## MSE 4499 — Mechatronic Design Project

# CubeSat Ground Station

by

Jianhui Li, Tushar Mahajan, Paavan Raj, and Annie Wu

**Phase 2 Report**

Faculty Advisor: Dr. Sabarinathan and Dr. Cross

Date Submitted: February 14, 2019

Total Word Count: 3988

## Executive Summary

Western University's is designing a CubeSat satellite to be launched in 2021. An important component in accurately tracking the satellite is the ground station, which must be able to communicate in both S-band and UHF frequency bands. To accurately track the satellite, the ground station must be able to physically align the antenna with the satellite, monitor and predict the future positioning, and correct antenna direction. This overall scope of this project is to design a fully functioning ground station that will be able to track an object already orbiting in space with easily accessible data.

The concept generation process was split up into three components: the base, the actuation system, and antenna placement. The selected concept consisted of a truss base with an outer, protective box, a main hollow pole that would house the azimuth, with a platform that holds the elevation motor, with a coupler to connect to the antenna's arm. Throughout the fabrication and electrical component selection process, this design was iterated multiple times.

The design was validated through various calculations included the critical torque requirement of both motors, minimum acceleration of motors, steady state operational speeds, and tipping point. Prior to ordering materials, the design was validated by the University Machine Services. In terms of software design, the overall system architecture and is based on the Robotic Operating System that will run on the Raspberry Pi, which will interface with Arduino to control the devices. Various aspects of the required software have been built and tested with smaller scale electronics.

The feedback from Design Review 2 was to reconsider the base enclosure material. The original idea was to use sheet metal to ensure the base would be resistant to precipitation and the feedback was that this would be extremely expensive given the size of the base. The base will now be waterproofed by using a wood coating and applying an additional sealant at the edges. Another concern was that our high voltage electric circuits would need ESA approval, but this problem was solved by ordering an approved power supply that gives various voltage rails, from 5V to 24V.

In terms of project progress, there have been several delays in the shipping times of some of the more integral components, which have halted the build and testing process. However, to cover for this lost time, a heavier emphasis has been made on the software design, where great progress has been made. Moving forward, the goal is to finish constructing the ground station and implementing the electronics in the next few weeks before beginning to interface the software as well.

The purpose of this report is to provide necessary background information and context for the project, outline the choices made throughout the design process and the supporting engineering analyses, provide all details thus far with regards to the final design and design iterations, and outline a detailed plan of tasks moving forward.

# Table of Contents

Executive Summary ..... ii
Table of Contents ..... iii
List of Figures ..... iiv
List of Tables ..... v
Introduction ..... 1
Background Information ..... 1
Project Scope, Constraints, and Objectives ..... 2
Concept Generation and Selection ..... 2
    Ideation and Selection Process ..... 2
    Selected Concept ..... 3
Design Validation ..... 4
    Supporting Calculations ..... 4
        Elevation Motor Torque ..... 4
        Azimuth Motor Torque ..... 4
        Tipping Point Analysis ..... 6
    Concept Iterations ..... 6
    Operation of Design ..... 7
        System Architecture ..... 10
        User Interface ..... 10
        Data Flow ..... 12
        Stepper Motor Control Simulation ..... 13
Progress, Schedule, and Planning ..... 13
    Plan of Action ..... 13
    Contingency Plan ..... 14
Resources and Budget ..... 14
Conclusion and Recommendations ..... 16
References ..... 17
Appendices ..... 18

## List of Figures


Figure 1: Selected concept....................3

Figure 2: Freebody diagram of dish antenna....................5

Figure 3: Torque curve for NEMA 17 stepper motor....................6

Figure 4: Concept truss base....................7

Figure 5: Modified base....................7

Figure 6: Concept azimuth subassembly....................8

Figure 7: Modified azimuth subassembly....................8

Figure 8: Concept elevation subassembly....................9

Figure 9: Modified elevation subassembly....................9

Figure 10: Final design....................10

Figure 11: System architecture diagram....................11

Figure 12: User interface diagram....................12

Figure 13: Data flow diagram....................12

Figure 14: FEA Analysis showing displacement for a round tube with three points of contact....................18

Figure 15: FEA Analysis showing displacement for a square tube with two points of contact....................19

Figure 16: Stepper motor set-up for testing....................20

## List of Tables

Table 1: Detailed project task list, dates, and time allocation for the group…………..…….13
Table 2: Costs of materials from electronics shop ……………………………………….……...14
Table 3: Costs of materials from University Machine Services …………………..…..……...15
Table 4: Costs of general project resources………………..….. ……………………………..15

## Introduction

Western University's CubeSat Project is designing two parallel CubeSat satellite systems, with different communication frequency bands, that will undergo a trade-off, launching only one of them in 2021. They lack an accessible system on the ground that can provide real-time communication with the satellite, monitor and predict movement of the satellite, and correct antenna direction based on its trajectory. This ground system must be robust to either communication schemes.

The problem statement is to design and build a functional satellite tracking ground station with high alignment accuracy between the CubeSat in orbit and the ground station, to achieve better communication and data transfer.

## Background Information

A CubeSat is a miniature satellite intended for low earth orbit (LEO), with each unit measuring some multiple of 10x10x10 cm and weighing less than 1.33 kg [1]. Western's two CubeSat designs will communicate in different frequency bands – UHF and S-band, requiring two separate Yagi and parabolic dish antennae. The parabolic antenna will need to rotate between 15 to 165 degrees and operate around 2 GHz and a Yagi needs a 30-degree field operating around 300 MHz [2].

Simplified General Perturbations (SGP) models are used to model the orbit of LEO objects. The two-line element set is a data format that encodes the orbital elements of a satellite at a given point in time. The two-line elements that are created from the SGP orbit determination can be used to model an adequately accurate propagation of the CubeSats motion [4].

Though ground stations have been a well-known aspect of satellite use, there has been a burst of do-it-yourself ground stations in recent years, constructed from various off the shelf, commercial components. Innovation Solutions in Space is a company with a variety of ground stations designed specifically for small satellites in LEO. They have one ground station that has both Yagi and parabolic antennae and can track satellites using a steerable antenna system that can be remotely controlled through the internet, with speeds up to 6˚/sec [5]. This ground station is unique in how it can function in both S-band and UHF band frequencies; however, lacks the ability to be fully autonomous and is meant to track multiple satellites at once, lacking detailed information for any specific satellite. GOMSpace's AS2000 ground station also supports both S-band and UHF but has a fixed height and needs to be mounted high above ground in a flat area [6].

After researching many existing ground stations, they all seem to lack some design element such as autonomy and portability. Our ground station aims to fill this gap and since it is focused on tracking a single satellite, it will have higher precision and be able to provide more detailed information. Additionally, we aim to add another level of innovation by including a GPS module so that when the ground station is moved to various locations, it can automatically locate the coordinates and recalculate the positioning of the antenna, as well as various housekeeping sensors that will monitor the outside temperature and run the electronics on a maintenance routine as needed, so their performance does not decline in the cold weather.

## Project Scope, Constraints, and Objectives

The overall scope of this project is to design a fully functioning satellite tracking ground station for Western's CubeSat. This project will cover the electrical and mechanical subsystems for motor actuation and sensory data acquisition, control system for mitigating tracking error, and the orbital prediction algorithm that the ground station will employ to track the satellite's orbit. However, since the satellite will not be launched by the end of the semester, the plan is to accomplish successfully aligning with and tracking an orbiting object in space whose data is readily available, such as the moon or the International Space Station.

Some of the key constraints that we must follow are that the ground station must maintain high alignment accuracy with the satellite, be able to predict the orbital trajectory of the satellite, correct the antenna's direction and orientation, and finally, be able to withstand the external environmental conditions. The main project objectives are for the ground station to be easily transportable, require low maintenance, and be manufactured from as many off the shelf commercial components as possible.

## Concept Generation and Selection

### Ideation and Selection Process:

Over the course of the design process, many concepts and iterations were examined and fine-tuned to develop the most recent design. To begin, the system was broken down into three main subsystems: the base, the actuation system, and the antenna placement.

A strong foundation was needed, so the base was the first portion of the design that was optimized. The main requirements for the base were for it be wide, strong in compression, lightweight for portability, and able to enclose the electronics. The original concepts were a box base, a wide truss, and a tripod design. The truss structure allowed for the box surface to be supported while also better distributing the large, concentrated load. The box structure was kept as a housing since it would easily protect the inner electronics and systems from the elements.

The next system that was designed was the actuation system. The initial concepts were to have the motors mounted in the base and use pulleys, use a rotator controller or use two motors to mimic a rotator. The most recent concept features two stepper motors with coincident axes of rotation, where one motor is mounted in the pole such that its axis is vertical, and the other motor is mounted horizontally on a platform. The reason why the motors are orientated this way is because the two angles that would need to be controlled, the azimuth (horizontal angle) and the altitude (vertical angle) rotate along these axes, making the system easier to model. The altitude motor would be under considerable torque loads, so counterweights were used to mitigate the weight.

The final sub-assembly that was considered was the antenna placement assembly. The original concepts featured simultaneously mounting both a Yagi and parabolic dish; however, only one frequency band would be used. A more efficient, modular design was used to accommodate either individual antenna.

With all the subsystems conceptualized and designed, the integration of all three was simple. The main pole would run through the base and be supported by the truss structure. This hollow pole is large enough to house the azimuth motor, just underneath the platform holding the altitude motor. The platform rests on a bearing so it may freely rotate when the azimuth motor spins. The altitude motor is connected to the gearbox, which has an outer ring that functions similarly to a large coupler with an additional antenna attachment, that will work with either antenna.

## Selected Concept

The selected concept's design using the discussed sub-systems is below. To better show the more important details, all figures of the CAD model have the box base and antenna suppressed.

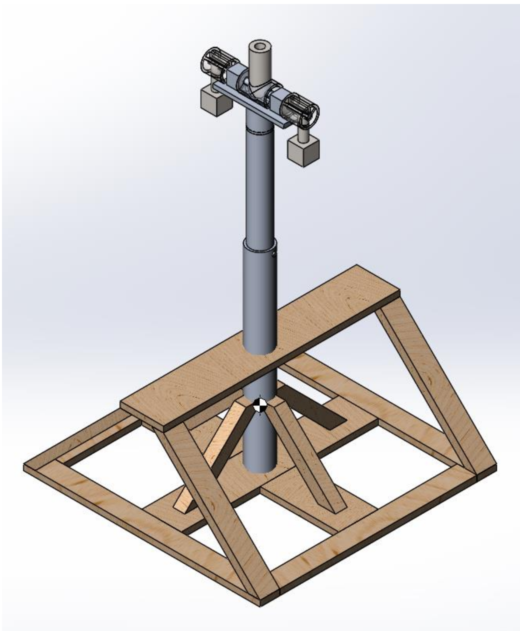

Figure 1: Selected concept

# Design Validation

## Supporting Calculations

### *Elevation Motor Torque*

Of the antenna possibilities, the parabolic dish is the heavier and more directional antenna, so it will be used as a basis for determining the required torque outputs. Approximating the mass of the antenna and the arm to be a 15 kg point mass acting halfway along the moment arm (0.20 m from the axis of the motor) (Figure 2 below), the following mass moment can be calculated:

$$\Sigma M = Fr = 15kg \times 9.81\frac{m}{s^2} \times 0.20\ m \times \cos(\alpha) = \ 29.43\cos(\alpha)\ N \cdot m \quad (1)$$

Given that the maximum uplink time is approximately 10 minutes [2], and the maximum angle that the azimuth motor would have to travel is 150° or $\frac{5}{6}\pi\ rads$ (15° is removed at either horizon due to signal power loss at low elevations), the maximum speed required is then:

$$\omega_{max} = \frac{\frac{5}{6}\pi}{10\ min} = 0.2618\frac{rad}{min} = 0.0417\ rad/s$$

From Equation 1, the maximum moment occurs when cos(α) = 1, or when the arm is horizontal. In addition to the mass moment, the motor will also need to overcome the inertial loads to begin moving, with a steady state operation of 0.042 rad/s from Equation 2 and inertial load of parabolic dish given as 0.5449 kg·m^2 (approximate calculation using a flat circle to model the dish). Additionally, an acceleration period from rest to a steady state operation of 0.1s is enough for tracking, so minimum required torque is then:

$$\tau = I_z\alpha$$

$$\alpha = \frac{d\omega}{dt} = \frac{0.043\frac{rads}{s} - 0\frac{rads}{s}}{0.1s - 0\ s} = 0.0043\frac{rads}{s^2}$$

$$\tau = 0.5449kg \cdot m^2 \times 0.043\frac{rads}{s^2} = 0.023\ N \cdot m \quad (2)$$

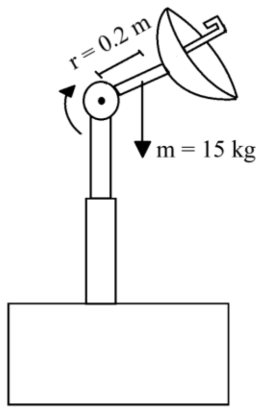


Figure 2: Freebody diagram of dish antenna

During steady operation, since the angular speed stays constant, the motor will not need to supply additional torque as there will be no acceleration.

Finally, the external force of wind has a large impact on the torque required to actuate the dish. In Canada, the maximum windspeed recorded is approximately 80 kph, which will serve as the worst-case scenario. Using the surface drag equation of a body in a fluid [7], the force exerted on the antenna when the face is at 45˚ to the wind can be found using the following equations:

$$F_{wind} = \frac{1}{2}\rho_{air} v_{wind}^2 A_{dish}$$

$$\rho_{air} = 1.2\ \frac{kg}{m^3}$$

$$v_{wind} = 80\frac{km}{h} = 22.2\frac{m}{s}$$

$$A_{dish} = 0.622\ m^2 \cdot cos45° = 0.439.82\ m^2\ (antenna\ tilted\ slightly\ up)$$

$$F_{wind} = \frac{1}{2} \times 1.2\ \frac{kg}{m^3} \times \left(22.2\frac{m}{s}\right)^2 \times 0.439m^2 = 130.0\ N$$

And the torque applied is thus:

$$\tau = Fr = 130.0\ N\ \cdot 0.4\ m\ \cdot \cos 45° =\ 36.8\ N \cdot m \quad (3)$$

With the values derived in Equations (1), (2) and (3), the maximum required torque (worst case scenario) is the summation of the three, which gives a critical torque of 66.25 N·m. With a 160:1 gear reduction however, the motor will only need to supply $1/160^{th}$ of this, therefore the maximum required torque is 0.414 N·m. The motor can provide this maximum torque, as evidenced by the torque curve from the motor specifications below.

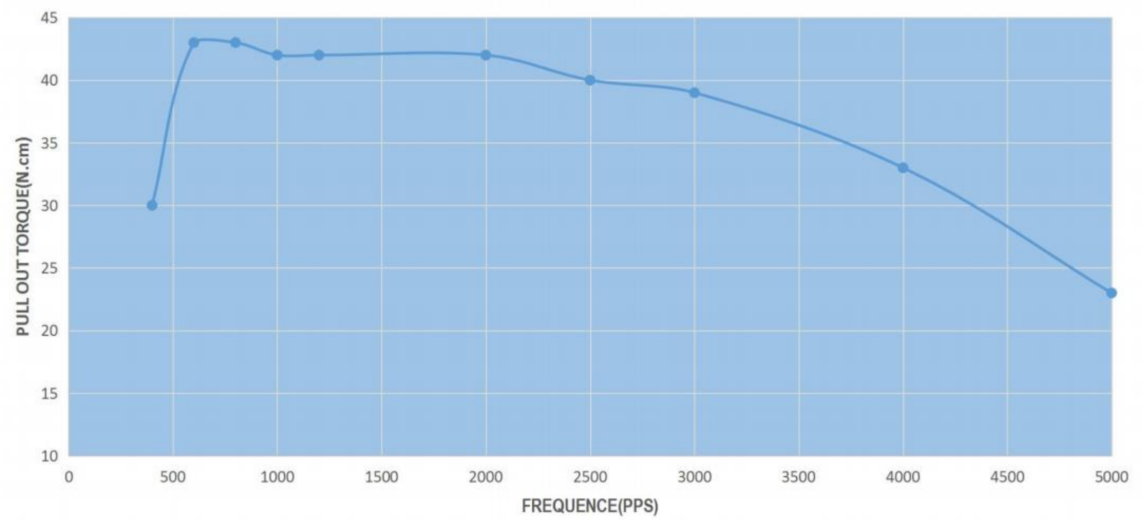

Figure 3: Torque curve for NEMA 17 stepper motor [8]

***Azimuth Motor Torque***

Since the axis of motion for this motor is vertically oriented, it experiences no mass moments from the rest of the components and the load is supported by a bearing, eliminating much of the friction that would oppose rotation. The only significant moment would then be from the wind loads, so assuming a similar worst case load as in Equation (1), the torque required for this motor would be 0.23 N· m.

***Tipping Point Analysis***

With the pole height and the significant wind force, a critical failure mode would be tipping. With pole height being 1.4m, and the entire mass of the system being 55 kg at the center of mass, only the wind force is needed before the tipping point can be calculated. The max wind force would be when the antenna faces the wind fully.

$$A_{dish} = 0.622\ m^2 * \cos(0) = 0.622\ m^2$$

$$F_{wind} = \frac{1}{2} \times 1.2\ \frac{kg}{m^3} \times \left(22.2 \frac{m}{s}\right)^2 \times 0.622 m^2 = 183\ N$$

$$M_{wind} = \frac{L_{base}}{2} \cdot m_{total} \cdot g$$

$$1.4m\ \cdot 183N = \frac{L_{base}}{2} * 55\ kg * 9.81 \frac{m}{s^2}$$

$$L_{base} = 0.9497\ m$$

This result is the minimum length each side of the base needs to be so that the ground station will not tip over.

## Concept Iterations

The chosen concept for the base included a telescoping pole design that allowed the height of the pole antenna to be adjusted as needed and allowed for easier transportation. This design used two circular metal tube whose diameters and tolerances would allow a loose running fit. The pole would be fastened to the base using custom, machined flanges. To reduce the cost and amount of labour, the telescoping poles were replaced with a 2” wide square metal tube. This iteration is easier to mount as angle brackets can be used. These angle brackets are inexpensive and can rigidly secure the square tube to the base. The square metal tubing is also relatively inexpensive so if a longer or shorter pole is required, another pole can be easily obtained.

The base design was further refined, as the selected concept used butt joints for all connections and required trusses to secure the larger tube into place (see Figure 4). The connections using angle brackets between the square tube and the top and bottom horizontal sections of wood have a larger surface area than the connections in the concept originally selected. Therefore, it was found that the additional trusses added between the horizontal sections of wood would be redundant. This was confirmed using an FEA simulation that used three points of contact with the circular tube and two point of contact with the square tube (See Appendix I). It was determined that the deflections in the square tube would be lower than the deflections in the circular tube. In order to increase the strength of the base, all butt joints were replaced with finger joints (see Figure 5).

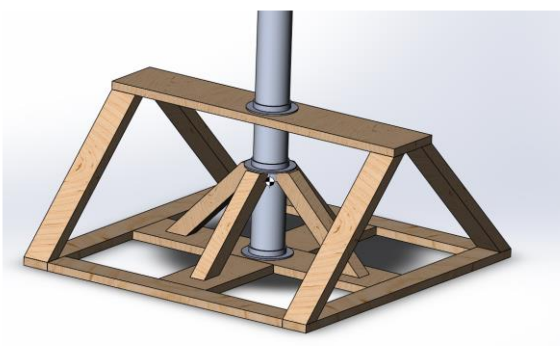

Figure 4: Concept truss base

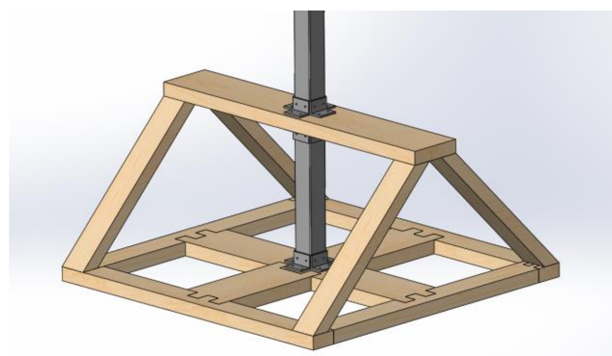

Figure 5: Modified base

The azimuth subassembly design used two machined components. The motor would be fastened to the bottom component and a thrust bearing would be placed on top of the motor, bearing axial loads exerted by the upper component (See Figure 6). However, this design did not consider the radial loads, so two ball bearings were added to the shaft to accommodate that load [9]. The original design made the motor inaccessible without disassembly, as it was fully enclosed. The entire azimuth subassembly was redesigned to better mount the subassembly to the pole and the shaft to the upper component as seen in

Figure 7 below. The square tube is connected to the motor using a circular plate that is welded onto a small section of larger square tube. The motor is directly mounted to this piece. A 4" long metal rod is machined to hold the radial and thrust bearings in place. It is welded to a circular plate and will be fastened to the motor plate. The bearings are press fit into this piece and the shaft is pushed through it. The top plate that holds the elevation subassembly is machined from a 1/4" thick metal plate so that 1/8" of the plate rests on the thrust bearing and is inside the metal rod that holds the bearings. The top plate has a hole drilled into the center where the shaft is welded to it. This design allows the components to be machined easier while reducing axial and radial forces on the shaft.

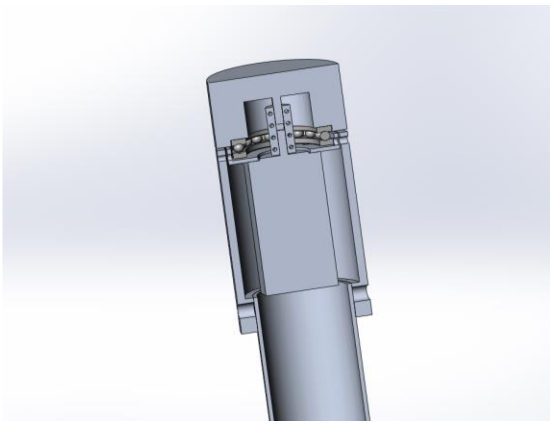

Figure 6: Concept azimuth subassembly

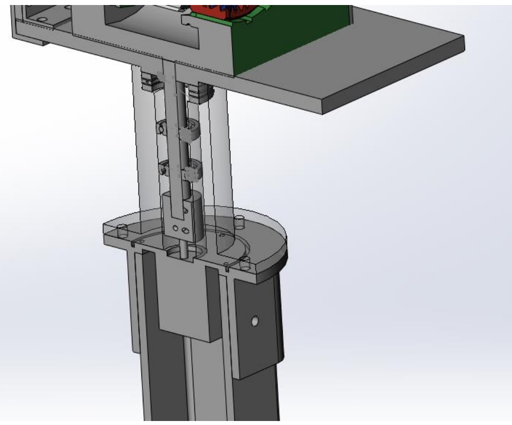

Figure 7: Modified azimuth subassembly

The elevation subassembly in the selected design used counterweights to reduce the amount of torque needed from the motor. However, this design required three large, custom couplers on the shaft and linear bearings in the top plate (see Figure 8). As this would add considerable expenses to the project, it was replaced with a 3D printed 160:1 planetary gearbox (see Figure 9). From Equation (3) previously, the elevation subassembly requires approximately 40 Nm of torque during periods of high winds. This requires approximately 0.1 rotation per minute.

$$T_{total} \approx 40Nm$$

$$RPM_{\max} = \frac{1}{2}\text{rotation with a period of} \sim 5 \text{ minutes} = 0.1\ RPM$$

The motor selected can provide 0.4 N·m of torque at 160 RPM (refer back to Figure 3) which outputs 64 N·m of torque at 1 RPM through the gearbox. The output is taken at 1 RPM to allow for faster positioning of the antenna during the setup phase. Therefore, the selected motor will be able to drive the gearbox even in the worst-case scenario as the output torque will not exceed 40 N·m. Open source testing of the gearbox shows it can handle torques in excess of 40 N·m, so it is unlikely to fail under these loads [10].

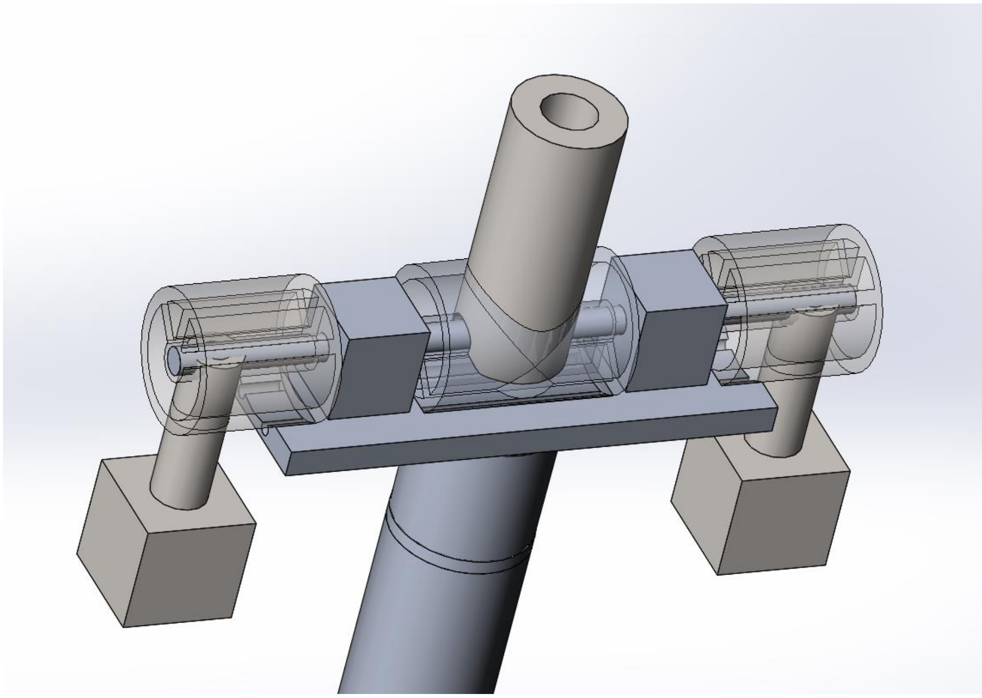

Figure 8: Concept elevation subassembly

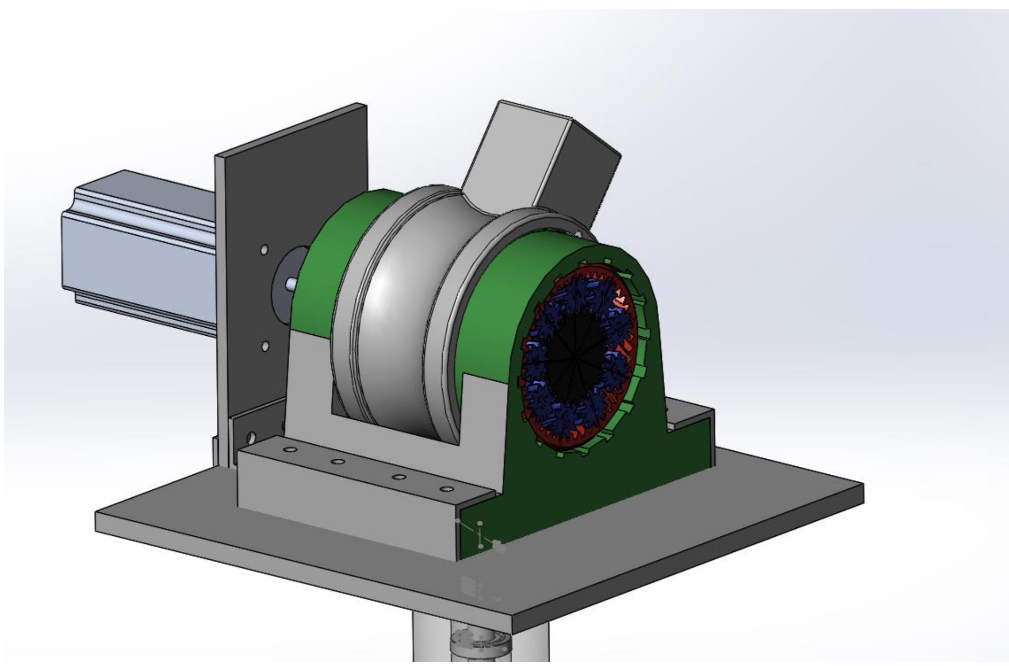

Figure 9: Modified elevation subassembly

The final design using the discussed subassemblies is below:

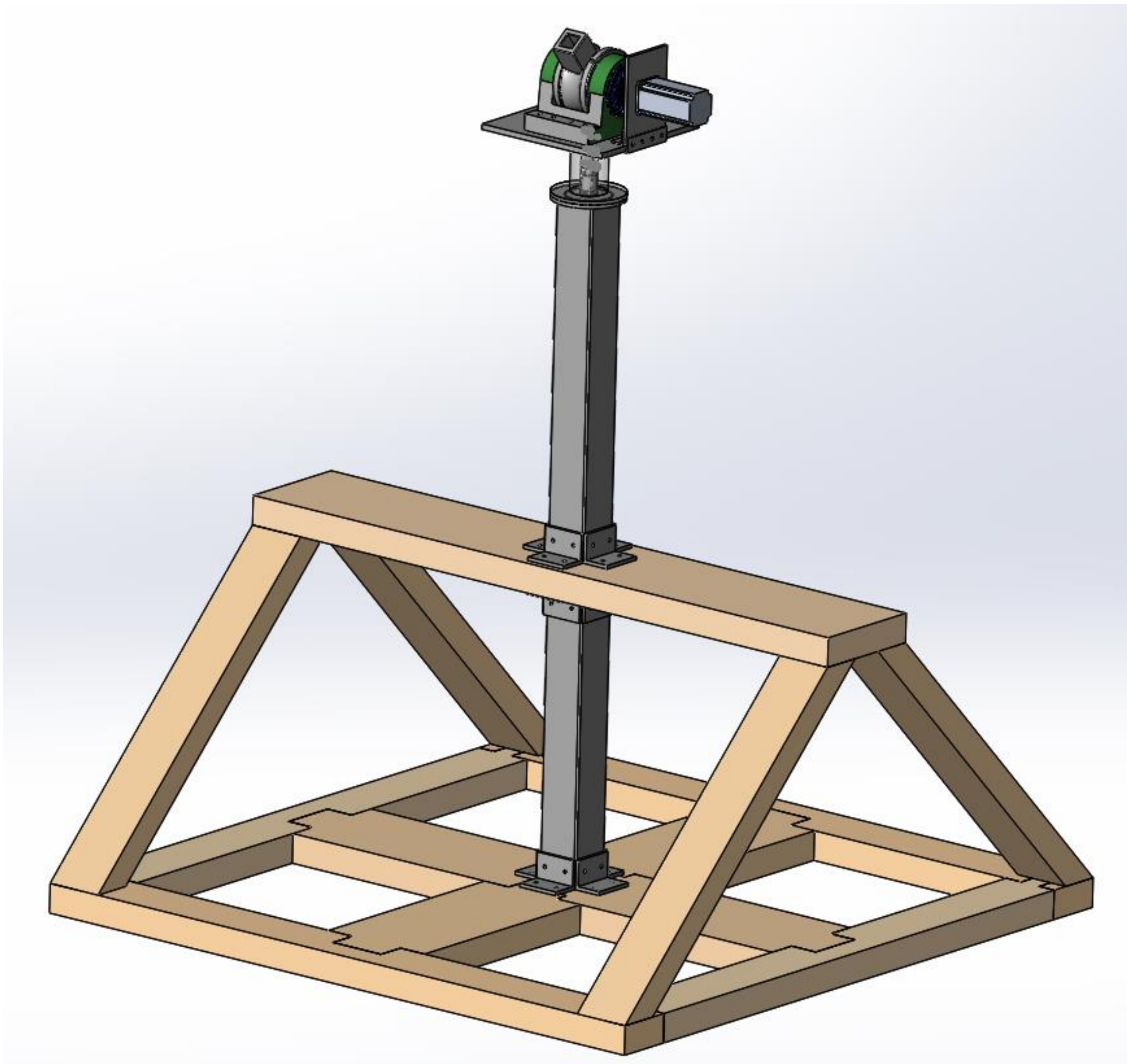

Figure 10: Final design

## Operation of Design

### *System Architecture*

In our software design, the core architecture is based on Robotic Operating System (ROS), an existing robotics middleware. ROS provides the services from an operating system, including low-level device control, implementation of commonly-used functionality, and package management. ROS is language-neutral and can be programmed in various languages. The kinetic version of ROS is installed on the Raspberry Pi (RPI), which can fit the Linux system Raspbian Sketch. There are three main parts in the system architecture: the control of devices through the Arduino board, CubeSat orbit prediction using Simplified General Perturbations models (SGP4 Python 1.4), and CubeSat location. As seen in Figure 11 below, the three parts work together as an individual node – the core ROS Master node.

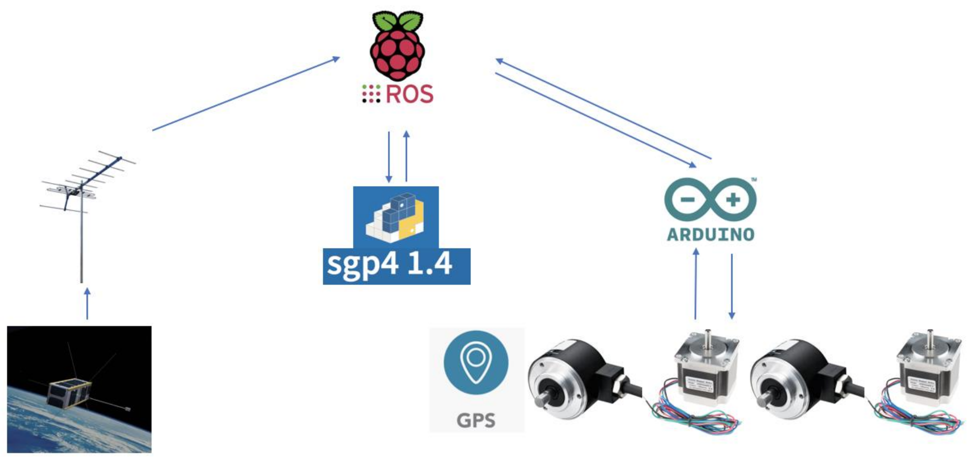


Figure 11: System architecture diagram

In the first part, the antenna works as a publisher. After the CubeSat is located, the initial position data sends a message to the RPI. Using the two-line element (TLE) data, angle of antenna can be calculated and set. Next, the SGP4 1.4 Python package receives the initial position data from the first node and calculates the next overhead time and position of CubeSat. When the RPI receives the next overhead position of CubeSat, the program will calculate the intersection angle between ground station and CubeSat. In the final part, the GPS module will get the current position of the ground station and send this information back to ROS master. Then, the Arduino board will control the stepper motors, which will utilize ROS topics to send goal messages to the server. While the antenna is moving to the goal position, the 'percent_complete' message periodically transmits feedback values showing the progress in the form of the percentage of the goal point reached. Rosserial needs to be installed on the RPI, which will provide an ROS communication protocol that works over the Arduino's Universal Asynchronous Receiver/Transmitter. In this project, the electronic devices connect to the Arduino board instead of the RPI, because the Arduino board is a microcontroller motherboard that can simply run one program at a time, over and over again, and only one RPI board cannot support sufficient general-purpose input/output (GPIO) pins to connect these devices. In addition, the Raspberry Pi 3B+ has four USB 2.0 ports; therefore, it can expand communication with up to four Arduino boards.

### *User Interface*

The user interface will enable the user to have simple control and be able to set tasks for the ground station. A convenient method to get remote access to the RPI is through Secure

Shell (SSH). With this method, users can access to the Linux control tools including the Raspbian desktop system and Linux console. There are two choices of the telnet clients, PuTTY and REALVNC, and the SSH connection can be through Ethernet or mobile Wi-Fi. The RPI are set as fix IP address 192.168.137.214 for Ethernet connection, and 192.168.137.245 for the WIFI connection. The user interface is illustrated in Figure 12 below.

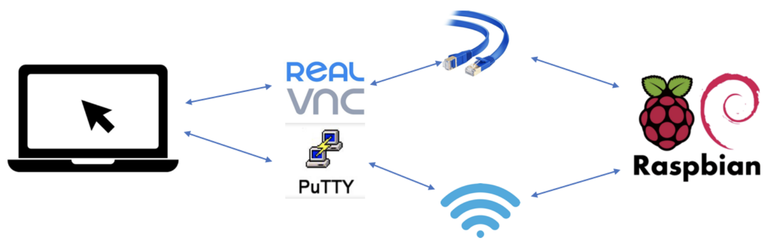


Figure 12: User interface diagram

***Data Flow***

The primary design for data flow is based on ROS. There are six nodes created for six different tasks, which can be seen in Figure 13 below.

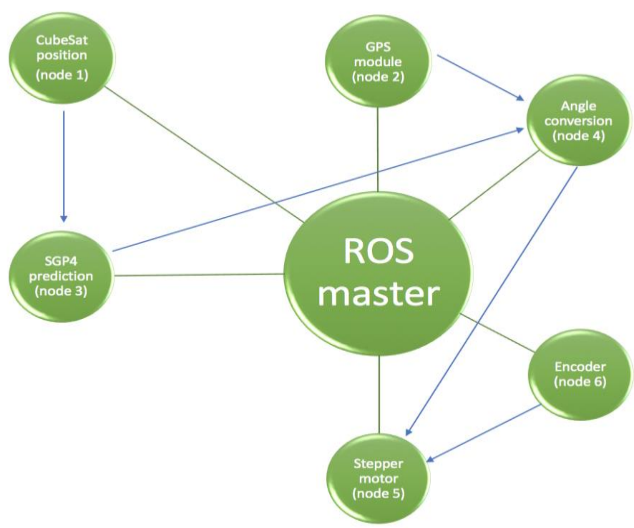


Figure 13: Data flow diagram

The ROS Master provides naming and registration services to the rest of the nodes in the ROS system. It tracks publishers and subscribers to topics as well as services. The role of the Master is to enable individual ROS nodes to locate one another. Once these nodes have located each other they communicate with each other peer-to-peer [11]. The GPS module (node 2) publishes the position of the ground station and sends this message to Angle conversion (node 4) as a subscriber. CubeSat position (node 1) sends the CubeSat position and speed to the SGP4 prediction model (node 3). After calculations are complete, node 3 publishes the next overhead position to node 4, which will convert the position of CubeSat to the earth-based coordinate system and publish the rotation angles for each of the stepper motors. The stepper motors (node 5) will rotate to their desired angles. The encoder (node 6) publishes the angle value to node 5, to correct the error.

#### *Stepper Motor Control Simulation*

Simulations were run to mimic the control of the stepper motor. The RPI sends a message to the topic “stspeed” (stepper motor speed), and the Arduino program uses this to control the speed of the motor. The simulation set up requires the RPI to have the Raspbian Sketch system installed, which is highly optimized for the RPI line’s low performance Advanced RISC Machines CPUs. ROS kinetic and Rosserial Arduino are both installed on this RPI. The RPI was controlled through SSH using the REALVNC client. The detailed steps to run the software simulation can be found in Appendix II. The result from this simulation was very successful in mimicking the ground station system. The RPI work was successfully monitored through SSH and run on ROS to communicate with the Arduino to control a stepper motor. These simulations were first run on very small stepper motors with weak specifications. Once received, the same simulation was run on the actual selected motors. The simulation was extremely easy to transfer and yielded equally successful results.

## Progress, Schedule, and Planning

### Plan of Action

The project has run delayed from our original plan of action. There are many reasons for these delays. First, the long design iteration process was not accounted for. Some machinability and mechanical design concerns were brought up in the initial meeting with the University Machine Services (UMS). This required redesigning and the back and forth with the UMS prior to the ordering of materials was unexpected. Also, though shipping delays were considered in the original project plan, the components have been coming in individually and we did not consider that some components depend on others to operate. Taking these and any future delays into account, a new plan of action was created and can be seen below.

Table 1: Detailed project task list, dates, and time allocation for the group

| Task | Start Date | End Date | Hours |
|---|---|---|---|
| Receive all ordered components | 2019-01-30 | 2019-02-22 | - |
| Complete mechanical build | 2019-02-05 | 2019-02-25 | 8 |
| Incorporation and testing of electrical components | 2019-02-22 | 2019-03-03 | 12 |
| Completion of software components | 2019-02-05 | 2019-03-07 | 30 |

| | | | |
|---|---|---|---|
| Integrate software with hardware | 2019-03-07 | 2019-03-15 | 15 |
| Testing of completed system with existing orbiting objects | 2019-03-15 | 2019-04-14 | 45 |
| Detailed design documentation | 2019-01-30 | 2019-03-14 | 25 |
| Final Report | 2019-03-15 | 2019-04-11 | 15 |

The first stage in our plan moving forward is to finish the foundation by attaching the main pole, and the elevation and azimuth motor holders. As the electrical components have been coming in, we have been testing and incorporating them into our system architecture. All electrical components will then be interfaced with the mechanical components. Throughout the semester, the software design has been split up into various components, completed, and tested using hardcoded values. The next step is to integrate the software with all the hardware and test the completed design. As well, we have made it a priority to stay on top of documenting the entire design process, to ease the work required for remaining deliverables to stay focused on iterating and testing.

### Contingency Plan

It is recognized that as the project progresses, it may be hard to follow this outlined schedule. To keep the project on track, all group members will review each member's progress according to this schedule at the weekly group meetings. Changes and adjustments to the phase periods and hours allotted will be made at these meetings. We are all prepared to dedicate the time necessary to complete this project. Additional weekly meetings occur between all CubeSat project teams to ensure all projects are on track and the use of Slack will be highly leveraged to stay in constant contact with each team's progress.

## Resources and Budget

The course allocates a budget of $75 per person, totalling $300 for a group of four. There is also a budget of $100 of free FDM for rapid prototyping from the Electronics shop. In terms of additional supervisor funding, there is a minimum of $1000 available for the ground station, with the overall Western CubeSat Project having a budget of $200,000. The detailed budget can be seen in the following tables, split up into the different places parts were ordered from.

Table 2: Costs of materials from electronics shop

| Item | Quantity | Cost/Unit | Shipping Cost | Total Cost |
|---|---|---|---|---|
| Rotary Encoder | 2 | $76.19 | $0.00 | $152.38 |
| GPS Module | 1 | $20.99 | $9.81 | $30.80 |
| Raspberry Pi 3B+ | 1 | $94.95 | $13.95 | $108.90 |

| | | | | |
|---|---|---|---|---|
| Nema 17 Motors | 3 | $7.33 | $24.64 | $46.63 |
| Stepper Motor Drivers | 2 | $15.56 | $0.00 | $31.12 |
| EVGA Power Supply | 1 | $79.99 | $0.00 | $79.99 |
| | | | Subtotal | $449.82 |
| | | | Tax | 13% |
| | | | **Final Total** | $508.30 |

Table 3: Costs of materials from University Machine Services

| **Item** | **Quantity** | **Cost per Unit** | **Total Cost** |
|---|---|---|---|
| 2" x 8" x 12' Wood | 1 | Exact breakdown unknown | |
| 2" x 4" x 8' Wood | 4 | | |
| 4' x 8' Wood Weather Treated | 2 | | |
| **UMS Wood Materials Cost** | | | $119.39 |
| 3" x 3" x 4' (1/4" thick) Square Metal Tube | 1 | Exact breakdown unknown | |
| 1" Steel Angle, 1/8" thick x 60" | 1 | | |
| 1" diameter, 1" long Metal Rod | 1 | | |
| 1/2" diameter, 4" length Shaft | 1 | | |
| 1" x 1" x 1", 12" long Z Bracket | 1 | | |
| 6" x 4", 1/4" Thick Steel Plate | 1 | | |
| Low-Carbon Steel Disc 1/2" Long, 5" Diameter | 2 | | |
| 4.5" long, 2.5" diameter Steel Rod | 1 | | |
| 10" x 10" x 1/2" Steel Plate | 1 | | |
| 1/2" long, 3" diameter Low-Carbon Steel Disc | 1 | | |
| **UMS Metals Estimate** | | | $87.86 |
| Thrust Ball Bearing | 1 | $22.25 | $22.25 |
| Ball Bearings | 2 | $11.12 | $22.25 |
| | | **Final Total (incl. Tax)** | $251.75 |

Table 4: Costs of general project resources

| **Resource** | **Rate** | **Estimated Cost** |
|---|---|---|
| Engineering Time | 880 hours at $50/hour | $44,000 |
| SolidWorks[12] | Standard license for each member 4x$8000 | $32,000 |

| | | |
|---|---|---|
| MATLAB & Simulink[13] | Standard license for each member 4x$(2850+4300) | $28,600 |
| Microsoft Office[14] | Standard license for each member 4x$200 | $800 |

## Conclusion and Recommendations

Since the last deliverable, we have run into quite a few roadblocks with having to redesign components because of the high costs to machine them and delays in receiving ordered parts. The new design has been validated and the final product is currently being constructed. We have compensated for the delay in our electronic components by placing a stronger focus on the software aspect of the project. Our next steps are to finish the construction of the ground station, integrating the mechanical and electrical components, and complete sub-assembly testing for the upcoming weeks. We will continue to work on the software components, and finally, the software and hardware components will be interfaced and testing of the overall ground station will commence. While behind according to our original Gantt chart, we have shifted timelines and priorities to still be on track to complete the project.

## References

1 What is a CubeSat. (2018, November 27). Retrieved from http://www.asc-csa.gc.ca/eng/satellites/cubesat/what-is-a-cubesat.asp

2 Cakaj, S., Kamo, B., Lala, A., & Rakipi, A. (2014). The Coverage Analysis for Low Earth Orbiting Satellites at Low Elevation. (IJACSA) International Journal of Advanced Computer Science and Applications, 5(6). doi:http://thesai.org/Downloads/Volume5No6/Paper_2-The_Coverage_Analysis_for_Low_Earth_Orbiting_Satellites_at_Low_Elevation.pdf

3 E., V. W. (1994). Mitigation of EMI generated by a variable frequency drive controller for an AC induction motor (Unpublished master's thesis). Naval Postgraduate School, September. doi:https://apps.dtic.mil/dtic/tr/fulltext/u2/a286045.pdf

4 TS Kelso, "Models for Propagation of NORAD Element Sets", 1980. Available: http://www.celestrak.com/NORAD/documentation/spacetrk.pdf

5 Full Ground Station Kit VHF/UHF/S-band. (n.d.). Retrieved from https://www.isispace.nl/product/full-ground-station-kit-for-vhfuhfs-band/

6 S-band Ground station antenna for LEO satellites. (n.d.). Retrieved from https://gomspace.com/shop/subsystems/ground-systems/nanocom-as2000.aspx

7 The Drag Equation. (n.d.). Retrieved from https://www.grc.nasa.gov/www/k-12/rocket/drageq.html

8 17HS19 Motor Torque Curve. (n.d.). Retrieved from https://www.omc-stepperonline.com/download/17HS19-2004S1_Torque_Curve.pdf

9 J. E. Shigley and C. R. Mischke, Mechanical engineering design. Boston: McGraw-Hill, 2001.)

10 High Torque 160:1 Compound Planetary Gearbox. (n.d.). Retrieved from https://www.thingiverse.com/thing:2375916

11 ROS Documentation. (n.d.). Retrieved from http://wiki.ros.org/Master

12 Taylor, C. (2018, December 05). SOLIDWORKS Price. Retrieved from https://www.cati.com/solidworks-price/

13 MathWorks Pricing and Licensing. (n.d.). Retrieved from https://www.mathworks.com/pricing-licensing.html

14 Microsoft Office Cost Comparison. (n.d.). Retrieved from https://products.office.com/en-us/compare-all-microsoft-office-products?tab=1

## Appendix I – Software Simulation Step

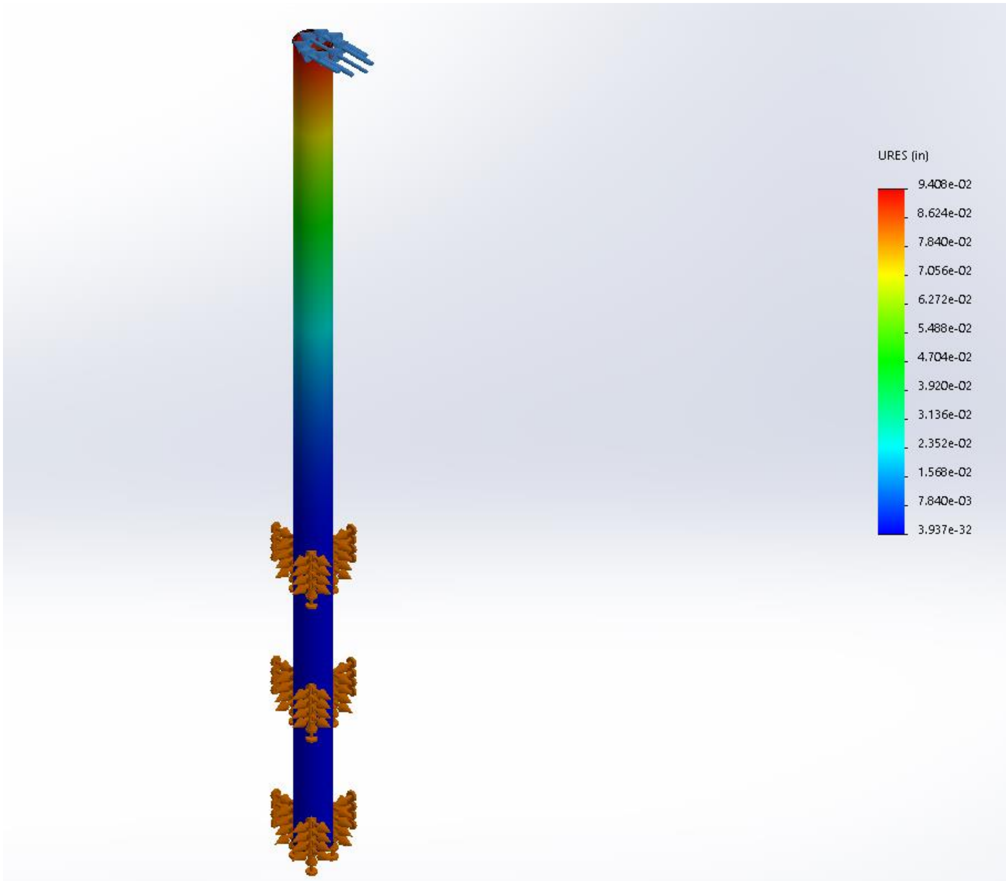


Figure 14: FEA Analysis showing displacement for a round tube with three points of contact

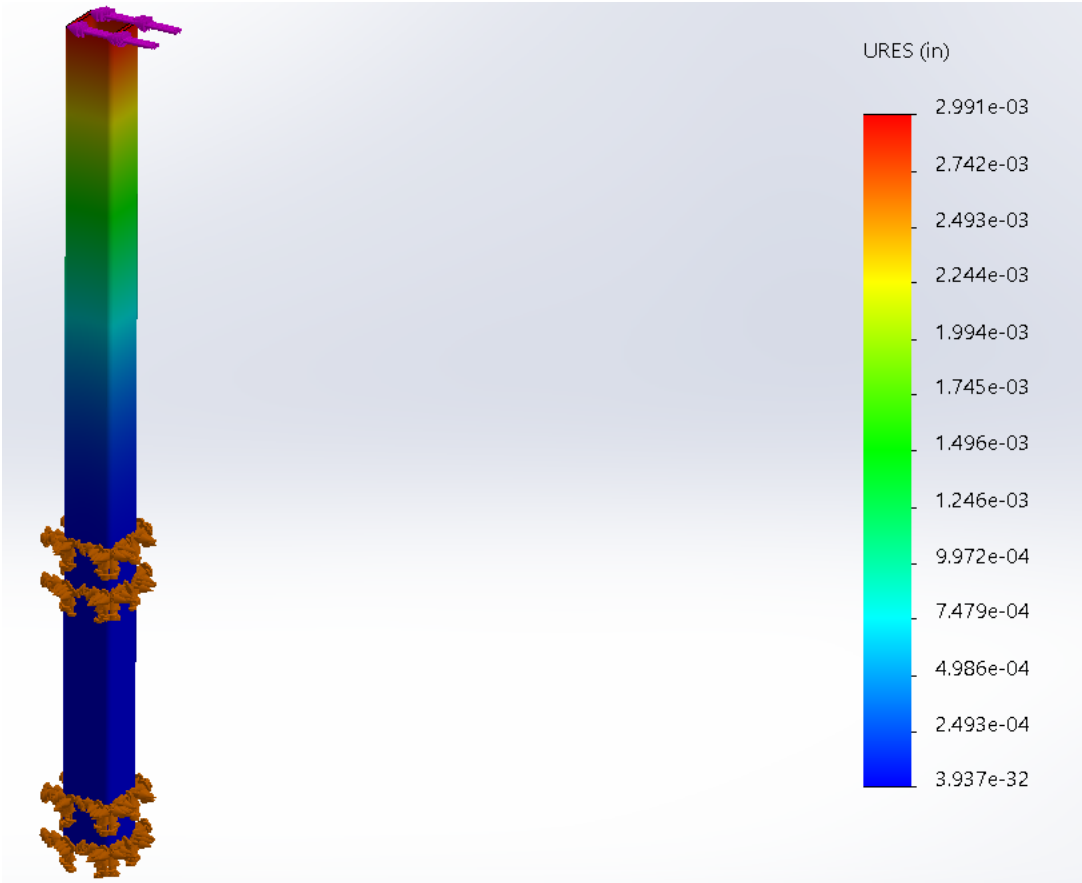


Figure 15: FEA Analysis showing displacement for a square tube with two points of contact

The displacement during extreme weather conditions of the square tube with two points of contact (0.0029 inches) was less than the displacement of the circular tube with three points of contact (0.094 inches). Therefore, the trusses would be redundant if used with the square tube.

## Appendix II – Software Simulation Steps

Once all the required systems are installed on the RPI, follow the steps below for the stepper motor simulation:

1) Connect the components according to the diagram below.

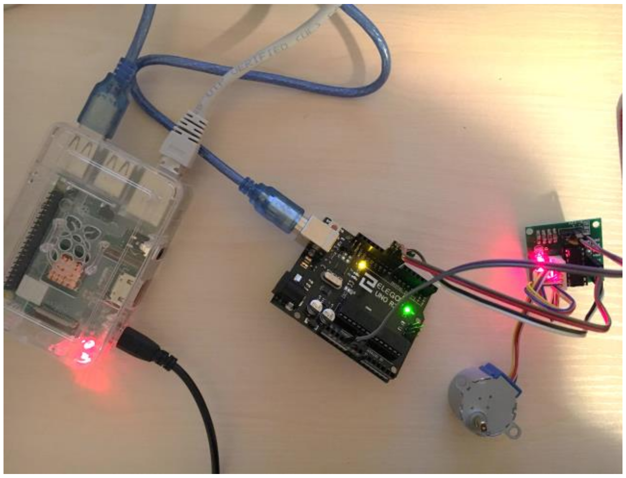

Figure 16: Stepper motor set-up for testing

2) Upload the following code to the Arduino board:

```
#include <Stepper.h>
#include <ros.h>
#include <std_msgs/UInt16.h>

ros::NodeHandle  nh;

const int stepsPerRevolution = 200;
int motorSpeed;
// initialize the stepper library on pins 8 through 11:
Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);
void stepper_cd( const std_msgs::UInt16& cmd_msg){
  motorSpeed=cmd_msg.data; //set speed of stepper
}

ros::Subscriber<std_msgs::UInt16> sub("stspeed", stepper_cd);
void setup() {
  // nothing to do inside the setup
  nh.initNode();
  nh.subscribe(sub);
}
void loop() {

  if (motorSpeed > 20) {
     myStepper.setSpeed(30);
     myStepper.step(stepsPerRevolution / 100);
```

```
 else
 {
  myStepper.setSpeed(10);
     myStepper.step(stepsPerRevolution / 100);
  }
   nh.spinOnce();//all ros callback
     delay (1);
}
```

3) Connect the motor to pins 8, 9, 10, and 11

4) Connect the Arduino with the RPI through a USB port. Find the right device port through ls –l /dev/, and then using the command: rosrun rosserial_python serial_node.py /dev/ttyACM0. The default baud frequency is 57600 Hz.

5) Publish a topic using: rostopic pub stspeed std_msgs / Unit 16 (data)

6) Run rpt_graph to see if the node connection is correct.