Rail Riding Robot
Final Report

Bereket Guta, Jessica Ma, Bowen Xue

Project Sponser: Dr. Mark Johnson, *Professor, UBC Ecohydrology*

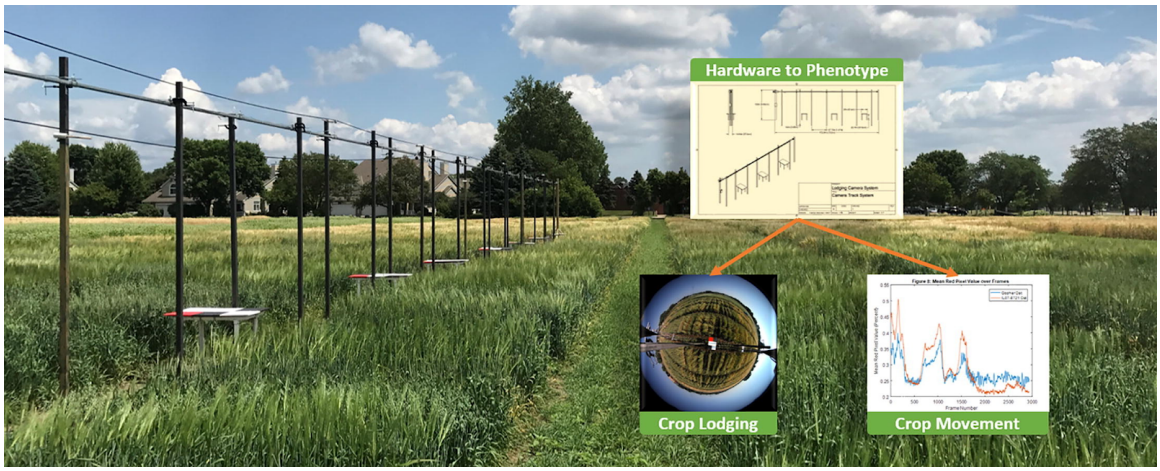
Project Number: 2013

ENPH 459

The University of British Columbia

UBC Ecohydrology

April 8, 2020



A similar imaging system[1]

[1]Image sourced from [Susko et al., 2018]

Contents

1 Executive Summary **iv**

2 Introduction **1**

2.1 Sponsor . . . 1

2.2 Current Methods . . . 1

2.3 Objectives . . . 2

3 Discussion **2**

3.1 Theory . . . 4

3.1.1 Photogrammetry and image processing . . . 4

3.1.2 Camera . . . 5

3.1.3 Movement of the robot . . . 6

3.2 Design . . . 6

3.2.1 Control system . . . 6

3.2.2 Rail . . . 7

3.2.3 Platform . . . 8

3.2.4 Wheels and Motor . . . 10

3.2.5 Power . . . 10

3.2.6 Software . . . 11

3.3 Tests . . . 11

3.3.1 Modular testing . . . 11

3.3.2 Power . . . 13

3.3.3 Integrated testing . . . 13

4 Conclusion **13**

4.1 Deliverables . . . 13

4.2 Recommendations and future considerations . . . 13

4.2.1 Recommendations . . . 13

4.2.2 Future Considerations . . . 14

Appendices **15**

A Rail Rigidity **15**

A.1 Bending . . . 15

A.2 Torsion . . . 16

B Software and Data **16**

C Design **16**

D Bill of Materials **16**

List of Figures

1 Phenospex Field Scan 1
2 Main components of system 3
3 Illustration of imaging technique 4
4 Map of Normalized Different Vegetation Index 5
5 Communication systems 7
6 view of the connector joining the two rails 8
7 Free Body diagram of the Platfrom 9
8 Wheel Labels 9
9 Powered Components 10
10 Flowchart for Software 12
11 Bill of Materials 17

List of Tables

1 Spectral bands of Parrot Sequoia 5

1 Executive Summary

Increasing global population creates a demand for intensification in methods of food production. Smart crop management is an environmentally sustainable way to increase and improve crop yield. As systems grow, we cannot rely on manual inspection and instead turn to automated systems.

We aim to design a modular, portable, extendable, and affordable ground-based rail system upon which a robot can move up and down a field row to image the crops below using a variety of cameras and sensors. This data can later be stitched together with software to create indexed maps and other models of the crops that inform farmers about the status and health of their land.

We present our solution as a water-jet cut robot "platform" that rides along a rectangular rail, driven by a single 12V motor and controlled with a Raspberry Pi that interfaces between the camera, sensors, and drive train. The speed of the robot is regulated with encoders, and the camera attached to the robot takes photos at a constant time interval to achieve a desired amount of overlap. The platform is easily manufactured with the water jet cutter, which will allow it to be extended to accommodate different cameras in the future. The "platform" robot is self-aligning and self-locking, allowing it to travel down arbitrary lengths of rail. This system can be extended to using a threaded metal plate to connect consecutive pieces of rail together. The total cost of our system is USD $ 311.82, with each additional length of 6 ft rail costing USD $ 34.30. This price excludes the support system, which is out of the scope of this project.

2 Introduction

As global population increases, so does the demand for high-quality sustainable methods of food production. Common practices such as increased use of fertilizers, plant growth regulators, and pesticides can lead to severe environmental damage when ecological factors are not prioritized. Instead, crop yield can be optimized with smart soil and water management. However, monitoring must be done throughout the growing season. The large scale of these systems mean that reliance on manual intervention becomes unrealistic and insupportable. Thus, we must pivot to technologically advanced, automated crop management tools that can characterize a variety of factors such as soil quality, rainfall, and crop yield.

2.1 Sponsor

This project is sponsored by Dr. Mark Johnson of the UBC Ecohydrology Research Group as part of an effort to gather multiple crop measurements per day using multispectral cameras. Using this data they propose to evaluate strategies for sustainable intensification in agriculture. We can obtain a multitude of information from crop imaging, including but not limited to irrigation variability, soil erosion, vegetation indices, and crop yield. The goal of this project is to create a versatile linear transport system that many different sensors and measurement tools can be mounted on later on.

Figure 1: Phenospex Field Scan

2.2 Current Methods

Systems for crop imaging can be categorized into aerial and ground-based devices. In both approaches, a data-capture module (usually a camera) is attached to the transport system and carried across the desired area. Both systems have their benefits and drawbacks, however, our sponsor favoured the use of a ground-based linear motion control system because they are robust in terms of precision and accuracy. Specifically, the Phenospex Fieldscan in Figure 1, is a rail-based solution that offers state-of-the-art grid monitoring capabilities. It provides frequent, consistent and autonomous data collection but is costly to install and difficult to use for the average consumer. However, there have been recent advancements in designs [Alcala et al., 2019], [Susko et al., 2018] to optimize this system to improve affordability and customizability while not sacrificing on the precision or accuracy. We aim to improve on the Phenospex by making our system more affordable, portable, and customizable.

2.3 Objectives

The objectives of our project changed as we investigated ways to achieve them. We will first address the initial objectives as envisioned by our sponsor before we took on the project. Pursuing and discussing the objectives guided us in the evolution of the project and led us to our final objectives. Originally, our sponsor wanted the following things from this project:

1. Weather-proof, fully autonomous system that can be unsupervised and left out to run indefinitely
2. Docking station supported with solar power used to charge the robot after a day's work
3. Automatic wireless transmission of photos
4. Portable, removable, and modular rail system that can be extended up to 100m

We realized certain aspects of our original goals were unrealistic and unnecessary. Multispectral cameras designed for photogrammetry are usually expensive and delicate; leaving them unsupervised for long periods of time, especially in variable weather conditions, is inadvisable. In fact, the Parrot Sequoia (the camera we have designed our system around) is recommended not to be used in any amount of precipitation. Since an operator will at least need to retrieve the camera at the end of each day, a charging station is not a priority. Finally, creating stable, level supports in dirt is a complicated task that requires large machinery and coordination beyond the scope of this project. Since our project would be a proof-of-concept of sorts, and the final form of this device may house many more cameras, we prioritized a design that is easily manufactured for quick iterative design. Thus, our refined objectives are:

1. A battery-powered, rechargeable robot that can perform full runs up and down 100 m of rail unsupervised.
2. A single camera (the Parrot Sequoia) that attaches to the robot and takes photos at regular intervals.
3. Infrastructure to transmit photos wirelessly as well as wired options.
4. Proof-of-concept of connecting two rails together.

In the next section we will introduce and justify our design.

3 Discussion

The robot consists of a moving platform attached to a rectangular rail with 5 wheels (1 driven wheel and 4 locking wheels). A rotary encoder and limit switches on either end allow us to regulate the speed and determine the direction of travel. A multi-spectral camera (the Parrot Sequoia) captures the desired image data at specified time and spatial intervals. The entire system is controlled by a Raspberry Pi, and powered with a 12 V battery.

The operation of the robot is split into three main stages. In the first stage (setup), we set up the camera by giving it a time interval and calibrating it as suggested by the user manual. We also calculate the speed at which the robot will move. We refer to a "run" as one trip down a length of rail and back. In the forward direction, the camera will take photos while the speed is carefully monitored. In the backward direction the robot will simply be moving back to the base. The camera is controlled via USB connection to the Raspberry Pi during the run. After a run (post-run), the user can download photos wirelessly via the web interface, using a USB connection, or using the SD card in the Parrot Sequoia.

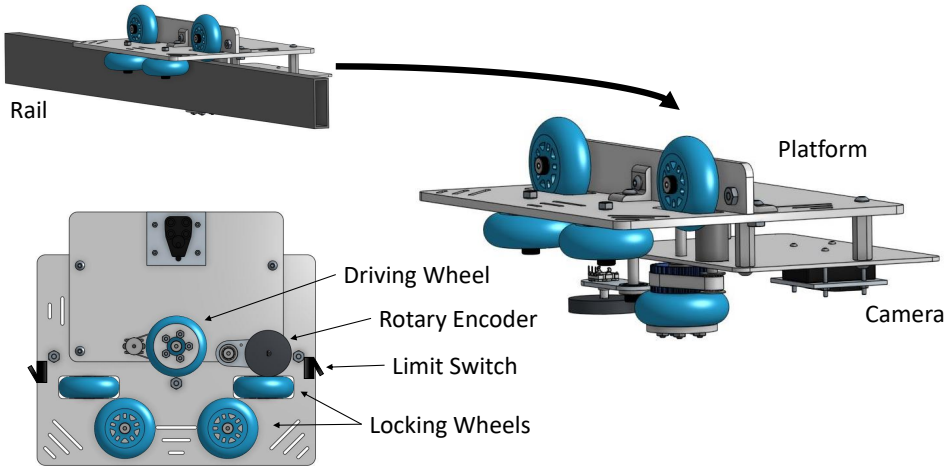


Figure 2: Main components of system

1. **Control System**: The control system consists of a central processor, the Raspberry Pi, paired with an encoder and two limit switches. This allows us to control locomotion and the camera.
2. **Rail**: The rectangular extrusion. Each solid piece is 3 ft
3. **Platform**: The platform consists of metal plates that houses all components and traverses up and down the rail.
4. **Wheels and Motor**: This drive system is responsible for providing locomotion to the platform in a smooth and controlled manner.
5. **Power**: The platform will be powered by a 12 V battery distributing power to the motor, control system, and the camera.

In the following section, we will first introduce the theory of our project. We will then discuss in detail each major component that achieves the goal of our project. We will then examine the tests we conducted to showcase the limits of the systems and present our results.

3.1 Theory

Our robot is the first step in a data analysis pipeline; specifically, we are responsible for the data acquisition. Therefore we must understand how the data is meant to be used. This section details the foundation of our design, including limitations of our supplies and key relationships between variables.

3.1.1 Photogrammetry and image processing

Photogrammetry is the science of making measurements and observations from photos. A photograph or series of photographs can be used to create a model, map, drawing, or measurement of the subject in real life. In the context of agriculture, we can distinguish "terrestrial" or near-ground photogrammetry and "aerial" photogrammetry.

Our project deals with terrestrial photogrammetry as our robot will operate on a rail supported on the ground, at a variable height of 1-2 m depending on the subject. Consecutive photos will be stitched together to create a full view of a line of crops. Photo stitching is done by analysing common elements in consecutive photos with a known amount of overlap. Overlap is the percentage of pixels that consecutive photos have in common. Figure 3 illustrates an example of how photos will be taken with our system and what we define overlap to be.

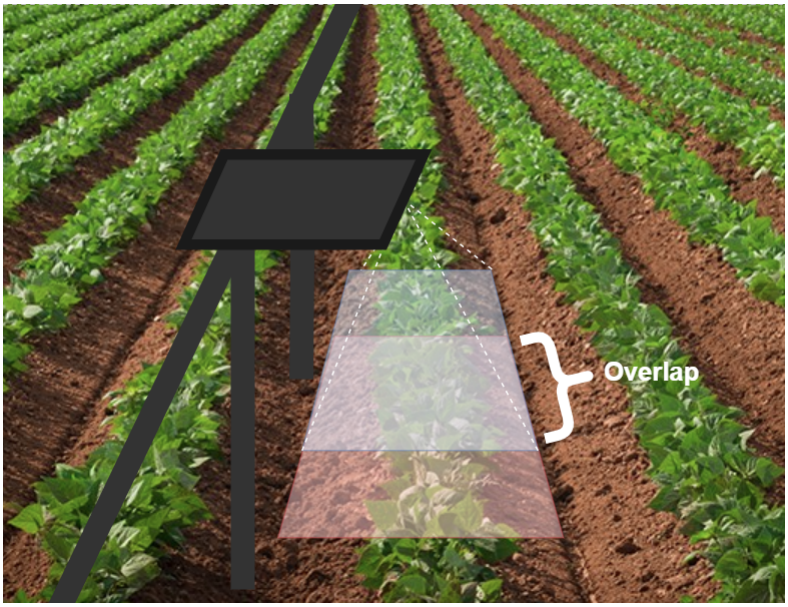


Figure 3: Illustration of imaging technique

[Linder, 2009] suggests 60% to 80% overlap, which we cross-referenced with the Parrot Sequoia user manual and our sponsor. We aim to operate with 75% to 85% overlap between consecutive photos.

Existing software recommended by the Parrot Sequoia user manual include PIX4D and MicaSense ATLAS.

These software programs can stitch together photos and create indexed maps. An example of an indexed map is one that plots the Normalized Difference Vegetation Index (NDVI), shown in Figure 4. NDVI is a graphical indicator that can assess the presence of live green vegetation. It is unknown whether the next stage of this project will use commercial software or have custom programming, and what particular metrics will be of interest.

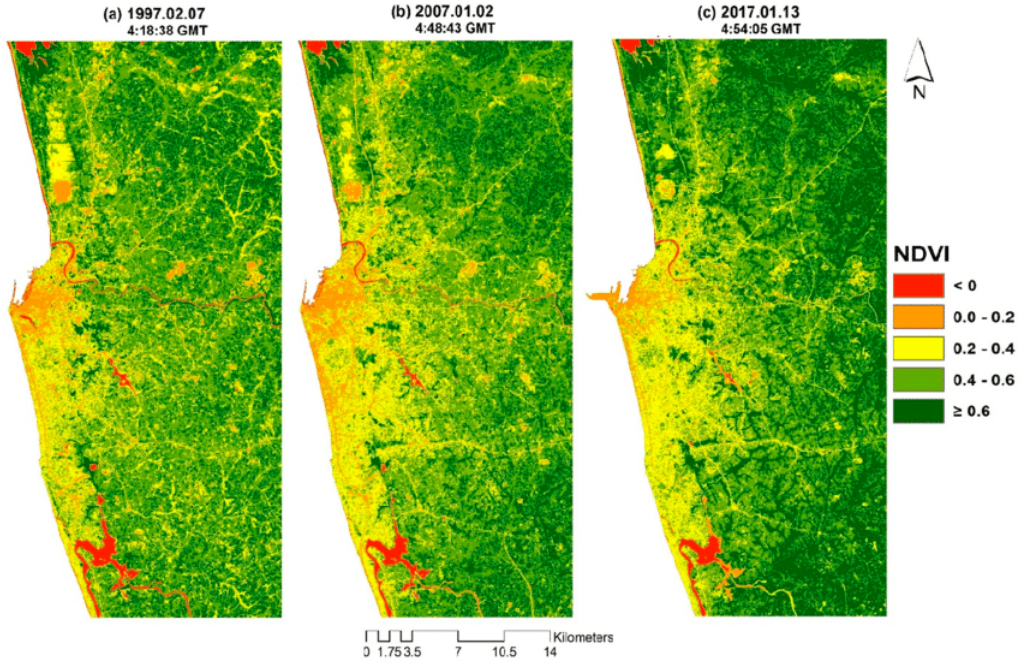


Figure 4: Map of Normalized Different Vegetation Index

3.1.2 Camera

The camera we are using for this project is the Parrot Sequoia (PS). It has 5 sensors that image in different spectra. Table 1 summarizes the sensors on the PS.

Name	Wavelength	Resolution
Green	550 nm	1.2 Mpx
Red	660 nm	1.2 Mpx
Red-edge	735 nm	1.2 Mpx
Near infrared	790 nm	1.2 Mpx
RBG	-	16 Mpx

Table 1: Spectral bands of Parrot Sequoia

A key limitation in our design is that there is a minimum 1 s delay between shots on the RBG camera, and a minimum 0.5 s delay between photos from the monochrome sensors. We must take this into account when we schedule the interval at which the camera takes photos.

We can control the settings and function of the PS in a variety of ways:

1. Web interface available via connection to WiFi hotspot
2. Picture Transfer Protocol (PTP) libraries and programs such as gphoto2, ptpy via USB
3. HTTP API accessible via WiFi connection

We determined that options 1 and 2 are the best methods for wireless and wired control respectively. The web interface is intuitive and user-friendly but requires manual control and a monitor and mouse system. The PTP-based libraries are powerful but operate via USB connection.

The HTTP API is accessed wirelessly. The user would have to connect to the WiFi hotspot of the PS, and then use the API to download files or control the camera. In most cases, the web interface is more efficient and user-friendly so the HTTP API is not necessary.

3.1.3 Movement of the robot

The Parrot Sequoia has three capture modes: single shot, time interval (s), and distance interval (m). Since the Sequoia is designed for drones, position tracking is done with GPS signals. The camera is expected to operate at altitudes of 30 - 150 m. In comparison, at 1-2 m above ground, the resolution of the distance interval shutter is not high enough for our purposes. Therefore we will regulate the speed of the robot and use the time interval shutter setting to achieve constant distance intervals. The relationship between the speed, time interval, and distance interval is:

$$\frac{\text{distance (m)}}{\text{shot}} = \frac{\text{metres}}{\text{second}} \times \frac{\text{time (s)}}{\text{shot}}$$

To find the appropriate speed, we use a spreadsheet to extrapolate the given data to lower heights, input a time interval ($>$1s if we are using the RBG sensor, $>$0.5 otherwise) and calculate the speed that will satisfy this relation.

3.2 Design

3.2.1 Control system

Our central processor is a Raspberry Pi 3 Model B. Specifications of this model are:

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 100 Base Ethernet
- 40-pin extended GPIO
- 4 USB 2 ports

- 4 Pole stereo output and composite video port
- Full size HDMI
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A

The main advantages are the built-in WiFi and Bluetooth capabilities as this allows us to connect to a variety of devices and will be able to support future work on this project that may include several cameras. The digital I/O are used for controlling and receiving input from the motor, collision switches, encoder, and start/stop buttons.

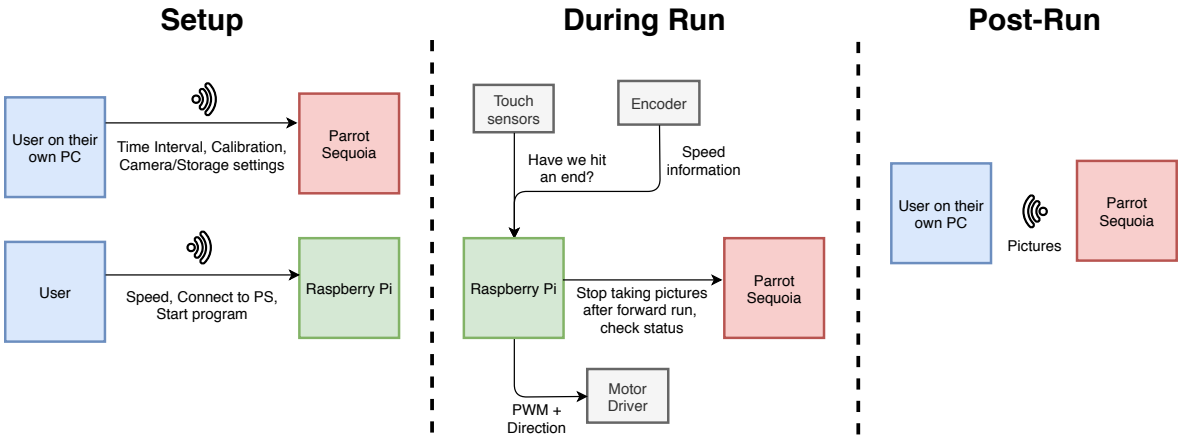


Figure 5: Communication systems

Figure 5 explains the general structure of the communication between the different components of the robot in the three stages.

3.2.2 Rail

We chose a rail structure based on four main criteria: rigidity (bending and torsional), weight, and cost. We needed a simple shape that is widely available. As there are many different types of extrusions, we considered a few key shapes (U, square, rectangle) and created a spreadsheet to analyze the relationship between price and rigidity. After we determined that a rectangular shape is the best option as it provides us with the greatest degrees of freedom for movement along the rail, we investigated the relationship between the dimensions/thickness and the rigidity. Finally, we concluded the best option is $3" \times 1" \times {}^{1}/_{16}"$ (width $\times$ height $\times$ thickness).

Two rails are joined together with a threaded aluminum plate that is inserted between the two rails as seen in Figure 6. We secure this plate with screws via the bottom of the rail that thread into the plate. The length of individual pieces of rail is a variable that depends on how often the rail is supported.

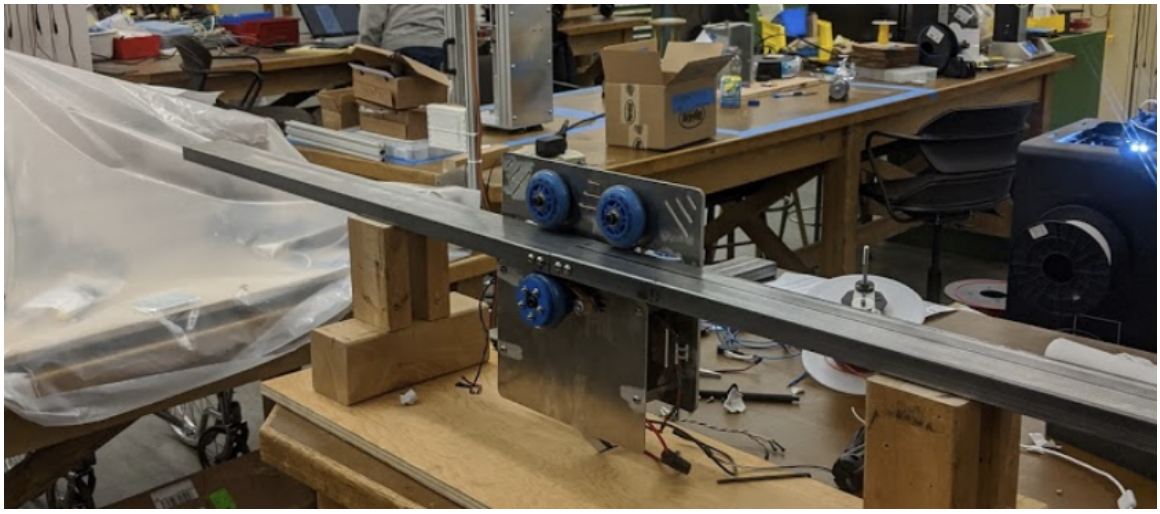


Figure 6: view of the connector joining the two rails

The spreadsheet we used is part of our deliverables, and the detailed calculations are given in Appendix A.

3.2.3 Platform

A key factor in designing the platform is ease of fabrication and assembly. Thus, our design is easily prototyped with a water jet cutter and assembled with off-the-shelf connectors and stand-offs. Decoupling the platform from the rail is a key part of maintaining extendability of the rail system. Hence, our design of the platform is such that it operates independently of the rail.

To maintain constant motion, the platform must be aligned with the rail, which is very difficult to do externally for long distances. To combat this we designed a self-locking mechanism where the moment created by the distribution of weight locks the wheels flush against the rail. This mechanism can be seen in Figure 7. The force F_{drive} on the wheels can be calculated using the center of mass of the platform and the relevant distances D and L.

$$F_{drive} = \frac{F_{\text{weight}} D}{L}$$

The relationship between F_{weight} and F_{drive} grows linearly as a function of the distance D which is dependent on the number of objects placed on the platform.

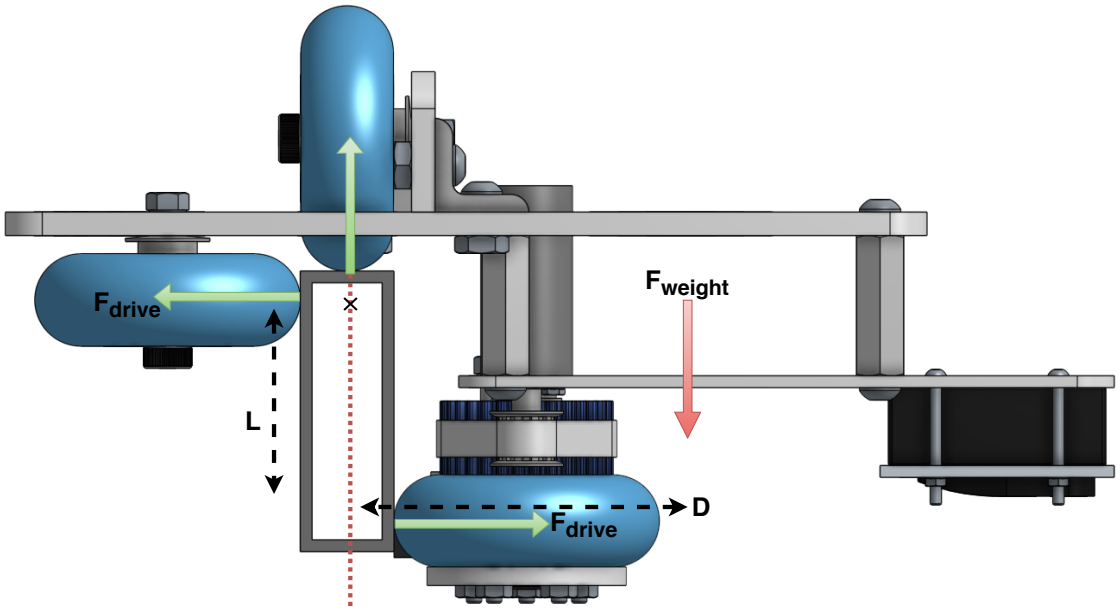
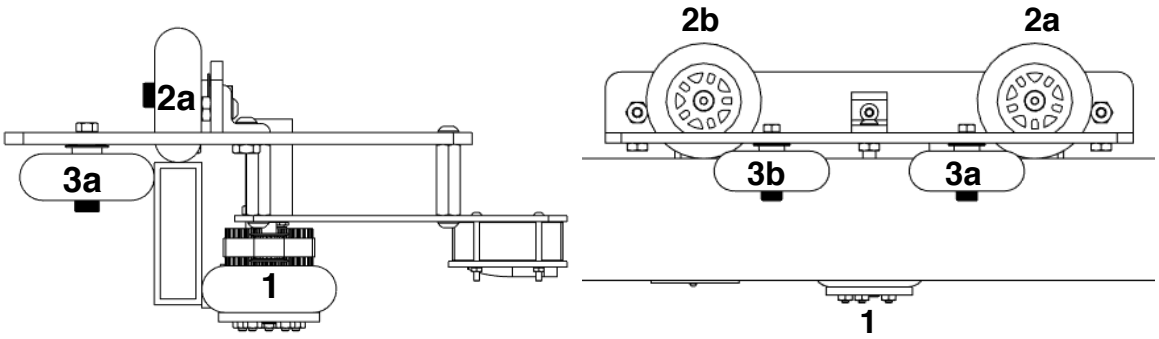


Figure 7: Free Body diagram of the Platfrom



(a) front view

(b) side view

Figure 8: Wheel Labels

3.2.4 Wheels and Motor

Movement of the platform should be constrained to only along the axis of the rail. As seen in from Figure 8, we used five wheels to achieve this minimal constraint. To prevent any imbalance or unwanted vibrations, we place two wheels 2a and 2b along the top of the rail. The distance between these wheels determines how impervious the platform is to rocking back and forth. The larger the distance the more impervious. The distance between the wheels placed on the longer sides of the rail (3a and 1) needed to be offset so that it could be pushed into the rail by the moment generated by our platform. We decided to use wheel 1 as our driving wheel since it was in the center of our platform and offers us the most balanced drive. The mechanism we decided to use to drive the wheel is a timing-belt and pulley attached to the wheel and a high torque motor. We decided to use this approach over a gear since the belt has a smoother operation and generates less noise. To drive the wheel, we simply clamped a pulley onto the wheel to ensure that the wheel and pulley moved as one. We attached the motor to our plate by inserting the motor shaft through the bottom plate and attaching the shaft to another pulley and connecting the two pulleys using a timing belt. We utilized slots for the motor mount to allow adjustment of the motor position, thus ensuring proper tension in the belt.

After interpolating the data as described earlier to find the relationship between speed, time interval between photos, and distance interval between photos, we found that the robot will operate at speeds in the range of 0.10 - 0.25 m/s. The calculations to find this are summarized in the Speed Calculations spreadsheet in our deliverables.

3.2.5 Power

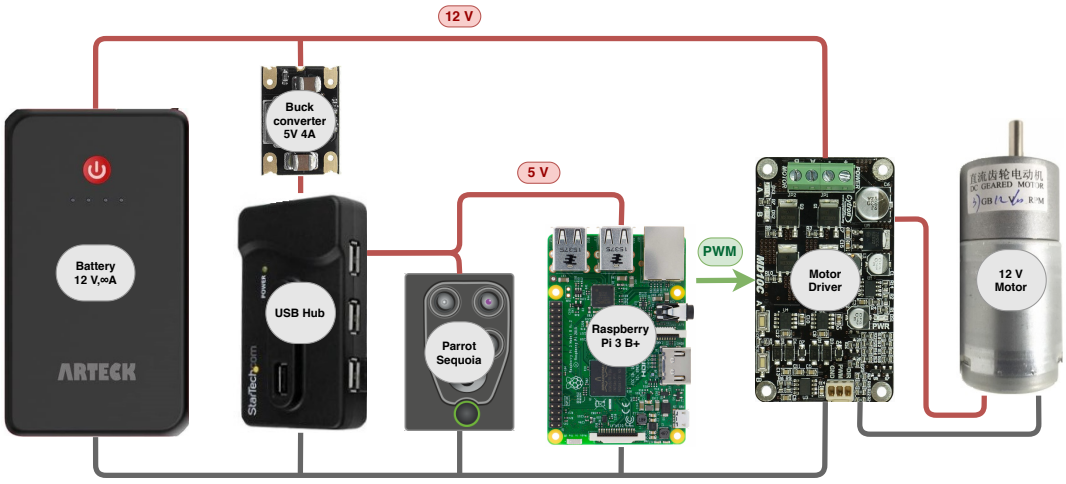


Figure 9: Powered Components

Our system is powered with a single 12 V, 4000 mAh source. Therefore the energy contained is:

$$12\,\mathrm{V} \times 4\,\mathrm{mAh} \times 3.6\,\mathrm{C/mAh} = 172.8\mathrm{kJ}$$

Assuming high performance, one run of 100 m at 0.15 m/s takes 30 minutes. During those 30 minutes, assuming worse-case power consumption we have 5V*2A for the PS, 5V*1.5A for the Raspberry Pi, and 12V*2A for the motor, we have:

$$10\,\mathrm{W} + 7.5\,\mathrm{W} + 24\,\mathrm{W} = 41.5\,\mathrm{W}$$

$$41.5\mathrm{J/s} * 1800\,\mathrm{s} = 74\,\mathrm{kJ}$$

Thus, with the current battery we only expect to be able to make a few runs on one charge, which is not ideal. However, since the majority of components are connected to the USB hub, changing our system to a larger battery should easy to implement.

We included the USB hub in order to have a separate power source for the PS and Raspberry Pi, as the Pi only draws up to 2.5 A, which is insufficient for also powering the PS. Here, we maintain a data connection via the hub while distributing 4 A between the Pi and the PS.

3.2.6 Software

All of the programs for our project are written in Python. Libraries:

- ptpy: PTP API written specifically for the Parrot Sequoia
- GPIO: control of digital I/O on the RP including interrupts and PWM

The software splits the motion of the robot into three stages that are triggered by the start button and the collision sensors on either end of the platform. A flowchart showing the logical sequence of the code can be seen on the next page in Figure 10.
The encoder is polled to keep track of the distance the robot should have travelled by some point in time. If the robot is detected to not be moving at the right speed, it will immediately stop the run and return back to the base. This is because likely there is some physical issue that is causing the robot to be stuck, and it is safer to return back and investigate rather than try to complete the run.

The full code can be seen in the Github repository. Note that there are some filler functions due to the fact that we were unable to complete calibration tests after restrictions from COVID-19.

3.3 Tests

3.3.1 Modular testing

We tested the major subsystems of our robot early on. Specifically, we measured and confirmed that our movement system works as expected by locking onto and moving back and forth along the rail. Our movement was robust enough to move without slipping upto angles of 45 degrees. While moving, the rail experienced minimal bending and torsion as expected from our design. We tested the cameras on their own and were able to capture and transmit the images we took wirelessly.

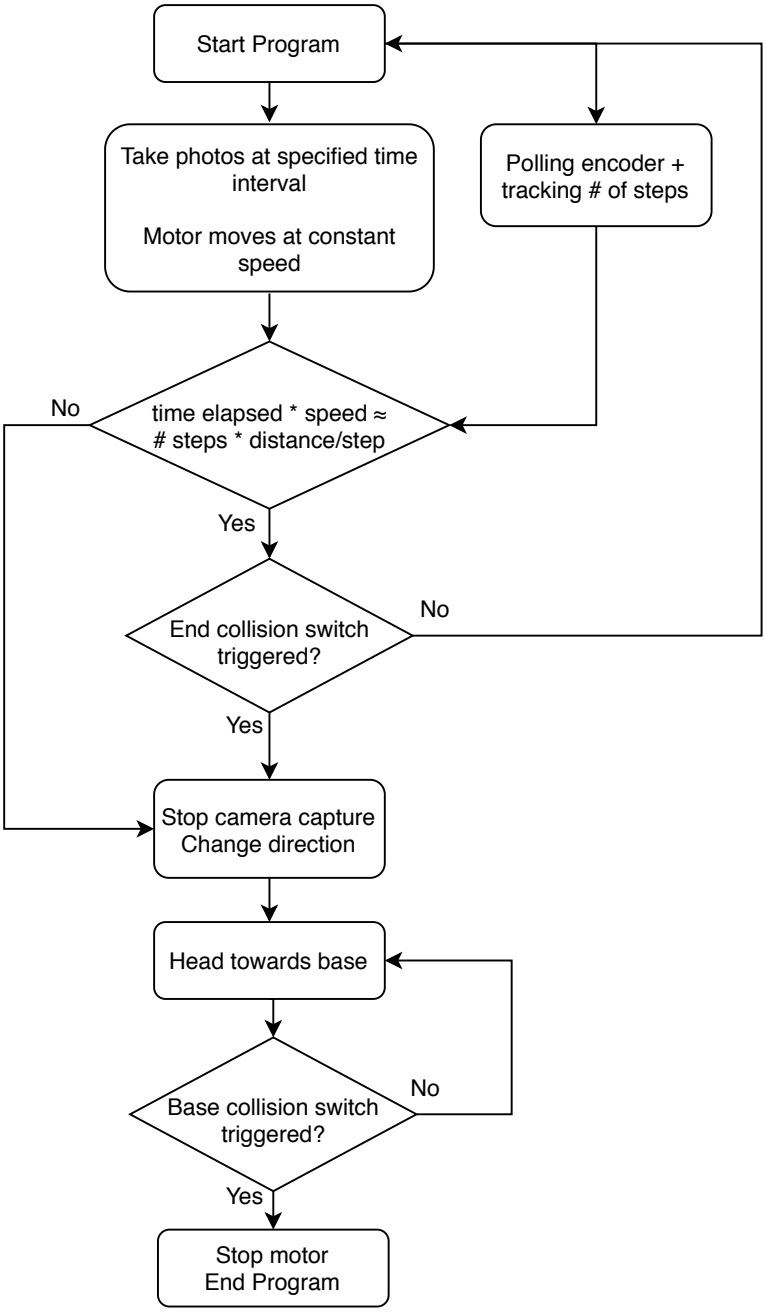


Figure 10: Flowchart for Software

3.3.2 Power

Unfortunately due to constraints from COVID-19 social distancing protocol, we did not complete testing for the power consumption of our system. However, we do not see this as a severe issue because of the ease of replacement of the battery. A similar, larger-capacity batter can be easily switched out. As shown in our Design section, we estimate that the current 4000 mAh battery is sufficient for several runs.

3.3.3 Integrated testing

Again due to the COVID-19 protocols, we did not get a chance to fully integrate each subsystem and test out the full functionality.

4 Conclusion

The rail riding robot is a robust and practical tool that can be used to monitor and gather multispectral data with little manual intervention. Sitting at heights between 1-2 m above the ground, it can capture and transmit images with overlap percentages ranging from 75% to 85%. It can run up and down the various lengths of rail and the rail system can be expanded with ease. It is an affordable and adaptable alternative to what is currently found on the market.
Despite these positive outcomes of the project, several major tasks remain unfulfilled. The most important being designing the support system that can be deployed out in the field for the rail. Any future work on this project should aim to solve this task first and foremost. In addition to this, a full integration test still needs to be done to prove that the system works as a whole.

4.1 Deliverables

1. A link to the CAD on Onshape
2. Bill of materials
3. GitHub repository of code for Raspberry Pi
4. Current prototype
5. Detailed on-site installation instructions

4.2 Recommendations and future considerations

4.2.1 Recommendations

These recommendations are based on things that we planned to accomplish but unfortunately were not able to due to restrictions on in-person meetings from COVID-19.

1. Add a button to make starting and stopping easier.
2. Test out the fully integrated system including the encoder and cameras.
3. Calibrate power to motor with speed of platform.

4.2.2 Future Considerations

1. Designing a support structure that can work on uneven ground. This is necessary for deployment in the field.
2. Expanding or lengthening the platform to accommodate multiple cameras.
3. Incorporating solar panels or some other source to automatically charge the battery of the system.

Appendices

A Rail Rigidity

Here we present the calculations that motivated our designs. The rail will be subject to both bending and torsion under normal operation.

A.1 Bending

The total deflection of the rail will be due to both the weight of the rail itself, and the weight of the platform. The deflection equation for a point load in the center of two supports is as follows:

$$\delta = \frac{PL^3}{48EI}$$

The deflection equation for a distributed load across the beam with supports on each side is as follows:

$$\delta = \frac{5\omega L^4}{384EI}$$

For the 3" x 1" x 1/16" steel tube, these are the material and geometrical properties:

$$\text{Area} = 312.5\,\text{mm}^2$$

$$I = 37251.83\,\text{mm}^4$$

$$E = 200\,\text{GPa}$$

$$\omega = 2.42\,\text{kg m}^{-1}$$

These are the assumed conditions in which the bending will occur in:

$$F = 44.4\,\text{N}$$

(Based on a 10 lb platform at center of rail.)

$$L = 1.82\,\text{m}$$

(Based on 6 feet per rail support structure.)

Therefore, the deflection due to the weight of the rail itself is:

$$\delta = 4.73 \cdot 10^{-5}\,\text{m}$$

The deflection due to the applied load will be:

$$\delta = 0.00076\,\text{m}$$

Under the assumed conditions, the total deflection of the rail will be:

$$\delta = 8.1 \cdot 10^{-4}\,\text{m}$$

This is within the acceptable deflection parameter of our design.

A.2 Torsion

The torsional deflection in degrees of a rail caused by an applied torque in is as follows:

$$\theta = \frac{TL}{JG} * \frac{180}{\pi}$$

For the 3" x 1" x 1/16" steel tube, these are the material and geometrical properties:

$$\text{Area} = 312.5\,\text{mm}^2$$

$$J = 132374.63\,\text{mm}^4$$

$$E = 200\,\text{GPa}$$

These are the assumed conditions in which the torsion will occur in:

$$T = 33.3\,\text{N m}$$

(Based on a 10 lb platform at 0.75 m offset at center of rail.)

$$L = 1.82\,\text{m}$$

(Based on 6 feet per rail support structure.)

Under the assumed conditions, the total angular deflection of the rail in degrees will be:

$$\theta = 0.132^\circ$$

This is within the acceptable deflection parameter of our design.

B Software and Data

The software for this project can be viewed at our **GitHub**.

C Design

A full view of our mechanical design can be see through our CAD. Which can be found **here**.

D Bill of Materials

The following is list of the components we obtained the development of the project prototype.

Category	Item	Description	Quantity	Unit Price (USD)	Cost (USD)	Source	Part #
Mechanical	L-Bracket	McMaster	3	5.21	15.63	https://www.mcmaster.com/47065T236	47065T236
	1/4 - 20 Screws - 5/8 length	McMaster	1	6.58	6.58	https://www.mcmaster.com/92949A539	92949A539
	Shoulder Bolts - 30 mm x 8 mm	McMaster	4	1.91	7.64	https://www.mcmaster.com/92981A205	92981A205
	Shoulder Bolts - 90 mm x 8 mm	McMaster	1	7.90	7.90	https://www.mcmaster.com/92981A794	92981A794
	Standoffs	McMaster	4	8.71	34.84	https://www.mcmaster.com/91115A414	91115A414
	Disk Spring	McMaster	1	3.87	3.87	https://www.mcmaster.com/96445K257	96445K257
	Black-Oxide Steel Oversized Washer	McMaster	1	3.87	3.87	https://www.mcmaster.com/98035A105	98035A105
	T5 Timing Belt Pulley	McMaster	1	8.7	8.70	https://www.mcmaster.com/1428N2	1428N2
	Timing belt (10T5 - 200mm)	SDP-SI	1	7.6	7.60	http://shop.sdp-si.com/catalog/product/?id=A_6T35MFD40100	A6T35MFD40100
	8 - 32 scews - 2' length	McMaster	1	5.77	5.77	https://www.mcmaster.com/92949A207	92949A207
	Aluminum Spacer	McMaster	1	1.80	1.80	https://www.mcmaster.com/92510a587	92510a587
	Steel Oversized Washer	McMaster	1	7.00	7.00	https://www.mcmaster.com/9111ba380	9111ba380
	3' x 1" x 1/16" Rail - 3 Feet Length	Metal Supermarket	2	22.36	34.30	https://ecommerce.metalsupermarkets.com/Material.aspx?ProductId=CTRT/31065	CTRT/31065
Electronics	GEAR MOTOR 12V 200RPM 25GA	Lees Electronics	1	10.69	10.69	https://leeselectronic.com/en/product/41512.html	41512
	Cytron 13A, 5-30V Single DC Motor Controller	Cytron	1	11.76	11.76	https://www.robotshop.com/ca/en/cytron-13a-5-30v-single-dc-motor-controller.html	
	DC-DC Power Module 25W	DF Robot	1	8.50	8.50	https://www.dfrobot.com/product-752.html	DFR0205
	Rasperry Pi 3		1	34.17	34.17	https://www.buyapi.ca/product/raspberry-pi-3-model-b-armv8-with-1g-ram/?src=raspberrypi	Pi3B
	Arteck Battery	Arteck	1	44.21	44.21	https://www.amazon.ca/Arteck-12000mAh-Portable-Automotive-Flashlight/dp/B06XRWHBHN	
	6 Port USB Hub	Startech	1	56.99	56.99	https://www.startech.com/Cards-Adapters/USB-3.0/Hubs/6-Port-USB-3-USB-Combo-Hub-with-2A-Charging-Port-2x-USB-3-4x-USB-2~ST7320USBC	ST7320USBC
Total					311.82		

Figure 11: Bill of Materials

References

[Alcala et al., 2019] Alcala, J. M. L., Haagsma, M., Udell, C. J., and Selker, J. S. (2019). Hyperrail: Modular, 3d printed, 1–100 m, programmable, and low-cost linear motion control system for imaging and sensor suites. *HardwareX*, 6:e00081.

[Linder, 2009] Linder, W. (2009). *Digital photogrammetry*. Springer.

[Susko et al., 2018] Susko, A., Gilbertson, F., Heuschele, D., Smith, K., and Marchetto, P. (2018). An automatable, field camera track system for phenotyping crop lodging and crop movement. *HardwareX*, 4.