General research and Problem Statement Generation 17th Feb – 3rd March

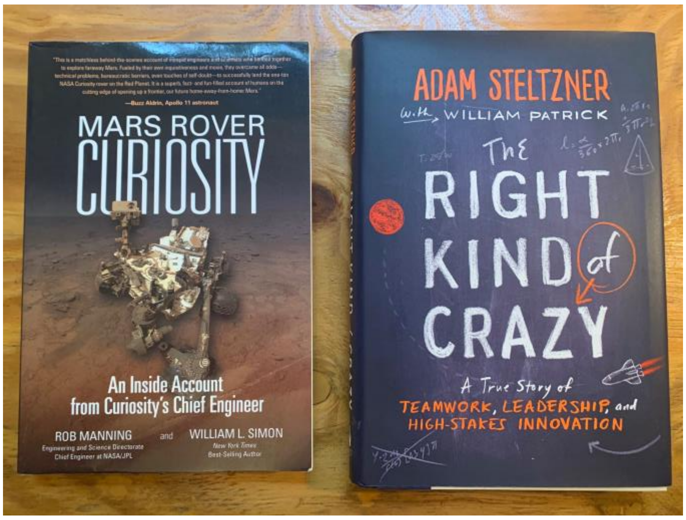
Chatted to our mentor Mitchel: Main takeaway – it's very hard for 1st years to build a rover in ten weeks. Should aim to at least produce 'something'.

Look at earth moving machines around Sydney and online – scoops are pretty prevalent.

Draw concept sketches – how big should it be?

Consult with the team – what do they think? Any resources we already have?

Read some general Mars rover engineering books - Both these books are about the engineering success of the rover Curiosity and the complexity of operating a machine in the Martian environment. Whilst not directly useful as the level of technology is orders of magnitude above what we're dealing with, both are pretty inspiring.



Sweeping roller could be a good idea.

Problem Statement Generation

Considerations

Who?

A group of first year engineering students are required to build a rover in a period of about 10 weeks. The design objective is '…your team is to design and build a prototype rover that is able to transport on a Mars-like surface and collect resources.'

The client is 'NASA' but in relation to the actual problem the client is the course assessor and at least to some degree, our team as we will be the ones operating it.

What?

Objectives:

The objective is to build a rover that can transit on a Mars-like surface (beach terrain with small rocks, sandhills and recesses) and collect resources (small pebbles or gravel).

Constraints:

The constraints are:

Limited skills - it must be built by undergraduate students with little or no experience in building robots.

Limited time: - a working prototype must be produced in ten weeks.

It must also operate in a 3*3m environment so must be small enough to manoeuvre in this space and large enough to carry approx. 1 kg to 5kg of simulated regolith. More is better.

The assessment marking emphasizes the following points:

Our mentor pointed out that from past years the probability of a team producing a rover in the time available is ~50% and the likelihood of producing a working rover is ~30%.

1. *Produce a rover. Binary Y/N*
2. *The rover is able to move about on a 'Mars like surface' and collect resources in the ten-week time period. Binary Y/N*
3. *A high rate of collection. Non-binary but measurable in kg per second.*
4. *Energy consumption – non-binary and measurable as mAh consumed.*
5. *Innovation - non-binary and subjective but related to execution of 1 – 4.*

Key Issues: - Limited time and skills

Action: - Simple - easy to design, build and operate.

Other considerations:

Powerful – able to move about and collect resource quickly. Must be considered with efficiency requirement.

Energy Efficient – no un-necessary systems. Must be considered in balance with power required.

Revised problem statement

As part of their assessment for a first-year engineering program students are required to produce a working prototype of a rover. The rover must be built by the students in ten weeks. It must be able to move about on a Mars like surface (simulated by beach terrain with small rocks, sandhills and recesses) and collect resources (simulated by small pebbles/gravel). The 'Mars-like' area is approximately 9 m^2 and the rover will be required to operate continuously for 4 minutes.

The project will be marked on performance against five objectives. Four are objectively measurable. (in order of precedence)

1. *Produce a rover. Binary Y/N*
2. *The rover is able to move about on a 'Mars like surface' and collect resources in the ten-week time period. Binary Y/N*
3. *A high rate of collection. Non-binary but measurable in kg per second.*
4. *Energy consumption – non-binary and measurable as mAh consumed.*
5. *Innovation - non-binary and subjective but related to execution of 1 – 4.*

Post problem statement exercise the best design will be the minimum viable product needed to move ~5kg of regolith in 4 minutes.

It should be:

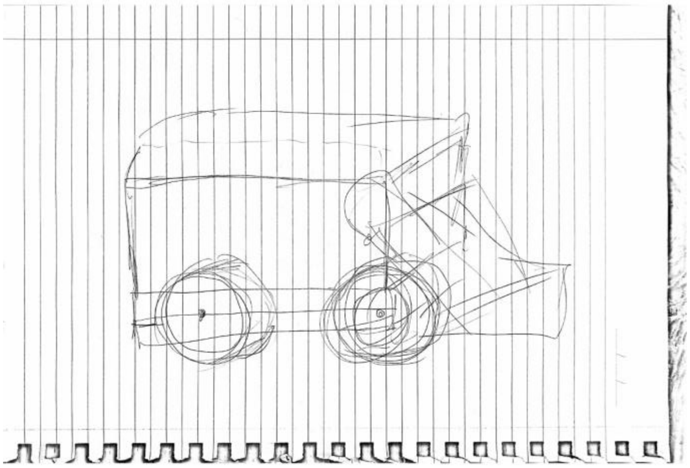
Simple - easy to design, build and operate.

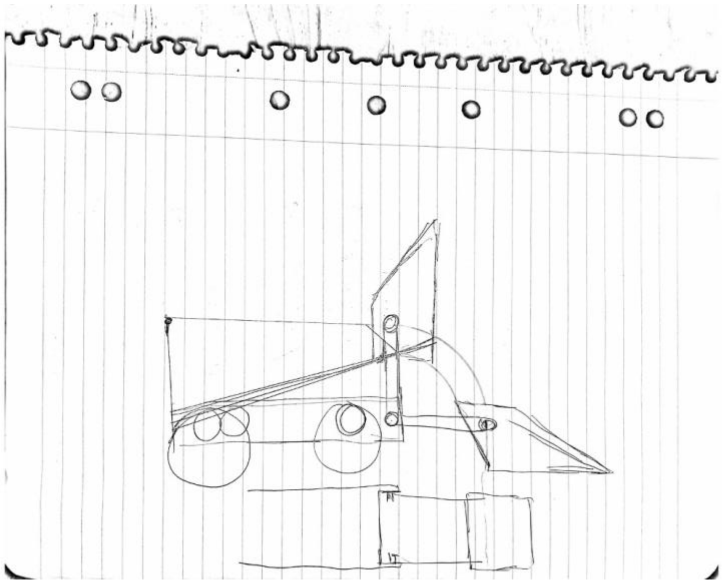
- made from readily available materials that don't require special tools.
- The simplest design possible with the least number of parts.
- Use as much 'off the shelf' electronics and parts as possible to minimize development time.
- Make use of resources we have already to minimize expense and complexity.

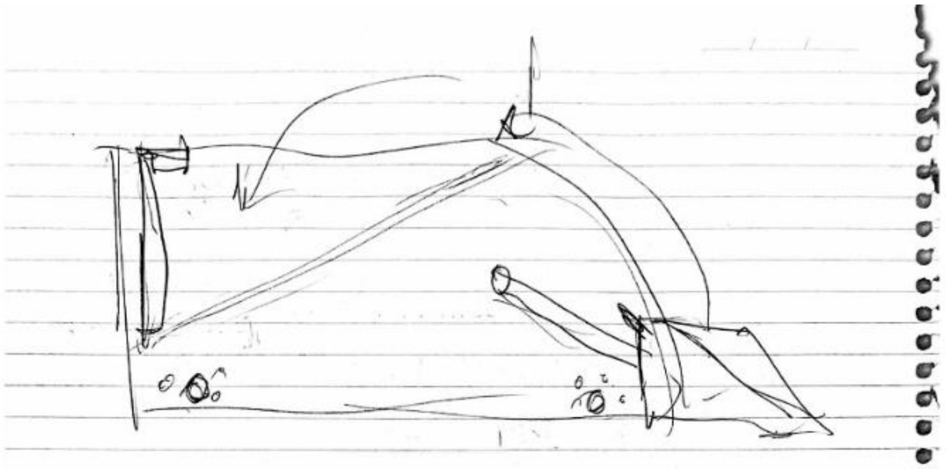
Powerful and large enough to move 5kg of Regolith in 4 minutes.

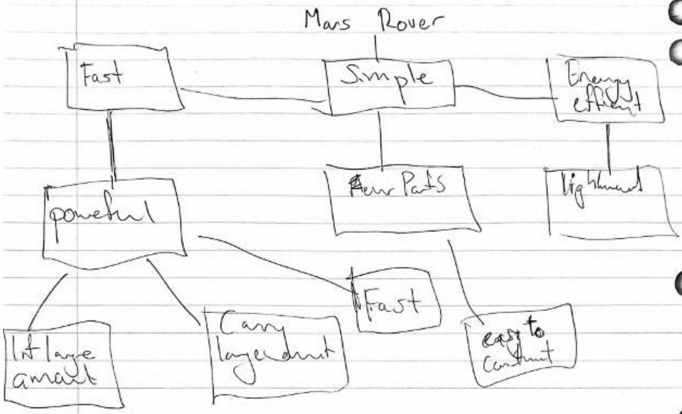
- Use high voltage (12 or 24 volts) instead of 5 volts.
- Be large enough to carry 2.5 - 5 kg at one time to minimize trips.

Initial Sketches and Feature Generation





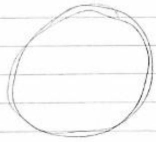




Large enough → Rover ← Small Enough

Carry lots of Regolith

Transit Surface
Enough Inertia
to pick up
regolith



Easy to construct
Not use too
much powers
manoeuvrable

Design Concept Generation March 4th – 15th

Function / Feature	Option 1	Option 2	Option 3	Option 4
Body Material	Plywood	Aluminum Sheet	Fiberglass	Carbon Fiber
Fabrication Method (subsystem of material)	Nuts & Bolts	Glue (Titebond or JB Weld)	Screws	
Power Source	Batteries	Fuel Cell	Solar Panels	Internal combustion
Battery Type (Power source subsystem)	Li-po	Lead-Acid	Ni-Cad	
Voltage (Battery subsystem)	24v	12v	9v	5v
Propulsion (power source subsystem)	DC Electric Motors (Direct Drive)	DC Electric Motors (Indirect Drive)		
Movement	Wheels (Four-Wheel Drive)	Wheels (Two-Wheel Drive)	Tracks	
Steering	Differential steering	Two moveable wheels	Four moveable wheels	
Control Method (Microprocessor)	Arduino	Raspberry Pi	STMicroelectronics NUCLEO-F401RE	
Control Method (Communication)	Radio Control 2.4 GH RC transmitter and receiver	Bluetooth	Wi-Fi	Autonomous
Regolith Collection	Scoop	Horizontal sweeping roller	Vacuum System	Rotating Drum with scoops
Scoop Actuator (Regolith Collection sub-system)	Linear actuator	Servo	Pneumatic/Hydraulic	
Regolith Dispatching System	Gravity fed with gravity opened gate.	“Dump-Truck” – tipping tray.		
Gate Actuator (Regolith Dispatching sub-system)	Solenoid	Linear actuator	Servo	

Morph Chart
Primary principle for feature selection: **Simplicity - easy to design, build and operate.**
Secondary principle: Powerful and large enough to collect and move 5kg of regolith in four minutes.

The design features were selected by following the primary design principle above unless it would seriously compromise the secondary design principle.

Body Material Plywood is really the only serious contender. It is light, strong, cheap and easy to work. We will use 7mm ply for the main body and 3mm ply for the scoop and rear gate.

In the age of composites, ply-wood seems like a backwards choice. However, plywood has many of the advantages of composites and almost none of the disadvantages. Team 9 is silly to disregard it. See link for details - A brief history of Plywood

Fabrication Method – We will use machine nuts & bolts in combination with angle brackets to construct the mechanical components. This allows us the flexibility to modify & rebuild the prototype throughout the build and test process. This is one of the most importance aspects of our design. The ability to develop the prototype during the build and test phases as inevitable issues and technical problems arise and solutions need to be developed. Some of these issues can be avoided by the use of CAD and simulation but built in flexibility is crucial to success. It also greatly improves the ease of maintenance once the prototype is operational.

Batteries are the obvious choice. We have available 6x 22.2 nominal voltage 1450 mAh Li-Po batteries. We will use these because Li-Po's are energy dense, light weight and expensive. Not having to buy batteries is a huge cost advantage. We also have the charger and they are pre-wired with Anderson Power-poles.
See link for Li-Po information regarding Li-Po's - Li-Po information

Battery type and voltage are dictated by our choice of Li-Po's. The advantage of using 24v is that the currents draw is lower. This means lighter gauge wires can be used and losses from heating are reduced. It also means that digital systems running from the main battery can maintain a steady 5V supply even under significant voltage drop from start-up/ stall loads etc.

Propulsion system is obviously electric, and DC as dictated by our choice of battery. We have chosen direct drive from geared down 24V motors with a no-load RPM of 120. Direct drive saves space and reduces the complexity of the design as there are no external gearboxes or axles.

Movement Here we have chosen to accept a slight increase in complexity. Four-wheel drive needs the addition of two motors and draws twice as much current, but gives a large increase in the ability of the rover to deal with sand and carry more Regolith with less risk of getting stuck on Mars. Our system is based on the robot detailed here: 6 Wheeled Robot

Steering Differential steering was the obvious choice here, no complex steering mechanisms (particularly as we have selected 4-wheel drive) and as the solution is software based it is infinitely adjustable to suit driver and robot operating conditions.

Microprocessor the Arduino is the obvious choice. There are dev boards with much higher specs (e.g Pi and the Nucleo) but the amount of documentation, support, libraries and open source code solutions already available on Github etc made this choice for us.

Control method We have a 7 Channel 2.4 GH Radio Control transmitter and receiver already available. Again, this choice means a slight increase in complexity because we need to code the Arduino to process the PWM signals from the receiver this is acceptable because the rover will be much easier to control as the transmitter is designed to control models. There is also a large amount of information available about this application and signal processing. See - RC Arduino

Regolith Collection Initially we were in favour of a horizontal sweeping roller to collect the Regolith as we felt it would offer a faster rate of collection. However, after the problem statement generation exercise and discussions with the team we decided to go with a scoop. The reasons for this were;

1. In all the previous rovers that we saw, no solution included a roller even though we know a lot of teams considered it. This made us think it was harder to implement than it seems.
2. We haven't seen a sample of the Regolith yet. It may or may not be suitable for a sweeping roller system. A scoop can deal with many different types of Regolith, but a roller would be more likely to need customisation to operate with a particular 'spec'.
3. A sweeping roller would need a system to raise and lower the roller arm and a system to spin the roller adding complexity. We are aiming for the simplest solution possible.
4. Scoop are a 'tried and true' system. We know they work from the many examples in earth moving and mining machines and the almost exclusive use of them by previous teams.

Scoop Actuator We have chosen a linear actuator over servos for three reasons.

1. Using a Linear actuator lets us apply force to the scoop arm at a distance from the hinge reducing the load on the system.
2. Most readily available servos run on 5 – 7.5 Volts. We want to use a standard voltage for the whole system to reduce complexity.
3. 24V linear actuators are relatively cheap ($60) compared to 24V servos ($200).

Regolith dispatching system – here we have made a compromise towards simplicity at a reduction in load capacity. A gravity fed system needs no moving parts but requires a triangular profile that cuts the useful volume of the rover in half. We have designed a gate that closes under gravity and opens under the weight of the Regolith. Therefore, the only electro-mechanical system need is a device to keep the gate closed whilst the Regolith is loaded and transported and allow it to open at the dispatching area.
The 'area under the triangle' is utilised for the battery and electronics so it is not 'wasted space'.

Gate Actuator – To allow the gate to open and close we will use a 24 Volt solenoid that when activated allows the gate to open and keeps it closed when shut.

After the feature selection process above we designed the rover in Fusion 360 as detailed below. Simplicity was the overriding design principle.

The design is a box shape with an angled 'floor' running most of the length to form the Regolith container and provide structural support. There are two cross braces at the top of the rear and bottom of the front to make a triangular support structure that is strong and light.

The motors are bolted directly to the side panels. The linear actuator is supported by 6mm stainless steel threaded rod and located in the centre. The rear gate hinge is offset so it closes under its own weight. The solenoid is located on the top brace and prevents the gate from opening.

Some rough Rover dimensioning calculations

Dimensioning the Rover.

Angle of repose: (from Wikipedia)

Gravel (crushed stone) 45°

Gravel (natural w sand) 25° to 30°

No regolith sample yet so making some estimates.

- Regolith is natural (not crushed) so less than 45°
- Mars has less weathering due to very little atmosphere / water so Regolith will be more angular than gravel.
- Therefore choose angle of repose of 35° degrees.

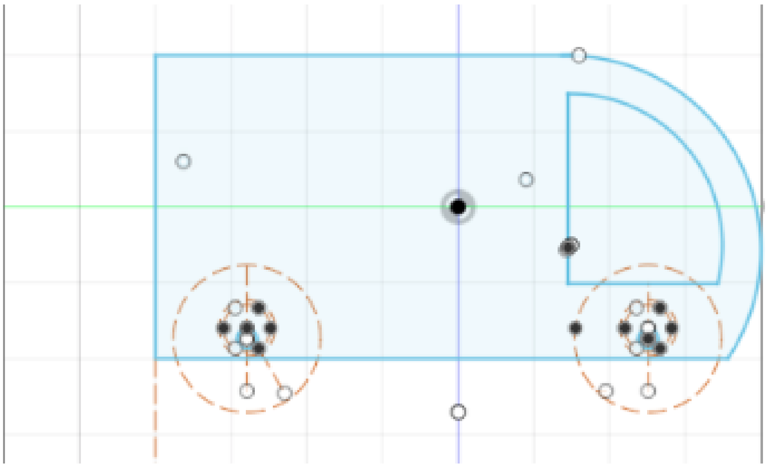
Density ≈ 1.20 $kg \cdot m^{-3}$

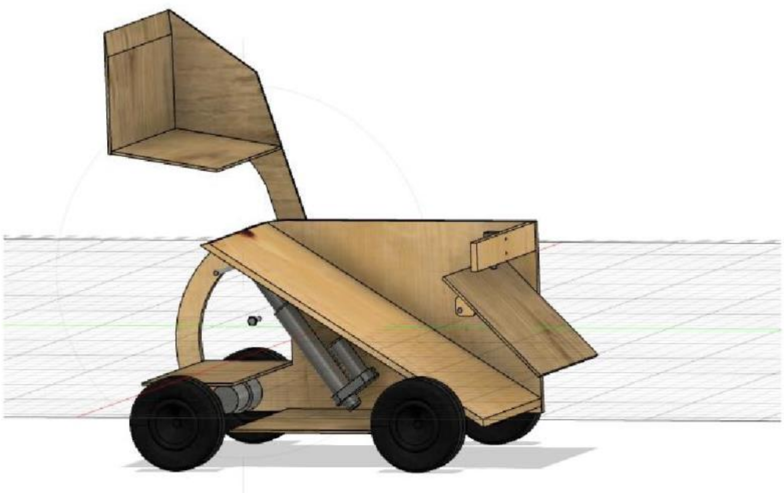
Therefore for a capacity of 5 kg regolith.

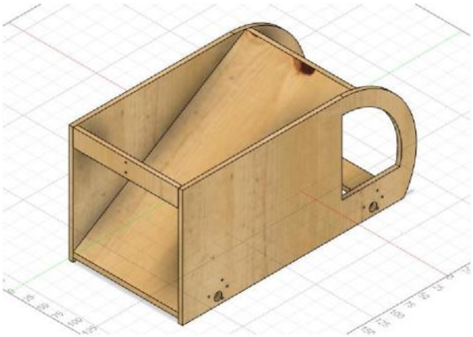
I don't want the rover to be more than twice its length in height or width.

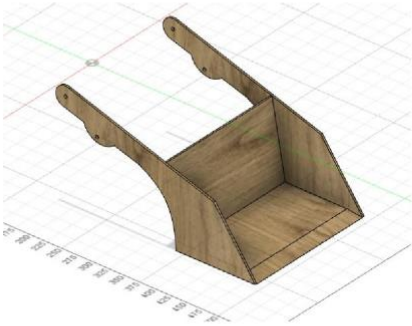
I want 3/4 of it length to be useful load carrying and 1/4 for mechanisms and electronics.

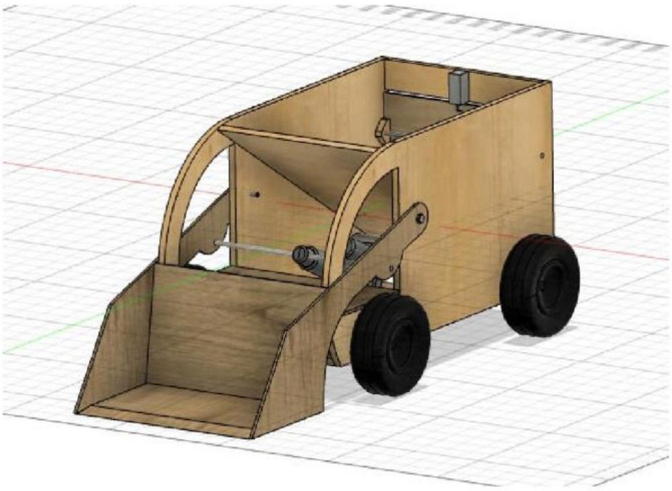
CAD Design images (Fusion 360)











Final Render

Mechanical Build March 16th – 20th

The mechanical build was fairly simple. We encountered one issue when laser cutting the body from ply as the sketch files from Fusion did not dimension correctly in Illustrator. However, the super helpful Lauren at MCIC was able to redraw them for us very accurately. Thanks Lauren!

Video of [Laser cutting scoop arms]

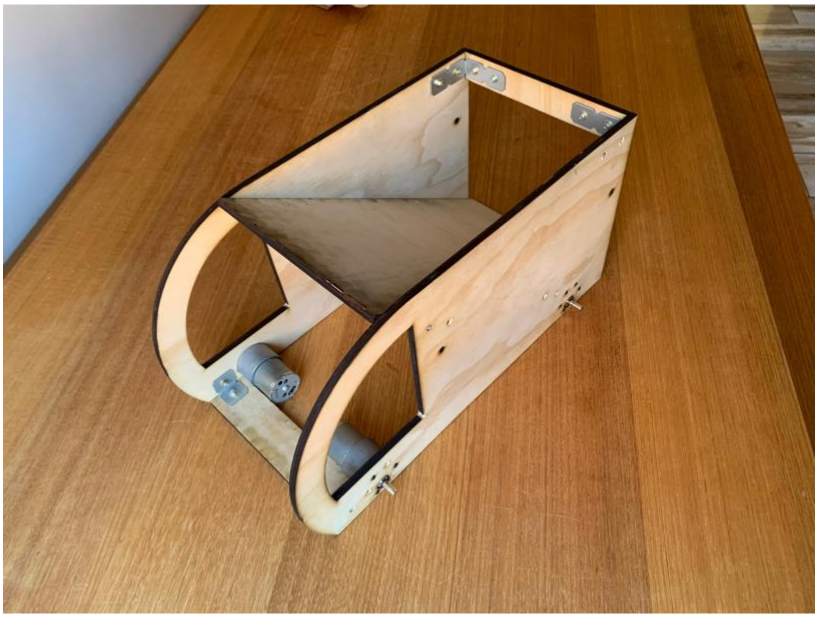
We were also able to drill the holes for the angle brackets using the laser cutter in 'freestyle mode'.

Video of [Rover body from CAD to Prototype]

Purchasing List

No	Description	Required Quantity	Vendor
Main Components			
1	Interior Plywood Sheet 897mm*600mm*7mm	1	Bunnings
2	Interior Plywood Sheet 1200mm*810mm*3mm	1	Bunnings
3	Threaded Stainless Steel Rod 1200mm*6mm	1	Bunnings
4	Angle Bracket 25mm*25mm*40mm	3	Bunnings
5	M5*35mm Machine Bolt	3	Bunnings
6	M5 Flat Washer 35Pk	1	Bunnings
7	M5 Hexagon Nut 20Pk	1	Bunnings
8	M6 Nylon Lock Nut 6pk	2	Bunnings
9	M6 Nylon Stainless Steel Nut 6pk	2	Bunnings
10	M3*15 mm Machine Bolt	3	Bunnings
11	M3 Flat Washer 35Pk	1	Bunnings
12	M3 Hexagon Nut 200Pk	1	Jaycar
13	Mini Brass Spade Connector 100Pk	1	Jaycar

Assembled Main Body



Assembled Main Body and Scoop

Video of [Rover Build Progress Update]

Electro-Mechanical Build March 21st - 27th

In contrast to the construction of the rover main body, the integration of the electromechanical systems has many issues and requires a major redesign of the scoop mechanism and general rover layout.

These stem almost entirely from trying to save on costs by purchasing a non-branded cheap linear actuator. The actuator sourced from (pictured below) is approximately five times cheaper than a brand name item. The main Issue is that it had no data sheet and the vendor could not obtain one. This meant when the CAD model was designed, we assumed actuator dimensions that are about fifty percent too small. The geometry of the scoop operation had to be redesigned with the actual actuator size.

To fit the large actuator, we moved the attachment of the scoop pivot from a metal rod between the scoop arms, to attached directly to the scoop. We also refabricated the floor and back of the scoop in sheet metal because it was flexing by 20mm under the load from the actuator. We also changed the diameter of the actuator pivot from 6mm diameter stainless steel rod to 8mm to eliminate some bending that was occurring during testing.

We also needed to remove the front brace and move the intended location of the Arduino because both was in the way of the actuator.

A few of these issues are also caused by the differences between the CAD models and real-world application. CAD models (or at least the simple model we used) have no friction, perfectly stiff materials, and zero efficiency loses.

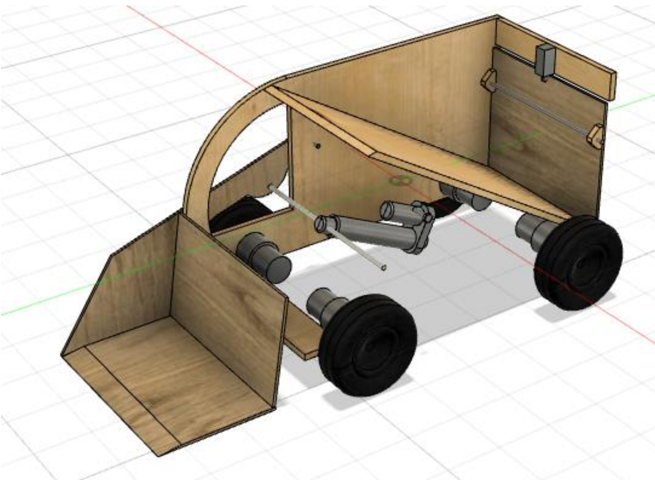
Our main mistake was to try and save money on such a critical part of the system. In particular, the lack of a datasheet meant a lot of the geometry had to be reconfigured when the main body was already built. It would have been better to speed the extra money on the actuator and recoup it on less critical parts.

On a positive note, the motors which we had datasheets for fitted perfectly into the design.

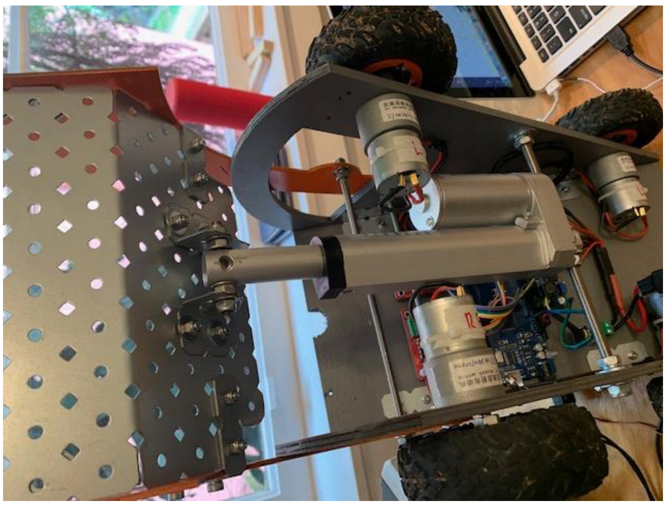
Apart from the extra weight of the actuator, the reconfiguration of the design brought some benefits. The parts count was lowered and attaching the actuator directly to the scoop made the overall structure much stiffer.

Another issue with the linear actuator is that it has no feedback mechanism to tell the Arduino what position the scoop is in. We wanted to use a rotary encoder as one of the scoop arm pivot shafts to provide this feedback. Unfortunately, the components were out of stock and will not arrive in time to be included. We intend to control the position of the scoop by a time reference in the Arduino code. It remains to be seen how successful this will be.

Initial CAD design of scoop and actuator assembly



Scoop as actually constructed after testing



Vendor photo of actuator. There is no size reference.

Electronic System 28th March to 1st April

07 April 2020
14:05

Twenty years ago, the best information was found in books and peer reviewed journals. This is still true. In researching the electronic system, we relied on two excellent resources.

Monk, S, & Scherz, P, 2016, Practical Electronics for Inventors, Fourth Edition, McGraw-Hill Education for general electronics information and layout ideas.

Warren, J, Adams, J, & Molle, H, 2011, Arduino Robotics, Apress for Arduino specific research and code development.

We had a few options when decided how to implement the electronic control system. As per our design principle of simplicity we had chosen an Arduino controlling four 24 volt motors and a linear actuator via three motor controllers. A 24 V solenoid was initially specified to secure and release the rear gate. Unfortunately, the eBay vendor cancelled the order so we replaced it with two servo's we had available.

Due to the COVID-19 related problems shipping electronics from china we had to source many of the electronic components from within Australia leading to budget overruns.

Purchasing list of main electrical components

Electronic Components			
16	MEGA 2560 R3 Arduino komp. Mikrokontroller Board Atmel ATmega2560 CH340G	1	eBay
17	37mm 12V/24V DC 5RPM - 1000RPM High Torque Gear Box Motor Reducer Reversible New	4	eBay
18	Module DC-DC Volt Regulator Arduino Comp	1	Jaycar
19	DC12/24V 20mm-500mm Multi-function Linear Actuator Motor Stroke Heavy Duty xN	1	eBay
20	Spektrum AR6210 2.4G 6CH Receiver	1	Secondhand
21	Arduino Compatible Stepper Motor Controller Module	3	Jaycar
22	Arduino Compatible 5A Current Sensor Module	1	Jaycar

For the electronics layout we considered two options:

1. Pre-constructed Arduino compatible modules wired together.
   Pros – simplest option, quick to implement, easy to test and you know it's (probably) going to work.
   Cons – takes up lots of space and requires more wiring, parts of some modules are not used.

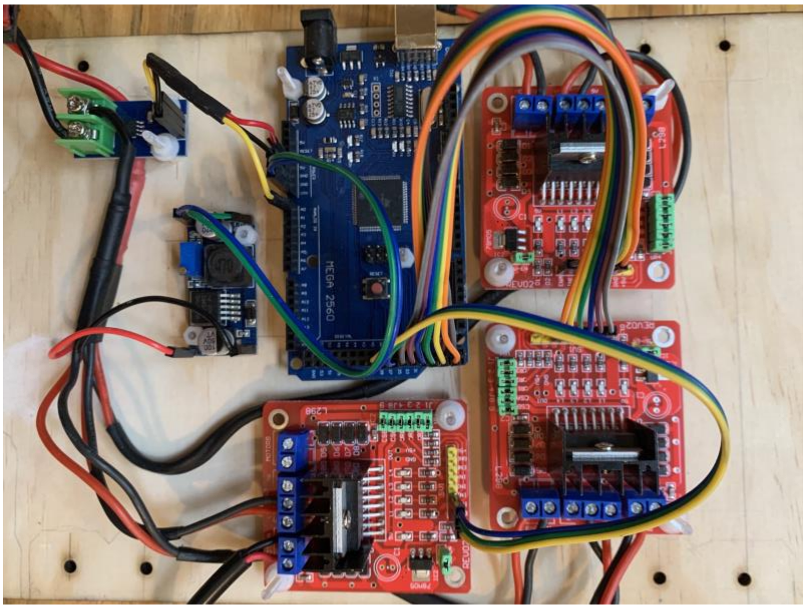
2. Design a separate PCB with Arduino, DC power stepdown, motor controllers on one board.
   Pros – compact and uses less power, only has the components needed, you know exactly how it works.
   Cons – much more work (I.e more time) to implement (designing, fabricating & testing PCB)

We chose option one due to project time constraints and lack of access to PCB manufacturing facilities. If we continue to develop the design in the future, we would like to design a custom PCB to make more space available for other sensors.

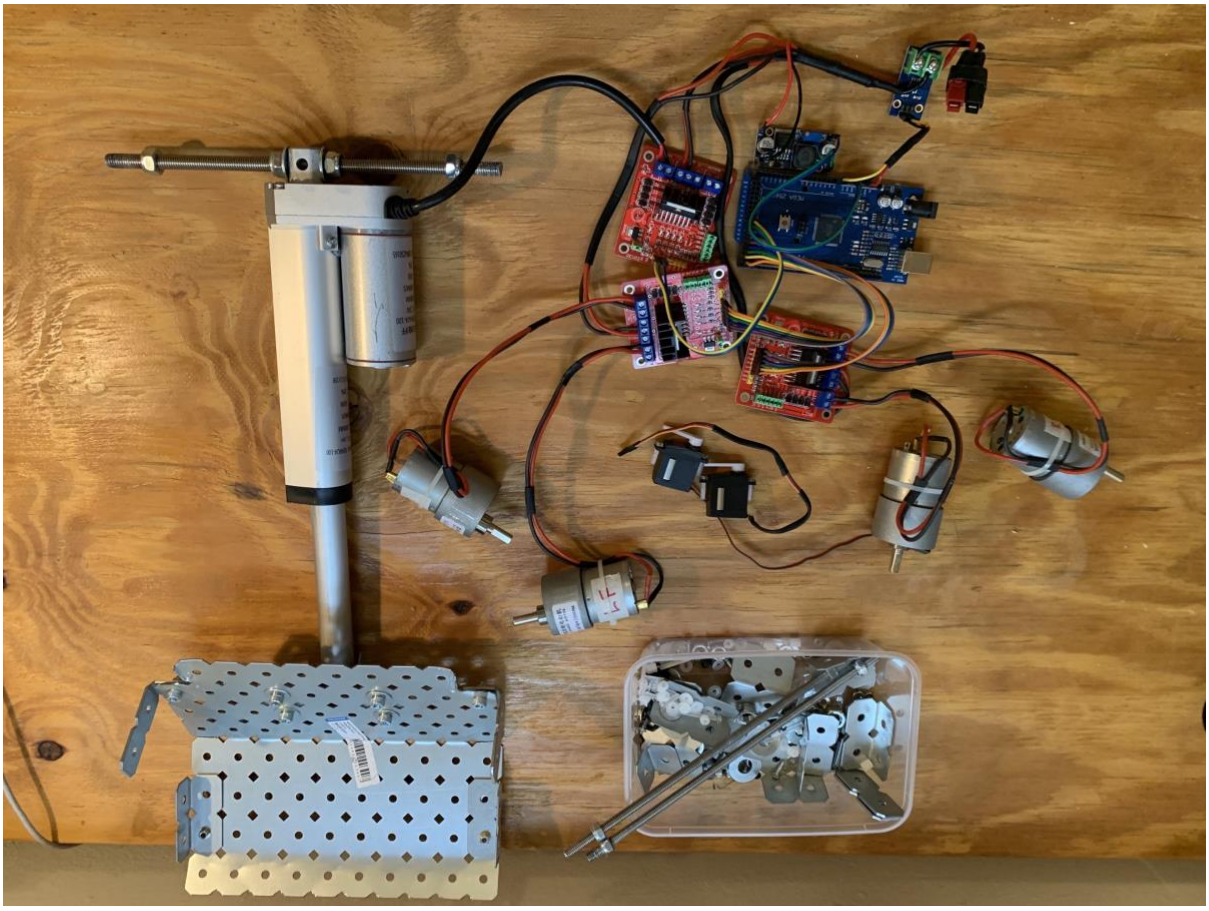
The system has seven components.
1. A receiver to receive (obviously) signals from the transmitter.
2. Arduino Mega microprocessor to process inputs and send outputs.
3. Three L298N Dual Motor Drivers – Left and right motors have one each and one to drive the linear actuator. The actuator driver has one redundant input/output.
4. A DC –DC convertor to provide 12V to the Arduino from the battery.
5. An ACS712 Hall-Effect Current Sensor to measure the system current draw.

The operation of Arduino, DC MOSFET motor controllers and PWM signals is well documented in many resources so we will not re-explain it here. A detailed explanation can be found in Monk et al. (2016) in chapters 13 and 15.

Electronics Layout (receiver not pictured)



Electro-mechanical System removed from main body



Most of the work on the electronic system involved constructing the wiring loom. A lot of time was spent making sure it would be reliable and not compromised by vibration or allow shorts to occur. This was done by heat shrink insulation around vulnerable areas and using nylon fastenings to secure the PCBs to the board.

Testing and Final Assembly April 2nd - April 8th

Once the electronic system was working and integrated with the electro-mechanical system, some testing was done on the regolith collection and dispatch system.

The regolith provided is angular and has a lot of flat surfaces. It was a high angle of repose and does not slide easily. This is difficult for gravity fed systems to handle and we had to adjust the angle of the scoop back to allow it to slide. To lower the friction co-efficient we constructed a coreflute inner on the scoop and regolith container to allow it to slide more easily. In initial testing this worked reasonably well. It is hoped that the vibration and inertia from the moving rover should facilitate this process.

When we were satisfied that the systems were working we disassembled and painted the rover main body components. This was done to increase the aesthetic appeal of the rover as it will be assessed by video. We chose an orange and grey colour scheme as a reference to 'The Martian'. Final assembly was carried out using Loctite on all nuts and bolts (except Nylocs) to reduce the chance of vibration induced failures.

Colour scheme Inspiration

Painting components

99% Complete

Code Development April 9^{th} - ongoing

The Arduino code development was done in two main steps.

First, the motor controllers were tested with a simple motor demo code that did not have any real time input. This allowed us to test the controller and Arduino were working correctly and become familiar with the outputs required.

One issue with the linear actuator motor controller was identified. The controller had power and the Arduino was switching the inputs correctly, however the controller was not supplying power to the motor output. After a few hours of testing everything independently we discovered the cable to the enable input of the controller had an intermittent fault that tested fine, but when put back in the system was not sending the PWM signal to the controller. This did not show up on the controller as the switching LED's do not required the enable signal to illuminate. We should have used the scope to test this signal instead of relying on the LED's.

Demo Code Below

****

```
// connect motor controller pins to Arduino digital pins
// Linear Actuator - A

int LIN_enB = 46;
int LIN_in3 = 50;
int LIN_in4 = 48;

// Left front - Motor - A
int LF_enA = 2;
int LF_in1 = 3;
int LF_in2 = 4;

// Left Back- Motor B
int LB_in3 = 5;
int LB_in4 = 6;
int LB_enB = 7;

// Right Front- Motor B
int RF_enB = 8;
int RF_in4 = 9;
int RF_in3 = 10;

// Right Back - Motor A
int RB_in2 = 11;
int RB_in1 = 12;
int RB_enA = 13;
```

```
void setup()
{
  // set all the motor control pins to outputs
  pinMode(LIN_enB, OUTPUT);
  pinMode(LIN_in3, OUTPUT);
  pinMode(LIN_in4, OUTPUT);

  pinMode(RB_enA, OUTPUT);
  pinMode(RB_in1, OUTPUT);
  pinMode(RB_in2, OUTPUT);

  pinMode(RF_in3, OUTPUT);
  pinMode(RF_in4, OUTPUT);
  pinMode(RF_enB, OUTPUT);

  pinMode(LF_enA, OUTPUT);
  pinMode(LF_in1, OUTPUT);
  pinMode(LF_in2, OUTPUT);

  pinMode(LB_in3, OUTPUT);
  pinMode(LB_in4, OUTPUT);
  pinMode(LB_enB, OUTPUT);
}

void demoOne()
{
  // this function will run the motors in both directions at a fixed speed
  // turn on Linear Actuator for 6 seconds
  digitalWrite(LIN_in3, LOW);
  digitalWrite(LIN_in4, HIGH);
  // set speed to 200 out of possible range 0~255
  analogWrite(LIN_enB, 255);

  delay(6000);
// Reverse linear actuator for 6 seconds
  digitalWrite(LIN_in3, HIGH);
  digitalWrite(LIN_in4, LOW);
  // set speed to 200 out of possible range 0~255
  analogWrite(LIN_enB, 255);

  delay(6000);


// Set Motors to forward for 6 seconds
//Right Back
  digitalWrite(RB_in1, LOW);
  digitalWrite(RB_in2, HIGH);
  // set speed to 200 out of possible range 0~255
  analogWrite(RB_enA, 255);

//Right Front
  digitalWrite(RF_in3, HIGH);
  digitalWrite(RF_in4, LOW);
  // set speed to 200 out of possible range 0~255
```

```
  analogWrite(RF_enB, 255);

//Left Front
  digitalWrite(LF_in1, LOW);
  digitalWrite(LF_in2, HIGH);
  // set speed to 200 out of possible range 0~255
  analogWrite(LF_enA, 255);

//Left Back
  digitalWrite(LB_in3, HIGH);
  digitalWrite(LB_in4, LOW);
  // set speed to 200 out of possible range 0~255
  analogWrite(LB_enB, 255);

  delay(6000);

// Set Motors backwards for 6 seconds
//Right Back
  digitalWrite(RB_in1, HIGH);
  digitalWrite(RB_in2, LOW);
  // set speed to 200 out of possible range 0~255
  analogWrite(RB_enA, 255);

//Right Front
  digitalWrite(RF_in3, LOW);
  digitalWrite(RF_in4, HIGH);
  // set speed to 200 out of possible range 0~255
  analogWrite(RF_enB, 255);

//Left Front
  digitalWrite(LF_in1, HIGH);
  digitalWrite(LF_in2, LOW);
  // set speed to 200 out of possible range 0~255
  analogWrite(LF_enA, 255);

//Left Back
  digitalWrite(LB_in3, LOW);
  digitalWrite(LB_in4, HIGH);
  // set speed to 200 out of possible range 0~255
  analogWrite(LB_enB, 255);

  delay(6000);

}

void loop()
{
  demoOne();
  delay(1000);
}
```

Video of [Motor demo test]

The second part of the coding process is developing the working code to allow real time RC control of the rover.

This process is quite complex for novice coders and is currently in development.

Below is a video of oscilloscope analysis of the PWM input signals from the receiver.

Video of [Analysis of reciever signal]

We can use the serial plotter to analysis the signal and calibrate the inputs to the Arduino.

The green is forward/reverse and the yellow is steering. The overlay is where the channels are mixed.

Serial Plotter Image

Almost ready for Mars!