Open Rover
Seth King

The overriding object of Open Rover is to create a low cost outdoor robotics platform. It will lower the barrier of entry for school projects, professional engineering teams or even the lone hacker in the basement. Open Rover will free teams to focus on end applications instead of hardware and basic navigation issues.

Two main goals presented for this project:

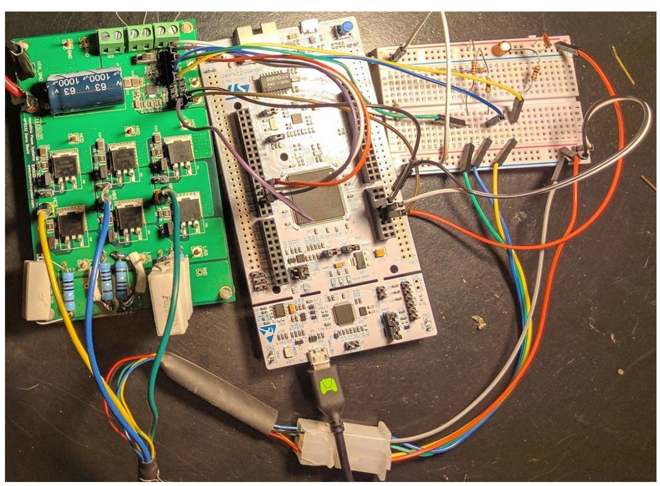
1. Develop a low cost motor controller that will allow the usage of off-the-shelf hub motors.
2. Create models of the different version of Open Rover and test them in a virtual environment. A stretch goal was to build a proof of concept rover and compare this to the models.

Development of motor controller

Three development paths were available in the proposal. All three paths were tested, and the results were conclusive as to which option produced the optimal results.

Option one: Use off-the-shelf evaluation boards

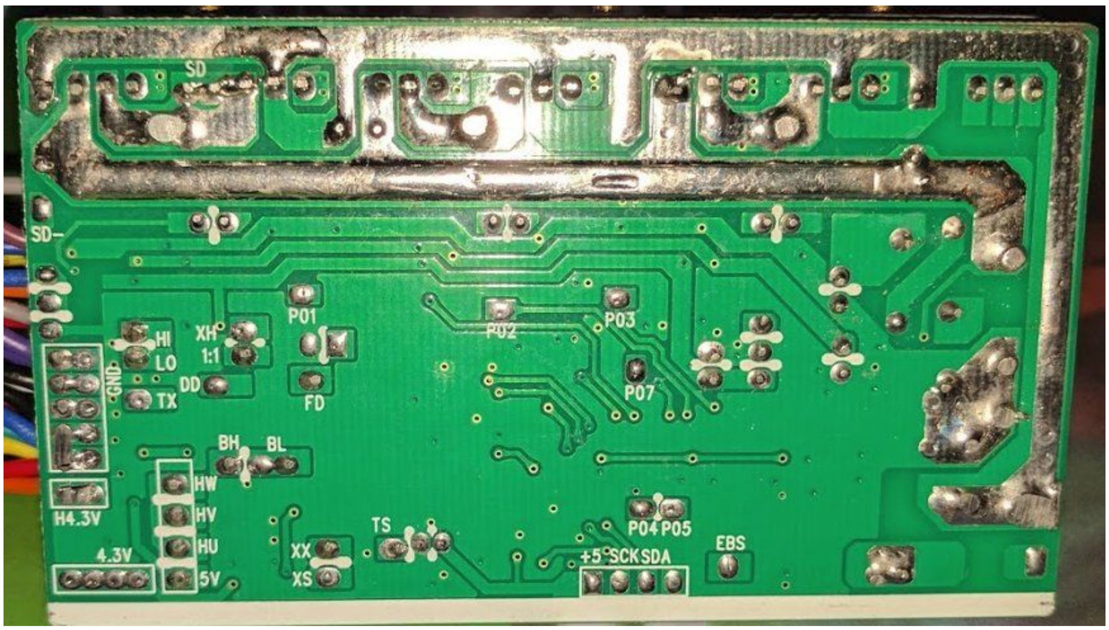
Two Monolithic Power Systems EV6532 evaluations boards were acquired. The boards require three control signals as well as power and motor and sensor connections. The three control signals, Speed (PWM), Direction (TTL) and Brake (TTL), were provided by an ST embedded controller. A 36V battery serves as the power source for the motor. Initially, a simple circuit was created to power the Hall Effect sensors in the motor and to pull up the voltage to the embedded controller voltage. An imbalance in the ground planes between the embedded controller and the evaluation boards was discovered as well as a significant noise issue that was causing the motor to turn off upon starting. Additional isolation and filter circuits were created, but the noise continued to hinder startup of the motor. At this point in development, option one was shelved until all of the other options were tested.



EV6532 with ST microcontroller and breadboard for pullup resistors.

Option two: Use off-the-shelf scooter motor controllers

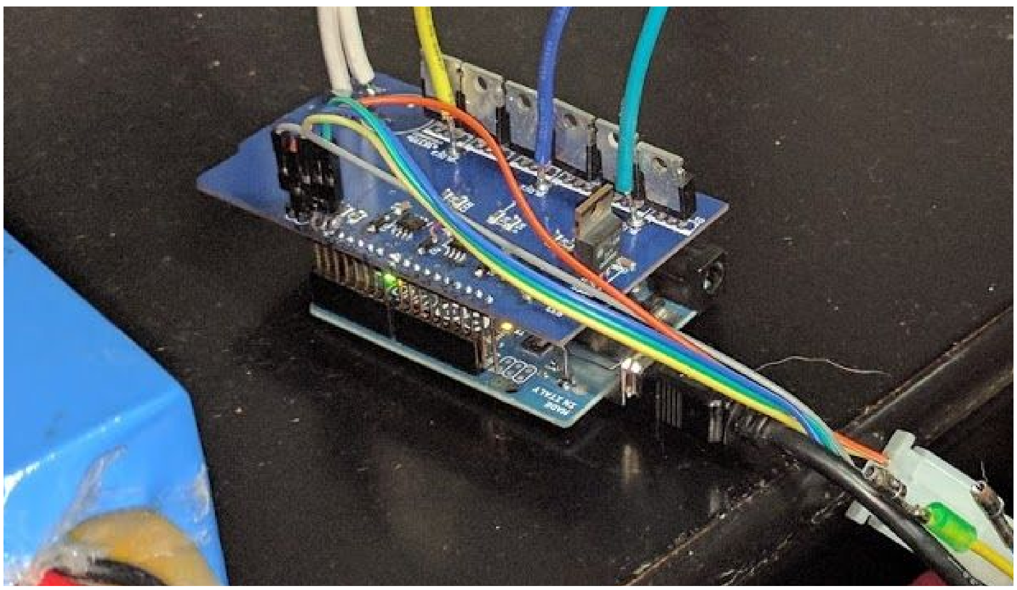
Three scooter motor controllers were purchased off of Alibaba. These controllers were low cost, and documentation was minimal. Only two controllers were needed, but a third controller was purchased in order to disassemble and reverse engineer. The controller came with 9 connectors. Battery, hall, and motor connections were similar to the connection on the motor. The vendor would not provide any additional documentation about the controllers so this path was abandoned. Despite not being able to use the controllers, knowledge was gained from disassembling one of them. The layout of the Field Effect Transistors(FET) was used in option three to allow for higher current usage.



Underside of Alibaba controller. FET layout was used in the custom shield design.

Option three: Design custom controller based on open hardware designs (Arduino)

A BLDC shield for the Arduino was designed based on criteria retrieved from reverse engineering the low cost Chinese controllers. The boards were assembled and tested. A design flaw was found pertaining to the voltage regulation of the FET driver, and a new part was used to dissipate the appropriate amount of heat (power). Once this issue was addressed, the hub motors were successfully started in both forward and reverse. A command protocol was implemented to allow the setting and retrieving of current, pwm, velocity, gains, and direction. A PI velocity control algorithm was implemented. An additional improvement for the future will be current regulated speed control or direct torque control.[1][2] There a multiple hardware improvements that need to implemented to make the controller more robust. These include reorienting the Field Effect Transistors to allow for a passive heat sink. Replacing the voltage regulator with a voltage switcher. And better strain support or mounting for the cabling.



Custom shield for Arduino

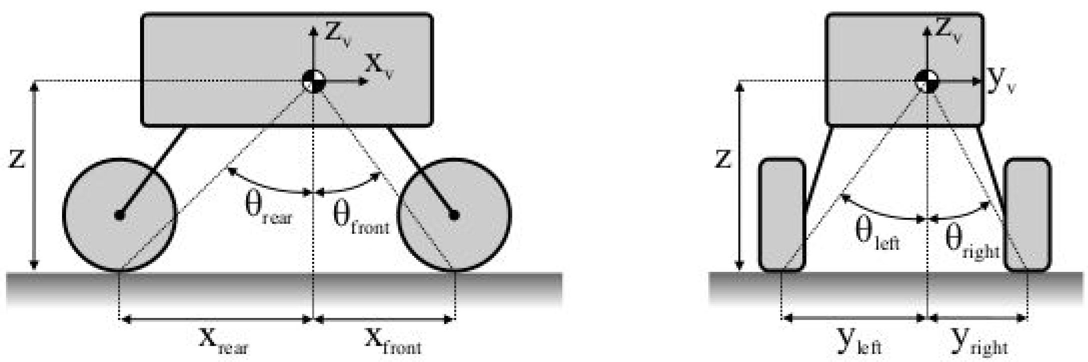
[1] http://ieeexplore.ieee.org/abstract/document/4270633/

[2] http://www.st.com/content/ccc/resource/technical/document/application_note/90/b9/df/c9/0c/cc/4c/4d/CD00075973.pdf/files/CD00075973.pdf/jcr:content/translations/en.CD00075973.pdf

Modeling and Parametric Selection

The second main goal of this Special Problems course was to model the various versions of the Open Rover. There are three version of Open Rover: 4 wheels(two hub motors, two passive motors), 4 wheels all hub motors, and 6 wheels all powered. The main requirement difference between the 4 wheeled and 6 wheeled rover was to give an additional option for applications that would require higher torques e.g. hauling a large load or climbing more difficult terrain. The 4 wheeled model with only two hub motors is the proof of concept that will be constructed. Only two hub motors are available for the PoC. Coming into this project, there was no true guiding principle that would dictate the Rover's dimensions. The general idea was to have sizing similar to a push mower. Multiple metrics were extracted from the PhD thesis of Thomas THÜER.[3] Theta of static stability is the maximum angle in which the stability margin becomes zero. The x(longitudinal) and y(lateral) variables are independent of each other and stability is calculated based on the geometry of the center of gravity compared to the points where the wheels contact the ground. The formula is as follows:

$$\theta ss = atan(Xrear/z)$$

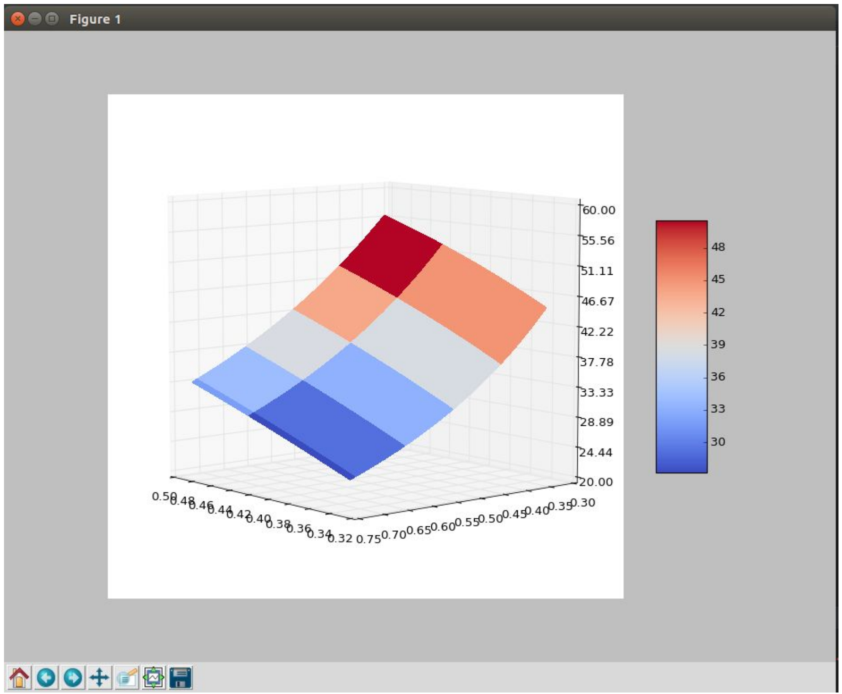


Longitudinal and lateral stability figured taken from Thomas THÜER PhD thesis

Xrear can be exchanged for Yleft/right, and both are half of the width and length respectively because the vehicle has a symmetrical center of gravity on both the x and y axis. Therefore, the wider and longer the Rover is at a specific height, the more stable it is. There is an arbitrary factor of, from a practical standpoint, how large the Rover be.

[3] Mobility evaluation of wheeled all-terrain robots - Thomas THÜER

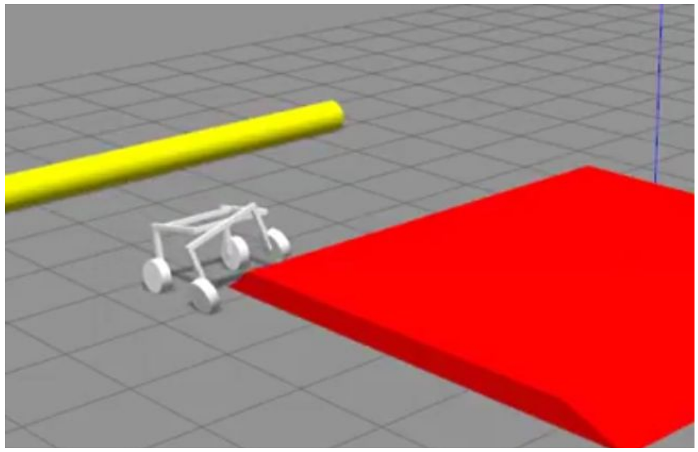
A three dimensional graph was created to give a better view of how the Theta of static stability changes with varying dimensions.



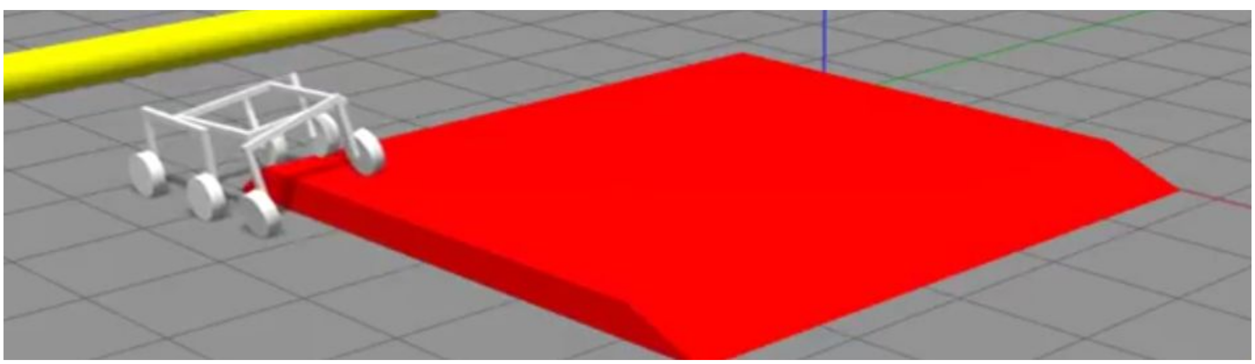
The first model had a length of 1 meter and a width of 0.5 meters. To maximize the Rover's stability in both the x and y, 0.75 meters was selected. Cost and practicality are limiting factors in making the rover any larger. For the target applications, the general size of a yard lawn mow used as a guiding principle. This improved the combined static stability as the table below shows:

Theta_SS_x	Theta_SS_y	y	x	z	combined
44.16	25.89	0.5	0.25	0.515	70.05
36.06	36.06	0.375	0.375	0.515	72.12

The initial design of the 4 wheeled Rover was going to be a static frame. By adding a simple pivot point to the front pair of wheels, the stability of the Rover is improved by allowing all of the wheels to maintain contact with the ground.



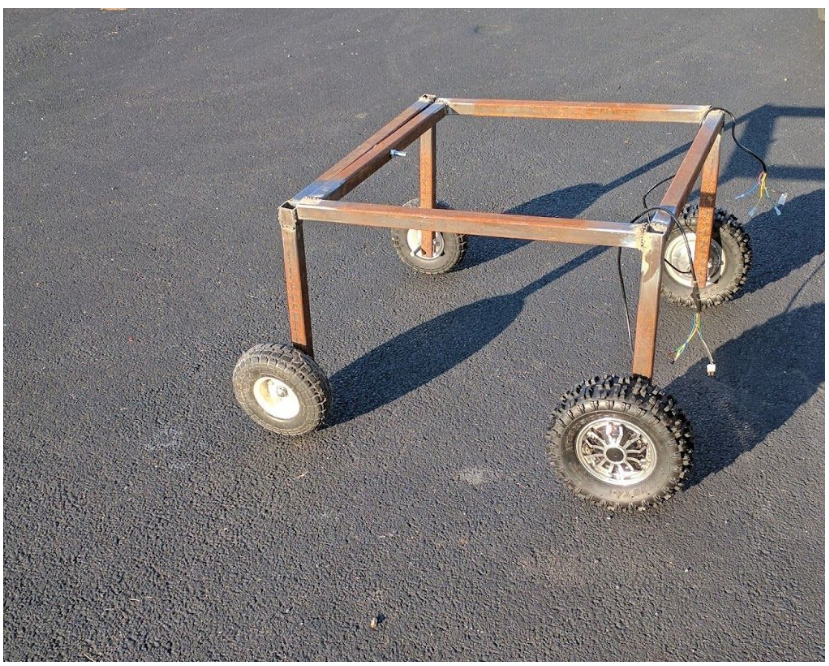
A six wheel model was designed for applications that would require higher torque. For the sake of reusability and consistency, the back suspension was replicated on each side of the main chassis for two independent boogie suspensions. This continues the dynamic of allowing all wheels to be in contact with the ground as an obstacle is traversed. The main frame expanded to 1.0 meters in length to allow enough clearance between the rear wheels and the middle wheels in the boogie. The ExoMars rover was inspiration for this suspension configuration.[4]



[4] https://en.wikipedia.org/wiki/ExoMars_(rover)

Construction of the Rover

The modeling was completed based on 50mm square tubing. The mounting bracket for the hub motors was 40mm, so 38mm square tubing was used in the construction of the Rover. Only two hub motors were available for the proof of concept, so two smaller, non-powered wheels were utilized on the front. To protect the motor from overcurrent damage a 10 amp fuse and switch were wired inline with each battery and controller pair. Robot Operation System(ROS)[5] is one of the industry standard for robotic software subsystems. It will be used to collect sensor data, calculate path planning and communicate the appropriate velocities to the left and right motor controllers. Simultaneous localization and mapping(SLAM)[6] will be used to map out the target environments and the Nvidia TX1 has very efficient computer vision capabilities that will aid in this task. These functionalities are not within the scope of the class so a Wiimote(bluetooth video game controller) will be used to simulate directives to the wheel subsystems. The Wiimote will communicate via bluetooth to a laptop. A ROS node will convert the analog joystick values into an appropriate velocity and direction which will be sent to each wheel via a serial communication bus. Additionally the ROS node will request current, velocity, gains and PWM settings. All of the wheel data will be bundled and sent to the system for instrumentation and data capture for future analysis.



[5] http://www.ros.org/
[6] https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping

Lessons Learned

The biggest breakthrough occurred in getting the hub motors spinning with the custom controller. One of the most important lessons learned involved a fundamental electrical engineering tenant: power dissipation of a linear voltage regulator versus switching power supply. The initial design used a linear voltage regulator that was consistently being damaged. The power that needed to be dissipated was too large, and the high temperature would then damage the regulator. This caused additional problems because the higher battery voltage would be applied to the motor drivers and damage them as well. Once this setup was replaced with a switching power supply, the controller was significantly more reliable. There were a lot of unknowns that were exposed throughout the duration of this project: sensors needed for SLAM, the importance of suspensions, basic ROS node architectures, static stability, and practical construction techniques.

Conclusion and Future Steps

The final goal of this project was to use to a WIimote to drive the Rover, thereby testing the movement model and suspension. This was unable to be accomplished because the motor controllers were not robust enough for extended use. Both controllers were damaged when attempting to test the change of direction command. The time and cost to repair the controllers would extend the project outside the allotted time. There are multiple improvements that can be implemented in order improve the construction and reliability of the motor controllers. Continuing to reverse engineer the low-cost Alilbaba controllers will hopefully result in more gained knowledge that can be applied to future versions. The quality of construction of the current Rover could be improved as well. The frame would need to be disassembled and welded with higher quality tooling.

Once the controllers are stable, the plan will be to implement a SLAM algorithm with one or two cameras. The laptop will be replaced with the Nvidia TX1. This will allow Open Rover to map an area that has been digitally fenced off. A simple smartphone app could be used to create the digital boundaries. Providing simple straightforward navigation APIs are a foundational feature of Open Rover. Additional hub motors will need to be acquired as well so that prototypes of the four wheel and six wheel Rovers can be constructed.

Additional Resources:

Open Rover Construction and Modelling videos