the bucket wheel. The bucket wheels were pinned to a central shaft and driven by a single sprocket.

-Power:

The Module electronics box accepted power from the Base through a 120A Anderson Powerpole connector. This was distributed directly to one 25A and one 60A motor controllers, two parallel 12V switching regulators for the electronics and cameras, and four parallel 12V linear regulators that fed another 25A motor controller. The 12V motor controller powered the two digger linear actuators. These were rated to pull a maximum of 5A of current each. The 12V linear regulators powering the motor controller combined to provide up to 20A of current. This configuration was chosen in order to reduce the heat generated by the linear actuators. The 60A motor controller powered the bucket wheel motor and the rear conveyor motor while the other 25A motor controller powered the screw conveyor motor and the front conveyor motor.

-Controls:

The Module controls hardware consisted of a Phidgets Interface Kit, a USB to Serial chip, an ethernet switch, and three Sabertooth motor controllers. The

Interface Kit was used because it provided an easy way to interface all the peripherals to the SBC and was software accessible via the Phidgets API.

The USB data from the Base/Module interface was fed directly into the USB input on the Interface Kit. The USB to Serial chip on the Module was connected to the USB hub on the Interface Kit and acted in the exact same manner as the Base chip, providing serial access to the motor controllers. All of the IP cameras plugged directly into the ethernet switch, which then passed the data to the Base via the ethernet line on the Base/Module connector. The kill switch line fed directly to the red switch mounted on the Module frame.

-Data Processing:

-Front End:

In order to fulfill the Front End specifications, the software leads decided to create a core library for controlling the robot that would be called from the other user interface options, such as the desktop program, website, or autopilot. This library made heavy use of the Component design pattern to create a generalized abstract tree structure for the lunabot. A diagram of the tree structure can be seen in Fig. 21. This structure allows any of the client accessors to recursively discover the capabilities of future Lunabots which inherit from the abstract tree, increasing our year-to-year flexibility.

The last major Front End detail that was determined in Phase C was the nature of the operator controls. A discussion was held on whether to have a single operator for the whole system (as was used in MOLE 1.0) or to divide the robot controls among two operators. With a single operator there would be no coordination issues, however due to the many moving subsystems the controls would have to be largely automated to prevent overwhelming the operator. With two operators coordination problems would arise and steepen the learning curve, however much finer control over the robot functions would be possible and therefore could be more efficient.

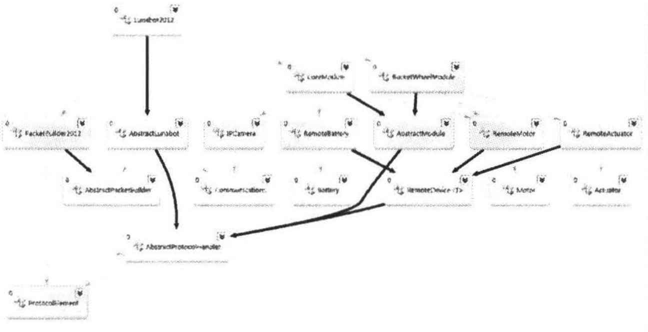


Figure 21: Front End software tree structure.

After discussion the team decided to choose a two operator Front End over a single operator. The determining factor was that the intended operation of the digging system was likely to damage the robot without precise control. Therefore the method that provided the finest control was chosen.

-Back End:

The final implementation of the Back End was a multithreaded daemon written in C and developed to reside on a Phidget SBC2 running the Debian GNU/Linux operating system. After initialization, the main thread executes as a single connection TCP/IP server with a custom application protocol consisting of a module, device, property, and data field. This thread sleeps while waiting for a connection or request. When a request arrives, the thread parses the protocol fields and descends the hierarchy until it passes the data to the correct controller. Responses from the controller, such as hardware status, return back to the server and are sent back to the client.

Each controller of the Back End is a separate thread created during the initialization process. The various hardware components that require constant monitoring or coordination, such as the wheel sweeping mechanism of the Base and the pair of actuators controlling the bucket wheel excavator arm, are constantly kept in a valid state, even while there is no client connected.

In addition to the server and controllers, the daemon also runs a watchdog timer that monitors the server connection for requests. If no requests arrive after a set amount of time and the timer expires, the Back End assumes the connection between the server and client is lost or unstable and stops all hardware movement as a safety precaution.

-Critical Design Review (CDR)

At the end of Phase C the team met with the Project Manager for the CDR. The Manager again expressed his concern for both the complexity of the designs and the large amount of custom fabrication that would be required. However he verified the maturity of the designs and the teams allotment of resources. The teams plan for system assembly, integration, test, launch and mission operations was confirmed to be sufficient to progress into Phase D. [4]

Phase D: System Assembly, Integration, Test, and Launch (SAITL)

The project is currently in Phase D. Each subsystem is being constructed and tested against its requirements based on the verification plans developed in Phase B. Several subsystems have been through iterations of adjustments based on testing results. Phase D will conclude with the successful launch of the MOLE 2.0 lunabot at the 2012 Lunabotics Competition.

-Base Testing:

Each Base subsystem was tested independently with very few issues arising. During the first full Base test all of the basic driving commands were tested and verified on a smooth terrain. After one minute of operation with a 80kg load all the subsystems seemed to function correctly and heat production from the power subsystem was minimal.

More vigorous testing was accomplished at the second Base test. The Base was driven unloaded in a sand environment and tested while driving over rocks and through craters. This test ended when the hot glue holding the noise canceling capacitors between the drive motor leads melted and the capacitors shorted to the motor housing. This showed that the motors did not have enough air flow inside of the Cuben Fiber sealing to cool off. To solve this issue the Cuben Fiber was replaced with a cotton based fabric that was more breathable. The Base main cavity would also need to be sealed in order to prevent regolith from penetrating the breathable fabric.

The next Base test was also held at the sand pit and included the same obstacles while carrying loads of 30kg and 60kg. Around 10 minutes into the test some team members noticed that the wheels seemed to be moving dangerously past their defined sweeping positions. After investigating this issue it was determined that the aluminum shaft and pin that drove the wheel sweeping system was too soft and was beginning to wallow out. To solve this issue the shaft and pin would have to be replaced by steel components.

During this investigation the team also noticed that metal shavings were accumulating below the worm. After a closer look the shavings were determined to be coming from both the worm and wormgear. Because this system was so sensitive, slight errors in the mounting of the worm and wormgear caused the meshing to wear excessively. An image of the wearing can be seen in Fig 22. To solve this the team would adjust the mounting positions and add lubricant to the system after it was sealed.

-Module Testing:

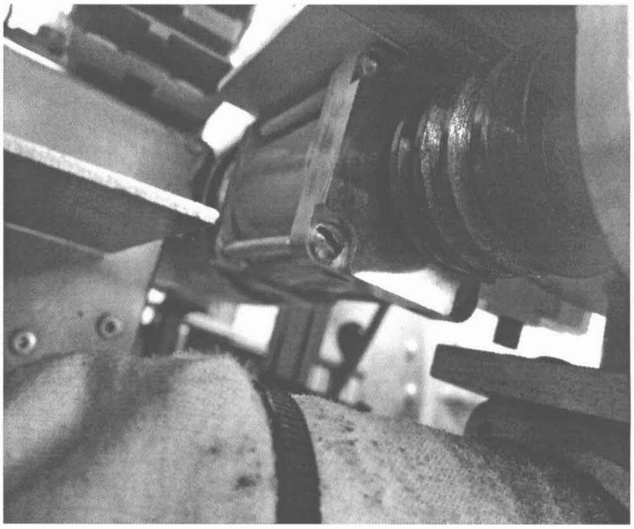


Figure 22: Worm wear and shavings after testing.

The Module subsystems have been tested independently of each other as described in Phase B. All of the individual components function correctly and as an integrated subsystem. The team is currently finishing construction of the Module subsystem, so a full subsystem test has not yet been attempted.

-Software Testing:

The client-server software-architecture was tested early in the project since only a network infrastructure was required.

The Back End was tested using a bottom-up approach. First, "classes" were verified with unit tests to ensure correct operation. Object members were replaced by stub interfaces when necessary until sub-components were tested. Integration testing was conducted on the robot as hardware and electronics were installed to validate the system.

As of this writing, the hardware is not to a point where full system tests can be performed, but the smaller unit tests are performing as intended.

Technical Management

-Technical Resource Budget:

Throughout the project the technical resources required by each subsystem were continuously monitored to ensure that the combined required resources did not surpass the total amount available. The two major resources for this project were mass and power. Other resources included bandwidth and processing speed. The technical resource budgets for mass and power can be seen in Tables 8 and 9.

-Risk Management:

The Lunar Engineering Handbook refers to risks as "undesired events and consequences that can adversely affect the project or mission" [3]. The

Table 8: Power Budget

Component		Allotted	Average Used (10 min)
Base		220Wh	208Wh
	Wheel Motor x 4	190Wh	(15A)(18V)(0.17hr)(4) = 183Wh
	Sweep Motor x 4	30Wh	(2A)(18V)(0.17hr)(4) = 25Wh
Module		150Wh	146Wh
	Front Conveyor Motor	60Wh	(0.17hr)(0.5)(25A)(0.75)(18V) = 29Wh
	Rear Conveyor Motor		(0.17hr)(0.25)(25A)(0.75)(18V) = 14Wh
	BWE Motor		(0.17hr)(0.5)(60A)(0.75)(18V) = 69Wh
	Screw Conveyor Motor		(0.17hr)(0.25)(60A)(0.75)(18V) = 34Wh
Total (4 x 18.5V x 5Ah battery)		370Wh	354Wh

Table 9: Mass Budget

Subsystem	Allocated	Estimated
Base	35 kg	30 kg
Module	45 kg	42 kg
Total	80 kg	72 kg

purpose of risk management is to identify these risks and handle them by implementing a mitigation plan. Lunar excavation is a high risk operation with risks ranging from the abrasive nature of regolith to high solar radiation. Although the robot created in this project will not proceed on a lunar mission, many of the same risks are present. A risk management legend can be seen in Table 10 and a list of foreseen risks and how they were mitigated can be seen in Table 11.

Project Management

-Configuration Management:

Team NASACAR based their project headquarters in an off-campus lab provided by the University of Alabama Engineering Department. All of the tools, materials, and documents were organized and stored in the lab. The team had access to four computers in the lab to work on CAD models, programming, and to store other electronic documents and files. Team collaboration and online file sharing was accomplished through Google Groups, Google Docs, Dropbox, and Subversion.

-Management Structure:

The team assignments were chosen early in the project timeline in order to help direct engineering efforts. Each disciplinary subsystem was assigned one or more team leads and the remaining team members were distributed according to their field of study. Due to the relatively small nature of this project, team members were not restricted to working solely within their assignment. This resulted in many concurrent engineering efforts. A diagram of the management structure can be seen in Fig. 23.

Table 10: Risk Management Legend

Failure Classification		
Code	Name	Description
4	Mission Failure	If this error cannot be mitigated, the mission will be a failure - no communication to the ground station.
3	Reduced Lifetime	If this error cannot be mitigated, the mission is still a success, but further research is needed to extend mission lifetime in future missions.
2	Reduced Capability	If this error cannot be mitigated, the mission is still a success, but further research is needed to provide increased capability.
1	Non-Critical	If this error occurs, the primary mission could still be accomplished without additional need for redundancy.

Table 11: Risk Management for MOLE 2.0 System

Subsystem	Component	Failure	Code	Mitigation
Base				
	Nuts/Bolts/Fasteners	Loose/No connection	2	Thread Locking
	Electrical Connectors	Loose/No connection	3	solder and zip tie connections
	Sweeping Assembly	gear/motor failure	3	lock wheels into skid steering
	Drive Motors	Motor Failure	3	4 Drive Motors
		Regolith Corrosion	3	Sealed Base and Motor Chamber
	Drive Wheel	No Driveshaft Link	4	Thru-Pin with Locks
	Drive Wheel	Wheel breaks		Use 2011 competition wheel
	Battery	Overdraw	4	circuit breakers + fuses
		Bad Connection	2	4 Cells in Parallel
		Low Power	3	Voltage Sensor
	Motor Controller Boards	Stall Current/Overheat	2	Heatsinked Boards
	Electronics Enclosure	Regolith Corrosion	2	IP54 Sealed Enclosure
	Electrical Connectors	Loose/No connection	3	Connector Lock
Module				
	Nuts/Bolts/Fasteners	Loose/No connection	2	Thread Locking
	Rear Conveyor	too low for dumping	4	adjustable heights in frame

-Schedule:

Throughout the project the team kept internal deadlines and goals for each systems engineering phase. After each deadline (whether it was reached or not) the team would re-evaluate the schedule and update the future goals leading up to the competition. The project schedule can be seen in Fig. 24.

-Financial Budget:

The Alabama Lunabotics project was funded from many different sources this year including the NASA ESMD Space Grant Program, the Alabama Space Grant Consortium, and Joy Mining. Each fund was put into a separate account within the Electrical

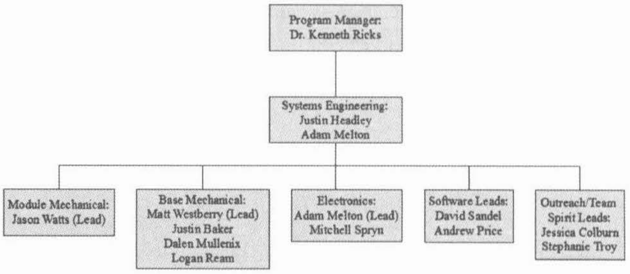
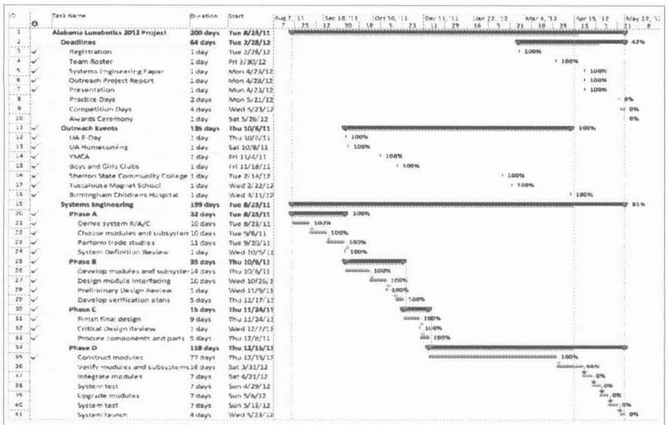


Figure 23: Management Structure



ID		Task Name	Duration	Start
1		**Alabama Lunabotics 2012 Project**	**200 days**	**Tue 8/23/11**
2		**Deadlines**	**64 days**	**Tue 2/28/12**
3	✓	Registration	1 day	Tue 2/28/12
4	✓	Team Roster	1 day	Fri 3/30/12
5	✓	Systems Engineering Paper	1 day	Mon 4/23/12
6	✓	Outreach Project Report	1 day	Mon 4/23/12
7	✓	Presentation	1 day	Mon 4/23/12
8		Practice Days	2 days	Mon 5/21/12
9		Competition Days	4 days	Wed 5/23/12
10		Awards Ceremony	1 day	Sat 5/26/12
11	✓	**Outreach Events**	**135 days**	**Thu 10/6/11**
12	✓	UA E-Day	1 day	Thu 10/6/11
13	✓	UA Homecoming	1 day	Sat 10/8/11
14	✓	YMCA	1 day	Fri 11/4/11
15	✓	Boys and Girls Clubs	1 day	Fri 11/18/11
16	✓	Shelton State Community College	1 day	Tue 2/14/12
17	✓	Tuscaloosa Magnet School	1 day	Wed 2/22/12
18	✓	Birmingham Childrens Hospital	1 day	Wed 4/11/12
19		**Systems Engineering**	**199 days**	**Tue 8/23/11**
20	✓	**Phase A**	**32 days**	**Tue 8/23/11**
21	✓	Derive system R/A/C	10 days	Tue 8/23/11
22	✓	Choose modules and subsystem	10 days	Tue 9/6/11
23	✓	Perform trade studies	11 days	Tue 9/20/11
24	✓	System Definition Review	1 day	Wed 10/5/11
25	✓	**Phase B**	**35 days**	**Thu 10/6/11**
26	✓	Develop modules and subsyste	14 days	Thu 10/6/11
27	✓	Design module interfacing	10 days	Wed 10/26/1
28	✓	Preliminary Design Review	1 day	Wed 11/9/11
29	✓	Develop verification plans	5 days	Thu 11/17/11
30	✓	**Phase C**	**15 days**	**Thu 11/24/11**
31	✓	Finish final design	9 days	Thu 11/24/11
32	✓	Critical Design Review	1 day	Wed 12/7/11
33	✓	Procure components and parts	5 days	Thu 12/8/11
34		**Phase D**	**118 days**	**Thu 12/15/11**
35	✓	Construct modules	77 days	Thu 12/15/11
36		Verify modules and subsystems	16 days	Sat 3/31/12
37		Integrate modules	7 days	Sat 4/21/12
38		System test	7 days	Sun 4/29/12
39		Upgrade modules	7 days	Sun 5/6/12
40		System test	7 days	Sun 5/13/12
41		System launch	4 days	Wed 5/23/12

Figure 24: Alabama Lunabotics 2012 Project Schedule

Engineering department under the Program Manager. The Systems Engineers assumed the responsibility of purchasing the project equipment with the approval of the Project manager, documenting each purchase, and keeping track of all accounts to ensure that the project stayed under budget.

A separate travel budget was created for the competition trip. Travel funding was provided by the University of Alabama Student Government Association (SGA). Expenses not covered by the SGA fund are expected to be picked up by a Graduate School travel grant. A portion of the current project funds has been put aside in case the Graduate School fund does not follow through. The financial budget can be found in Table 12.

CONCLUSION

Using the Systems Engineering approach combined with previous work and experience, the 2012 Alabama Lunabotics team, Team NASACAR, has successfully developed a modular system capable of performing multiple tasks depending on its Module configuration. The current MOLE 2.0 robot can be seen in Figure 25. The final system validation and system launch will be the completion of a successful mission at the 2012 Lunabotics Competition in May, 2012.

Table 12: Summarized Financial Budget

Funds/ ECE Account #	Fund Amounts	Funds Spent	Encumbered Travel Funds	Remaining Funds
ASGC/23665	$7,000.00			
NSGF/30257	$4,000.00			
Walter Resources/30257	$1,000.00			
Joy Mining/30257	$5,000.00			
IECS/30257	$1,000.00			
Zoe's Kitchen/30257	$89.34			
SGA	$2,600.00			
Total	**$20,689.34**	**$15,406.00**	**$4,000.00**	**$1,283.34**

The team expects this design to continue to evolve in the future as it already has from the previous year. If all goes well then the next Alabama Lunabotics team will have the opportunity to develop multiple excavation modules that can be compared against this year's design and each other as well as some non-excavation modules. It is the teams hope that NASA will find this design to be a viable option for future lunar missions.

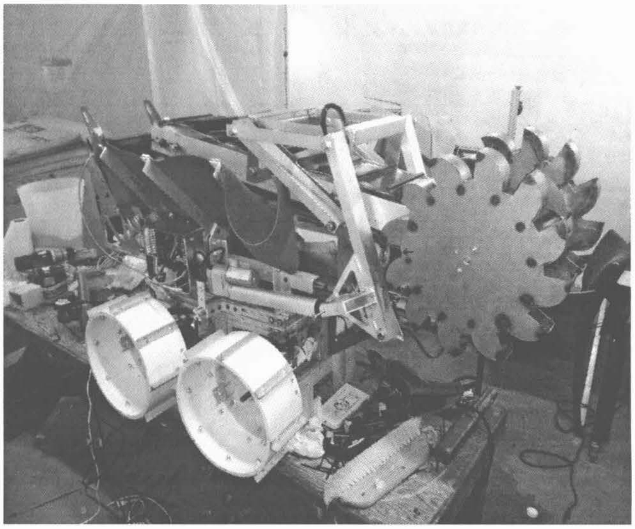


Figure 25: Current MOLE 2.0 Lunabot

Bibliography

[1]NASA's Lunabotics Mining Competition 2011 Rules & Rubrics, 2010Kennedy Space Center http://www.nasa.gov/pdf/390619main_LMC%20Rules%202011.pdf

[2]NASA Lunabotics Mining Competition Website, Susan Sawyer2011Kennedy Space Center http://www.nasa.gov/offices/education/centers/kennedy/technology/lunabotics.html

[3]Lunar Engineering Handbook, Dr. David BealeAuburn University http://education.ksc.nasa.gov/esmdspacegrant/LunarRegolithExcavatorCourse/index.htm

[4]NASA Systems Engineering Handbook,Rev1, ,2007 http://education.ksc.nasa.gov/esmdspacegrant/LunarRegolithExcavatorCourse/Site%20Documents/NASA%20SP-2007-6105.pdf

[5]Shigley's Mechanical Engineering Design, 8th, Nisbettt, J. Keith, Budynas, Richard G., 2006