NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

CUBESAT PASS QUALITY ANALYSIS AND PREDICTIVE MODEL

by

John J. Leone III

June 2018

Thesis Advisor:	James H. Newman
Second Reader:	Giovanni Minelli

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2018	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE
CUBESAT PASS QUALITY ANALYSIS AND PREDICTIVE MODEL

5. FUNDING NUMBERS

6. AUTHOR(S) John J. Leone III

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

The current ground station infrastructure of the Naval Postgraduate School (NPS) Mobile CubeSat Command and Control (MC3) Network can sufficiently manage the existing three CubeSats that it has been tasked to monitor and control. However, the projected growth in the quantity of these satellites will produce greater likelihood of conflicting passes, a condition where multiple CubeSats are above the horizon simultaneously with respect to a single ground station and are competing for limited downlink antenna assets. Consequently, as more CubeSats are placed in orbit, the ground stations that communicate with them need to become more efficient and require optimization strategies to make best use of the network. Using the MC3 historical data set for its PropCube satellites, this thesis creates a grading scale that can be used to prioritize which PropCube should have antenna priority based on the expectation of downlinking data and when during the satellite's pass an operator should expect to capture that data. This value function can be used by an optimization program, such as that being developed in the NPS Small Satellite Laboratory, to maximize the data retrieved from a constellation of satellites. Additionally, the tools developed for analyzing and understanding MC3 PropCube passes can be used to determine value functions for other ground station networks in order to efficiently schedule CubeSat contacts.

14. SUBJECT TERMS
Mobile CubeSat Command and Control, MC3, Satellite Operations Center, SOC, Cube Satellite, CubeSat, PropCube, conflicting pass, antenna prioritization, pass prioritization, ground station optimization, saturation, scheduling, pass density, pass quality metric, pass value function, predictive model, benefit value function, whole pass analysis, segmented pass analysis, ascending pass, descending pass, preamble, decode, noise environment, free space loss, interference, MATLAB, STK

15. NUMBER OF PAGES
143

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 | Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

CUBESAT PASS QUALITY ANALYSIS AND PREDICTIVE MODEL

John J. Leone III
Major, United States Marine Corps
BS, U.S. Naval Academy, 2008

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SPACE SYSTEMS OPERATIONS

from the

NAVAL POSTGRADUATE SCHOOL
June 2018

Approved by: James H. Newman
Advisor

Giovanni Minelli
Second Reader

James H. Newman
Chair, Department of Space Systems Academic Group

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The current ground station infrastructure of the Naval Postgraduate School (NPS) Mobile CubeSat Command and Control (MC3) Network can sufficiently manage the existing three CubeSats that it has been tasked to monitor and control. However, the projected growth in the quantity of these satellites will produce greater likelihood of conflicting passes, a condition where multiple CubeSats are above the horizon simultaneously with respect to a single ground station and are competing for limited downlink antenna assets. Consequently, as more CubeSats are placed in orbit, the ground stations that communicate with them need to become more efficient and require optimization strategies to make best use of the network. Using the MC3 historical data set for its PropCube satellites, this thesis creates a grading scale that can be used to prioritize which PropCube should have antenna priority based on the expectation of downlinking data and when during the satellite's pass an operator should expect to capture that data. This value function can be used by an optimization program, such as that being developed in the NPS Small Satellite Laboratory, to maximize the data retrieved from a constellation of satellites. Additionally, the tools developed for analyzing and understanding MC3 PropCube passes can be used to determine value functions for other ground station networks in order to efficiently schedule CubeSat contacts.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

θ	true anomaly
λ	wavelength
ω	argument of perigee
ϕ	latitude
a	semi-major axis
AER	azimuth, elevation, range
AFIT	Air Force Institute of Technology
AOS	acquisition of signal
AZo	initial azimuth
AZf	final azimuth
BVF	Benefit Value Function
CPA	closest point of approach
CRC	cyclic redundancy check
CSV	comma-separated values
CubeSat	cube satellite
dB	decibel
e	eccentricity
HSFL	Hawaii Space Flight Laboratory
i	inclination
KISS	"keep it simple, stupid" communication protocol
km	kilometer
LEO	low earth orbit
LOS	loss of signal
L_S	free space loss
MATLAB	Matrix Laboratory
Max El	maximum elevation
MC3	Mobile CubeSat Command and Control
MHz	megahertz
MLB	Melbourne
NPS	Naval Postgraduate School

PropCube	Picosats Realizing Orbital Propagation Calibrations using Beacon Emitters
PTSUR	Point Sur (ground station currently located at NPS)
RF	radio-frequency
SDL	Space Dynamics Laboratory
SOC	satellite operations center
SQL	structured query language
STK	Systems Took Kit
TLE	two-line element set
UAF	University of Alaska, Fairbanks
UNM	University of New Mexico
USCGA	United States Coast Guard Academy
USNA	United States Naval Academy
V	BVF scaling value

ACKNOWLEDGMENTS

I would like to express gratitude to my advisor, Dr. James Newman, and second reader, Giovanni Minelli, for their continuous support throughout my research. Without their knowledge, patience, and enthusiasm, this study would not have been possible. I could not have asked for better mentors during the thesis process.

I also wish to acknowledge the assistance provided by my fellow classmates, NPS instructors, and Graduate Writing Center staff for their guidance and useful critiques of this research work.

To my parents, Sandra and John, thank you for the love and encouragement you have provided through my entire life. Without you, I would not be the person I am today.

Most importantly, I would like to thank my wife, Lauren, and son, Johnny, for their love, patience, and understanding, as they allowed me to spend countless hours away from home to complete this work. They are the most important people in my world, and I dedicate this thesis to them.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

In recent years, Cube Satellites (CubeSats) have increasingly become a subject of interest for commercial, military, and academic institutions due to their cost-effective nature, numerous opportunities for research, and availability of launch options [1]–[4]. The current ground station infrastructure of the Naval Postgraduate School (NPS) Mobile CubeSat Command and Control (MC3) Network can sufficiently manage the existing three CubeSats that it has been tasked to monitor and control. However, the projected growth in the quantity of these satellites in MC3, and networks like it, may soon overwhelm the current ground support systems, and then communication will be constrained by simultaneous satellite passes. The underlying assumption is that a ground station antenna can only service one, or few satellites at a time due to its use of directional antennas and mission-specific radio and network configurations [1]. With many satellites persistently in view, they will compete for these limited resources, creating bottlenecks in capability. As more CubeSats are placed in orbit, the ground stations that communicate with them need to become more efficient, requiring optimization strategies to make best use of the ground station network [1], [2], [4], [5].

The rise in ground station pass density, or the number of satellites in view of any ground station at a given time, combined with the requirement for successful downlinking of data, creates a scenario in which satellites will need to compete for antenna priority. Ground station utilization will come at a premium, and, in the case of conflicting satellite passes, failed attempts at closing the communication link will waste valuable resources. In an equal use operating model, priority can be given to the satellite that has the best opportunity to transmit and receive data from the ground station.

In order to strategically plan before reaching ground station saturation, network operators can examine any existing CubeSat pass data and analyze the factors that define a successful pass. With CubeSats, uplinking is typically less constrained because ground-based amplifiers can use large amounts of power to overcome losses created by great distances and small receive antennas. However, given the size and power constraints on board, CubeSats must use substantially less transmit power, leading to a much more challenging downlink scenario on the ground [2], [3], [4], [6], [7]. Stations routinely have to overcome free space path loss of approximately 1,000 km, tumbling satellites, Doppler shift, and terrestrial sources of noise [2], [3], [4], [6]. This thesis creates a grading scale that can be used to prioritize which CubeSat should have antenna priority based on the expectation of actually downlinking data and when during the satellite's pass an operator should expect to capture that data. The remainder of Chapter I provides an example showing the effects of CubeSat proliferation and describes the sources of ground station pass data available for analysis.

A. CONFLICTING PASSES

As the number of satellites using a network grows, the capability of each ground station to effectively manage accesses becomes a major operational concern. The network must facilitate satellites to maintain frequent contact with the ground infrastructure for command uplink and data downlink. A Systems Took Kit (STK) model and associated access data, shown in Figure 1 and Table 1, demonstrate the consequence of a growing number of satellites on ground station access time. Each of the four STK scenarios simulates a single ground station located at NPS. The study utilizes Writt's Ground Station Saturation Monte Carlo code to model the space environment and generate satellites that grow in number for each successive simulation [8]. This simplified scenario uses a CubeSat-like, circular orbit with a semi-major axis of 7,020 km. The code randomizes the remainder of the orbital elements to include right ascension of the ascending node, argument of perigee, true anomaly, and inclination, which is constrained to a maximum of 70°. The simulated NPS ground station is monitored over a 48-hour period for the frequency and duration of *conflicting passes*, a condition where there is more than one satellite above 10° elevation angle at a time.

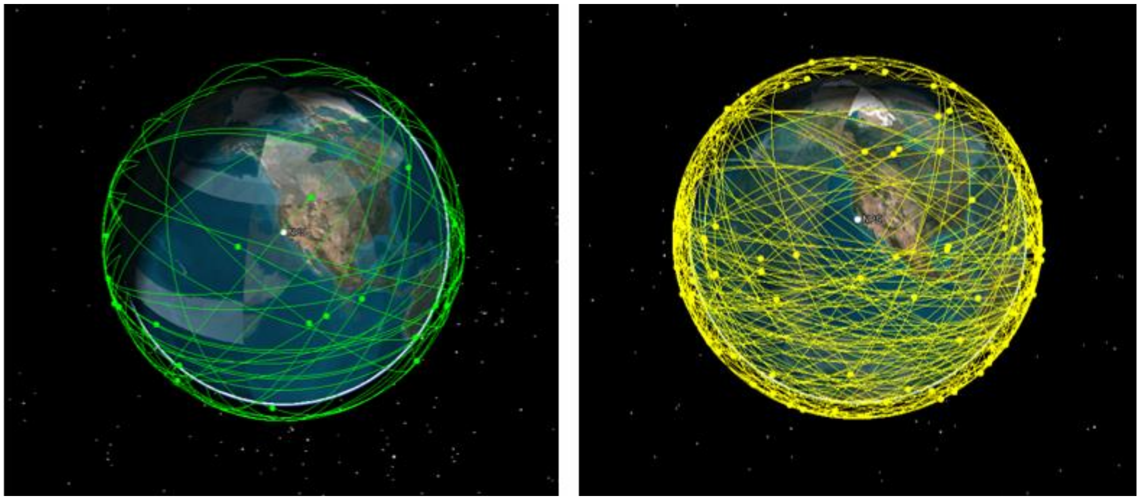


Figure 1. STK Models of 30 (left) and 120 (right) Satellites

Table 1. STK Model Conflicting Access Times

Satellites	Total Access Time	Multiple Access Time	Conflicting Access Time
30	67.4%	33.7%	50.0%
60	86.1%	64.3%	74.7%
90	97.9%	89.0%	90.8%
120	99.3%	95.1%	95.6%

Ground station saturation is assessed by comparing two metrics in each of the four scenarios. The total access time describes the percent of the time that the NPS ground station has line-of-sight with any one satellite in any given period of time whereas the multiple access time represents the duration that it has two or more satellites in view. By expressing these metrics as a percent of the total simulation time, and relating them to each other, the proportion of conflicting access is calculated. This value represents the time that the simulated NPS ground station is constrained to give antenna priority to one satellite at the expense of ignoring others.

As the number of satellites in the network grows from 30 to 120, the access time spent in conflict increases significantly. Operating with 30 satellites, the network maintains a total access duration for 67% of the simulation. However, half of the total access time is spent in conflict with simultaneous passes of at least one additional satellite. As the simulated network reaches 120 satellites, the NPS ground station is in near-constant contact with the constellation but spends over 95% of that time with multiple satellites overhead. In this particular scenario, at numerous times throughout the simulation, the ground station has up to 13 satellites within line-of-sight. The increase in conflicts limits the amount of data that can be transferred to the ground network as it becomes saturated with progressively more simultaneous accesses. Looking at the case where the number of satellites vastly outnumbers the number of ground stations, the antenna assets must, therefore, be used optimally to communicate with the satellites that have the greatest chance of downlinking data successfully. For a more complete analysis and discussion of ground station saturation and the factors that affect it, please see the thesis by Writt [8].

B. MOBILE CUBESAT COMMAND AND CONTROL (MC3) NETWORK

The MC3 Network Satellite Operations Center (SOC), located at NPS, Monterey, CA, has (as of May 2018) collected over a year and half of data on the Picosats Realizing Orbital Propagation Calibrations using Beacon Emitters (PropCube) Satellites, Flora and Merryweather [1]. Additionally, the SOC has collected nearly six months of pass data on the PropCube satellite, Fauna, placed in orbit more recently. This data, taken at numerous ground stations throughout the network, includes pass geometry and transmission decodes, a significant measure of data transfer described later in the thesis.

As shown in Figure 2, the available MC3 Network ground stations are located in Monterey, CA (NPS-PTSUR), Honolulu, HI (HSFL), Fairbanks, AK (UAF), Logan, UT (SDL), Albuquerque, NM (UNM), Melbourne, FL (MLB), and Wright-Patterson Air Force Base, OH (AFIT). In the coming months, plans for additional ground stations are underway for Annapolis, MD (USNA) and New London, CT (USCGA). This distributed infrastructure allows for numerous access opportunities for satellites in the MC3 Network. However, as the number of CubeSats desiring to use the network continues to grow, the ground stations may soon become saturated from the simultaneous overhead passes.

Figure 2. MC3 Active (Green) and Planned (Yellow) Ground Stations

The reason for decreased data flow efficiency is that there may be more than one satellite overhead at a time, as shown in Table 1. To begin to improve the data flow efficiency, network operators can look to the MC3 pass tracking algorithm. In the event of two satellites with overlapping access times, the ground station simply defaults to tracking the first satellite that reaches 10° elevation angle. The antenna will subsequently track that satellite for the entire duration of its pass. Once the initial satellite pass is complete, the antenna will not transition to tracking the second satellite. Operator inputs can change the priority satellite, but the network remains constrained by ignoring the other as the conflicting pass.

In addition to the shortcomings of the pass tracking algorithm, data flow in the MC3 Network is also constrained by the characteristic size of its CubeSats. The satellites, due to low-gain antennas and small power budgets, operate with reduced link margins as compared to their larger counterparts [2], [3], [4], [6], [7]. The signal transmitted by CubeSats, therefore, is highly susceptible to pointing errors, atmospheric attenuation, and noise environments near the ground station antenna [1], [2], [5], [9].

All other considerations equal, the satellite with the greater likelihood of data transfer should be prioritized over other satellites. For real-world implementation, analyzing and grading individual portions of a pass, such as ingress towards and egress away from a ground station, is also necessary. Pass segment grading provides a metric to determine the optimum time to switch during a pass between two conflicting satellites in order to deliver the best overall communication. This information, when used by a robust optimization algorithm, should maximize performance of the antennas.

Minelli's dissertation [10] on the optimization of satellite passes includes a thorough treatment and discussion of this topic, referred to as the "many satellites, few ground stations" problem. It is expected that the value function determined in this thesis will be useful for an actual implementation of Minelli's optimization code to any PropCube conflicts for a given ground station.

C. THESIS OVERVIEW

This thesis, consisting of five chapters, determines a metric for measuring the quality of a satellite pass as a function of several variables and establishes criterion to predict the likelihood of successful data transfer for upcoming passes in order to manage ground station antenna scheduling. This first chapter was an introduction to the "many satellites, few ground stations" problem soon to be faced by networks, such as MC3, due to the projected growth in the number of CubeSats.

Chapter II provides a general overview of the CubeSat orbital characteristics, exploring the relationship between the initial azimuth angle and maximum elevation of a given pass. By relating these metrics, the study verifies that maximum elevation angle, and therefore minimum range, is dependent solely on the initial azimuth of the ascending or descending pass. Chapter II also provides an in-depth description of the MC3 Network data collection. This chapter outlines the methods of collection, ground station constraints, initial form of recorded files, and basic import of the data into MATLAB to be used for later analysis.

The third chapter discusses the analysis of CubeSat passes using MATLAB and the PropCube data set. First, a correlation is established between the azimuth and elevation of the satellite and the number of successful data downlinks. Preliminary data analysis suggests that the main driver of a successful pass is the maximum elevation angle of the satellite overhead the ground station. However, since all high elevation passes do not result in successful data downlink, other contributing factors, such as ground station location and time of year, must be explored. After this view on the macro level, considering passes as a single event, a more granular analysis will be used to investigate the factors that contribute to data transfer success during particular segments of an individual pass.

Following data analysis using MATLAB, Chapter IV establishes a grading criteria weighted towards the more influential factors of a successful pass. It uses this pass quality metric to predict the successful downlink likelihood of future Flora, Merryweather, and Fauna passes throughout the MC3 Network. By modeling the pass, predicting its chance of success, and comparing to actual data received, the predictive model and grading metric can be validated and further refined to establish a more accurate representation of pass priority handling.

D. THESIS GOALS

Applying the tools developed in this thesis to the PropCube data results in a pass value function that characterizes the value of future ground station-satellite passes as a function of initial azimuth and elapsed time. The pass value function can then be used by an optimizing program, such as that being developed in the NPS Small Satellite Laboratory [10], to determine how to best optimize the data retrieved from even a small constellation of satellites. In addition, this quality metric can be applied to any similarly configured CubeSat ground station in order to efficiently schedule contacts and maximize use of communication assets. As the number of CubeSats continues to grow on these networks, commercial, civil, and military system operators will be able to predict the quality of upcoming passes and prioritize satellites with a high likelihood of link success.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND AND MC3 CUBESAT DATA COLLECTION

Chapter I outlined the reduced data transfer associated with increasing number of CubeSats and the desire for more efficient pass scheduling. As CubeSat constellations continue to grow, resulting in an increased number of conflicting passes, operators will likely give priority to the satellite with the best opportunity to communicate with the ground station infrastructure. Several factors, such as ground station location and orbital parameters, determine the pass quality, and therefore, affect prioritization. This chapter discusses each of those factors and the overall MC3 Network data set of PropCube passes, including constraints in collection.

A. ORBITAL CHARACTERISTICS OVERVIEW

Prior to conducting data analysis, it is critical to understand the physical nature, geometry, and characteristics of a satellite pass over a ground station. The pass initial azimuth (AZo) is defined as the ground station-relative angle at which the PropCube begins its access whereas the final azimuth (AZf) is the angle at which it ends its access. Part of this thesis validates the assumption that for a given pass initial azimuth, there is a specific, associated maximum elevation (Max El) angle. As a result, given the CubeSat orbital elements, the trajectory of a pass from the recorded acquisition of signal (AOS) to loss of signal (LOS) can be characterized by referencing the pass initial azimuth. By defining ground station-satellite tracks using a single variable, the passes can be more effectively grouped to recognize trends in data transfer rates and model future passes. Ultimately, the CubeSat initial azimuth is one of the most important factors correlating to the pass value function explored later in the thesis.

1. Pass Geometry

If a satellite's orbital inclination is less than the ground station latitude, there will be no overhead passes of that ground station. However, if the orbital inclination is greater than the ground station latitude, the satellite can travel overhead in two unique ways. If the pass occurs as the satellite's Earth-relative latitude is increasing, it is defined as an *ascending pass*. Conversely, if the pass occurs as the satellite's relative latitude is decreasing, it is defined as a *descending pass*, as shown in Figures 3 and 4.

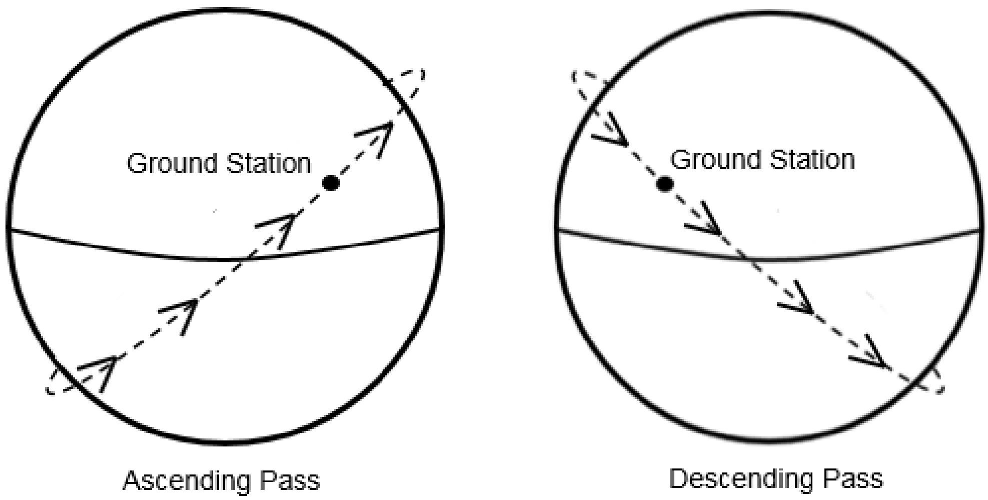


Figure 3. Satellite Ascending (left) and Descending (right) Passes

Figure 4. STK Model of Ascending (left) and Descending (right) Passes

In the polar depiction of a single pass, the outer edge represents 0° elevation angle, which corresponds to the ground station horizon. The ground station itself is positioned in the center of the plot co-located with a 90° elevation angle. The satellite ingress is the beginning part of a pass starting at AOS, from an elevation of 10° up to a possible elevation of 90°, and egress is the last part of the pass from elevation of 90° back down to 10° at the LOS. The closest point of approach (CPA) is the location between ingress and egress where minimum range occurs between satellite and ground station. As shown in the example descending pass in Figure 5, the AOS begins at an elevation of 10° and an AZo of 335°. At the LOS, the elevation is 10° and the final azimuth is 146°.

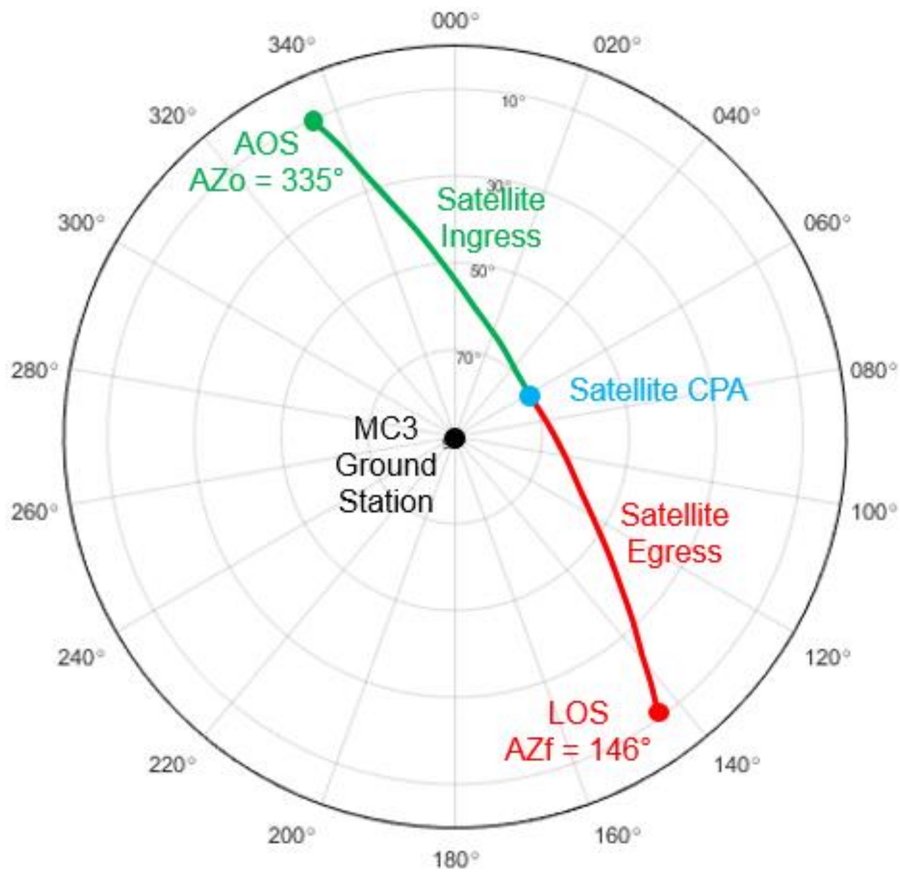


Figure 5. MC3 CubeSat Pass Geometry

As explored later in this chapter, the MC3 Network software is constrained to only enable passes that reach at least 40° elevation angle. As such, the majority of recorded MC3 data set passes occur within the 40° elevation threshold range, as shown in Figure 6.

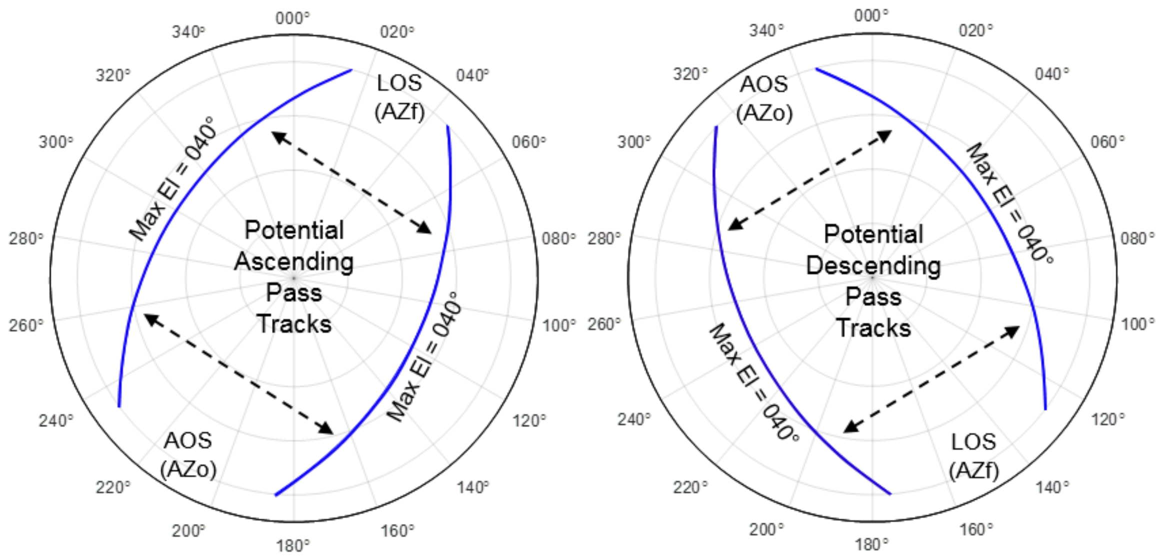


Figure 6. Ascending (left) and Descending (right) 40° Elevation Angle Constraint

2. Defining a Pass from Initial Azimuth

In the case of a nadir CubeSat pass, or one that travels directly overhead the ground station, this thesis mathematically verifies the presence of a unique initial azimuth by using MC3 pass data in a simple launch azimuth calculation [11]. Since the analysis occurs along a small segment of the orbit, the factors of Earth rotation and the change in crossing angle as the satellite approaches its highest latitude are small. As such, when viewed on a polar plot relating azimuth and elevation angle, the nadir passes are virtually straight line tracks that start at the initial azimuth, pass directly over the ground station, and terminate at the reciprocal final azimuth. Figure 7 shows two such passes, one ascending and one descending, for the PropCube, Flora, overhead the PTSUR ground station.

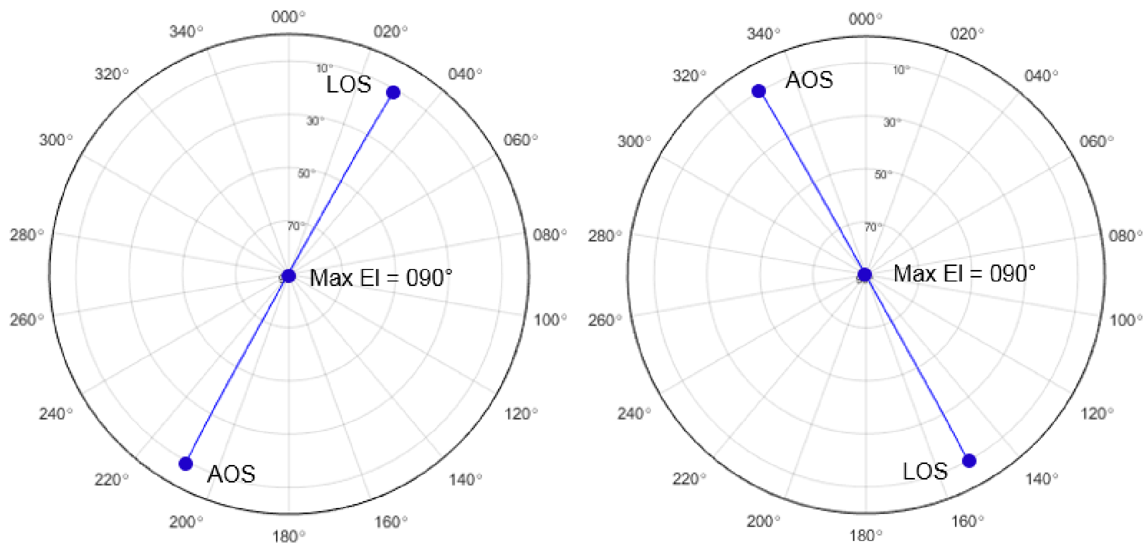


Figure 7. Ascending (left) and Descending (right) Nadir Passes

Given a desired satellite inclination (i) and launch site latitude (ϕ), Curtis's method calculates the launch azimuth (A) of a body being placed in orbit [11]. Similarly, given a satellite inclination and ground station latitude, this thesis determines the initial azimuth of a nadir pass, which by definition must have a 90° maximum elevation angle. In the case of a PTSUR-Flora access, the 64.8° average satellite inclination and 36.6° ground station latitude are consistent among all passes. Therefore, the entry and exit azimuths must also remain fixed and are inextricably linked to the maximum elevation angle of the pass.

$$cos(i) = cos(\phi)\, sin\,(A)$$

$$cos(64.8°) = cos(36.6°)\, sin\,(A)$$

$$A = 32°$$

$$Ascending\ Pass\ AZo = 180° + A \qquad Descending\ Pass\ AZo = 360° - A$$

$$Ascending\ Pass\ AZo = 212° \qquad Descending\ Pass\ AZo = 328°$$

As shown in Figure 8, which depicts historical MC3 pass initial azimuths and their associated maximum elevation angles, the ascending nadir pass has an initial azimuth of approximately 212° whereas the descending pass has an initial azimuth of 328°. Using

Flora's inclination and the PTSUR ground station latitude, the launch azimuth calculation verifies the expected initial azimuth for both the ascending and descending pass.

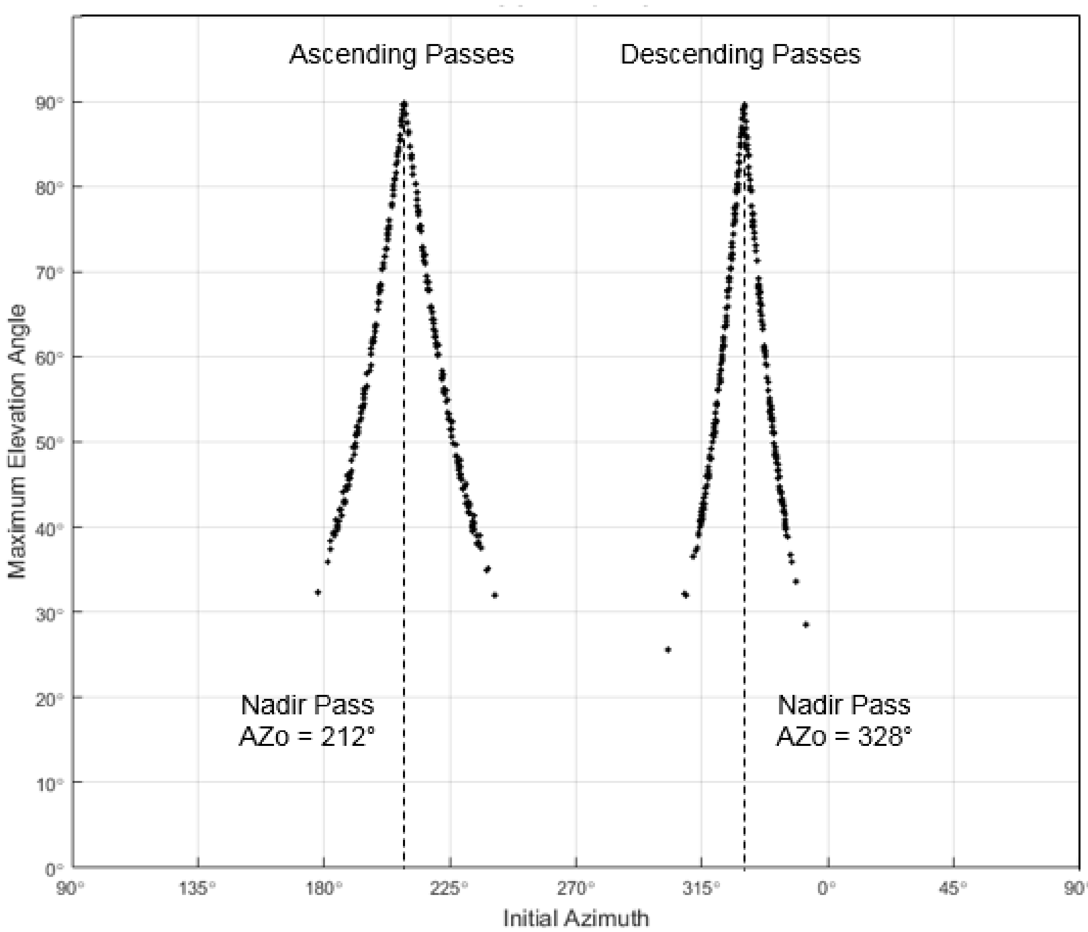


Figure 8. PTSUR-Flora AZo vs Max El

Furthermore, Figure 8 depicts the correlation between the remaining non-nadir initial azimuths and their maximum elevation counterparts. Utilizing MATLAB to analyze the MC3 data set further proves the consistent relationship between initial azimuth and maximum elevation angle. The two distinct peaks represent the grouping of ascending and descending passes with respect to a MC3 ground station-PropCube satellite pair. The track pass data has been collected in a polar format, using azimuth and elevation. Consequently, the northern azimuths represent a smaller physical distance than the southern azimuths due to the circular ground station projection onto a spherical Earth. As one travels to higher latitudes, the distance between the lines of longitude decreases, until ultimately intersecting at the pole. This phenomenon can be seen in Figure 8 where the descending initial azimuth pass data, which begins to the north of the ground station, is narrower than that of the ascending group.

3. Circular and Elliptical Orbit Pass Geometry

This thesis has theorized that the best opportunity for data downlink occurs at or near the maximum elevation within a pass as this location correlates to the satellite minimum range. However, due to orbital eccentricity, these two points may not always be coincident. In order to determine if the difference has a significant impact on pass quality analysis, this section analyzes the relationship between maximum elevation and minimum range.

Using STK, this study models a circular orbit with parameters similar to that of the MC3 CubeSats and generates an azimuth, elevation, and range (AER) report for a single pass over the PTSUR ground station, as shown in Figure 9 and Table 2. The elapsed time between the location of maximum elevation and minimum range is approximately 0.2 seconds. Therefore, in the case of a circular orbit, the difference is negligible, and the two positions are effectively coincident.

Figure 9. Circular Orbit Pass

Table 2. Circular Orbit Max Elevation and Min Range

	Time	Range (km)
Max Elevation	18:43:29.94	656.3113
Min Range	18:43:30.13	656.3099

Since this thesis analyzes the pass quality of satellites with non-circular orbits, the study subsequently models the MC3 PropCube, Flora, using historical two line element set (TLE) for a simulated pass over the PTSUR ground station. By analyzing the difference between the maximum elevation and minimum range locations using a realistic eccentricity, the study aims to determine if this disparity is significant enough to affect the results of the pass analysis value function, as shown in Figure 10 and Table 3.

Figure 10. Elliptical Orbit Pass (True Anomaly 90°)

Table 3. Elliptical Orbit Max Elevation and Min Range (True Anomaly 90°)

	Time	Range (km)
Max Elevation	19:58:34.73	647.8688
Min Range	19:58:33.14	647.7735

At the time of this pass, Flora is centrally located between its perigee and apogee as indicated by its 90° true anomaly (θ). As its track passes over the PTSUR ground station, the maximum elevation occurs 1.6 seconds after the minimum range. During this time, Flora climbs nearly 100 meters as it travels from the lower altitude portion of its orbit towards its maximum altitude at apogee.

The Flora STK model illustrates that maximum elevation and minimum range are not necessarily coincident in elliptical orbits. In most cases, the satellite is climbing or descending in altitude with respect to the ground station throughout its passes. However, if a pass occurs in close proximity to either apogee or perigee, there is no climb or descent, but rather the satellite has reached its maximum or minimum altitude. Therefore, passes with true anomaly values near 0° at perigee or 180° at apogee will not see the same effect as compared to other locations in the elliptical orbit, as shown in Figure 11 and Table 4.

Figure 11. Elliptical Orbit Pass (True Anomaly 0°)

Table 4. Elliptical Orbit Max Elevation and Min Range (True Anomaly 0°)

	Time	Range (km)
Max Elevation	00:57:38.32	529.8103
Min Range	00:57:38.11	529.8081

The STK model exhibits that there may be a slight difference in time and location of the maximum elevation and minimum range. However, due to their minor eccentricity and the pass value analysis occurring over a small segment of the overall orbit, the altitude difference throughout the CubeSat passes is negligible with respect to link margin calculations. Therefore, these changes are not considered during determination of the pass quality and value functions.

4. Repeating Pass Geometries

The semi-major axis and, therefore, the orbital period of each PropCube have remained relatively constant throughout their lifetimes. For example, from the first available TLE in October 2015 to the most current (as of May 2018), Flora's semi-major axis has decreased 4.8 km, equating to only a six second reduction in orbital period [11]. Due to the constant rate of Earth's rotation, combined with the consistent orbital period of the PropCubes, the pass geometries and initial azimuths for a particular ground station-satellite pair repeat at regular intervals. Table 5 shows three pass groups, each group consisting of four consecutive passes of the PropCube, Flora, over the PTSUR ground station.

Table 5. PTSUR-Flora Repeating Passes

Pass Group	Pass Date	Type	AZo	Max El	AZf
1	8 Feb 2017	Descending	331.8°	82.3°	148.4°
	8 Feb 2017	Ascending	191.4°	50.8°	41.0°
	9 Feb 2017	Ascending	222.2°	57.8°	21.7°
	10 Feb 2017	Descending	339.4°	53.6°	136.0°
2	5 Mar 2017	Descending	331.5°	83.7°	148.9°
	6 Mar 2017	Ascending	191.6°	50.8°	41.0°
	7 Mar 2017	Ascending	222.9°	56.2°	21.9°
	7 Mar 2017	Descending	339.1°	54.7°	136.7°
3	31 Mar 2017	Descending	331.5°	83.7°	148.9°
	1 Apr 2017	Ascending	191.3°	49.9°	41.3°
	2 Apr 2017	Ascending	223.1°	55.6°	21.0°
	2 Apr 2017	Descending	339.1°	55.0°	136.8°

The first pass group is defined by initial azimuths of 331.8, 191.4, 222.2, and 339.4°. The next occurrence in the PTSUR-Flora data set that has initial azimuths within 1° of those passes is approximately 25 days later, notably in the same order as group one. Similarly, the four consecutive initial azimuths repeat, again 25 days later, in the third group of passes. This regular interval of like-AZo passes not only shows the repetitive nature of single accesses, but also exhibits a pattern of predictable consecutive passes over the lifetime of the orbit, critical to the analysis of changing pass duration.

5. Pass Duration

As explored in Section 3, even a small eccentricity can have an effect on the geometry of a ground station-satellite pass. Due to J_2 perturbation effects, the argument of perigee (ω) will change throughout the orbital lifetime of the MC3 PropCubes [11]. For a single pass, this thesis ignores the disparity between maximum elevation and minimum range caused by a changing argument of perigee.

However, the rotation of perigee has a significant effect on the CPA and, therefore, the overall link budget and ability to communicate with ground stations throughout the lifetime of the orbit. As the perigee location rotates nearer to the latitude of the target ground station, the average pass duration decreases due to a reduced ability to maintain line-of-sight with the satellite. Additionally, the average altitude of the pass overhead that ground station will decrease, resulting in an improved signal-to-noise ratio during these passes, as further explored in Chapter III.

This effect is first recognized during pass duration analysis of the PTSUR-Flora data set. When comparing initial azimuth and pass duration, the expected output is two sets of grouped data, representing the ascending and descending orbital passes. It was anticipated that each group would reach a maximum duration at the nadir pass initial azimuth due to the longest line-of-sight opportunity. However, the assessment yields unexpected results as the durations vary throughout the data set. Not only do the durations differ for like-AZo passes, but there is significant disparity upon comparing the ascending and descending groups, as some descending passes reach a maximum duration of nearly

650 seconds whereas the maximum ascending pass duration is approximately 550 seconds, as shown in Figure 12.

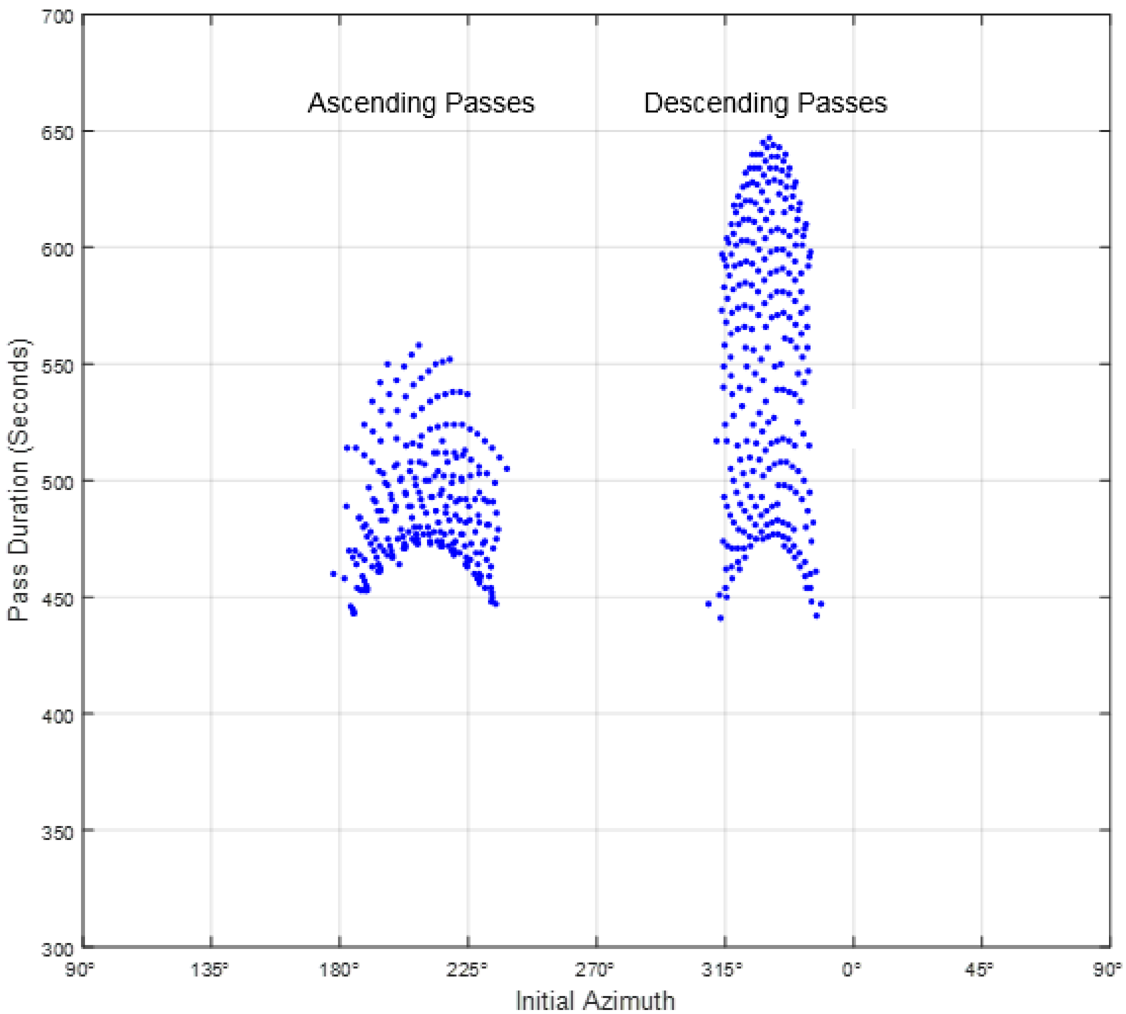


Figure 12. PTSUR-Flora Pass Duration

Both the ascending and descending plots consist of multiple bands of like-duration passes. These patterns represent the consecutive pass groups described in Section 4. As each pass's initial azimuth is repeated, however, the bands do not remain at the same relative pass duration. Instead, when the next like-AZo pass group occurs approximately 25 days later, the average pass duration changes, shifting the entire band of consecutive passes up or down accordingly, as shown in Figure 13.

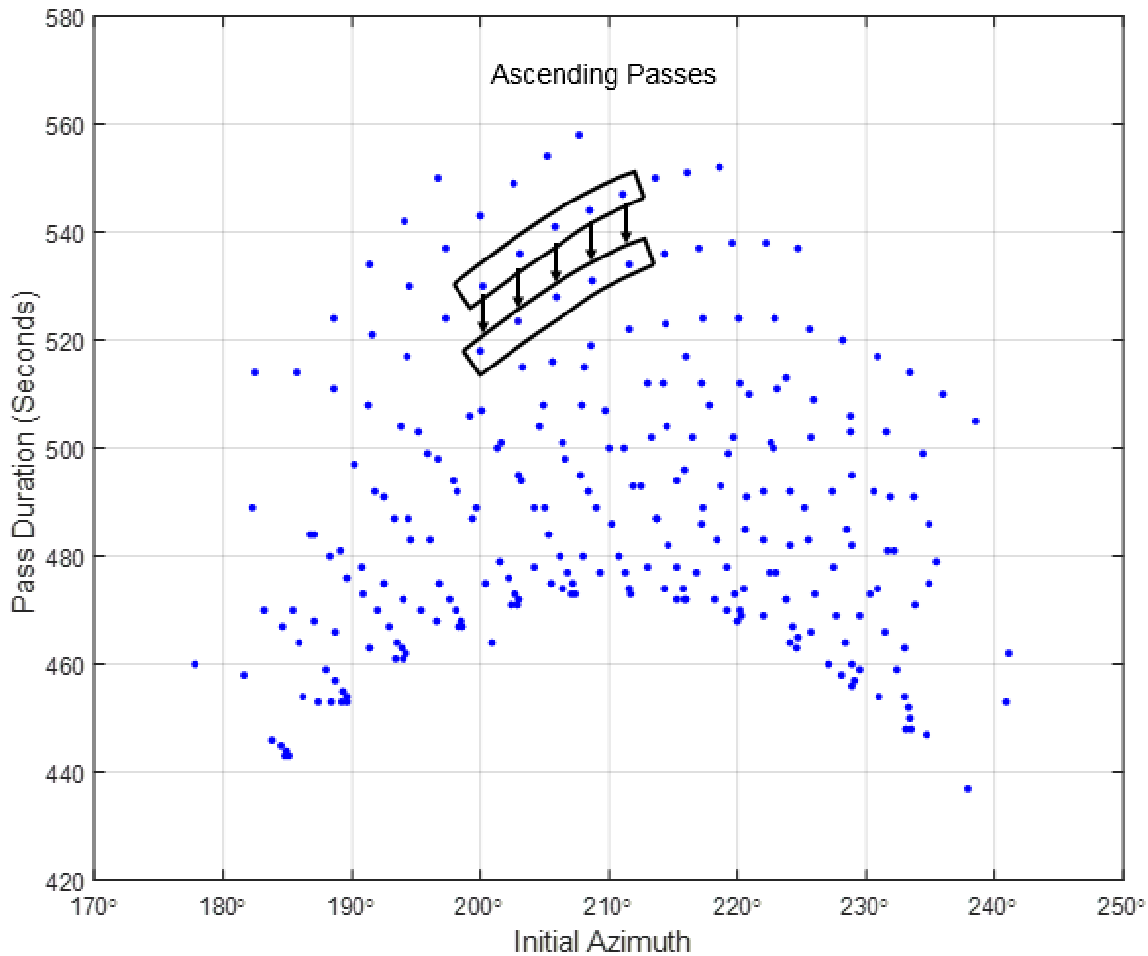


Figure 13. PTSUR-Flora Ascending Pass Duration Bands

Based on this observation, these data points were not only grouped by ascending and descending passes, but also by the date on which they occurred. Grouping the data points by month, in order to visualize the temporal variations in pass duration, Figure 14 depicts the pass duration change throughout a nine month period. As the argument of perigee rotates throughout the year, the shorter duration passes occur when perigee is closest to the pass latitude of the ground station. On the other hand, when apogee is located at the pass latitude, the pass duration is at a maximum. The consecutive pass groups, therefore, maintain a similar initial azimuth pattern, but the duration of each group can change as much as 200 seconds based on the relative position of perigee to the ground station latitude.

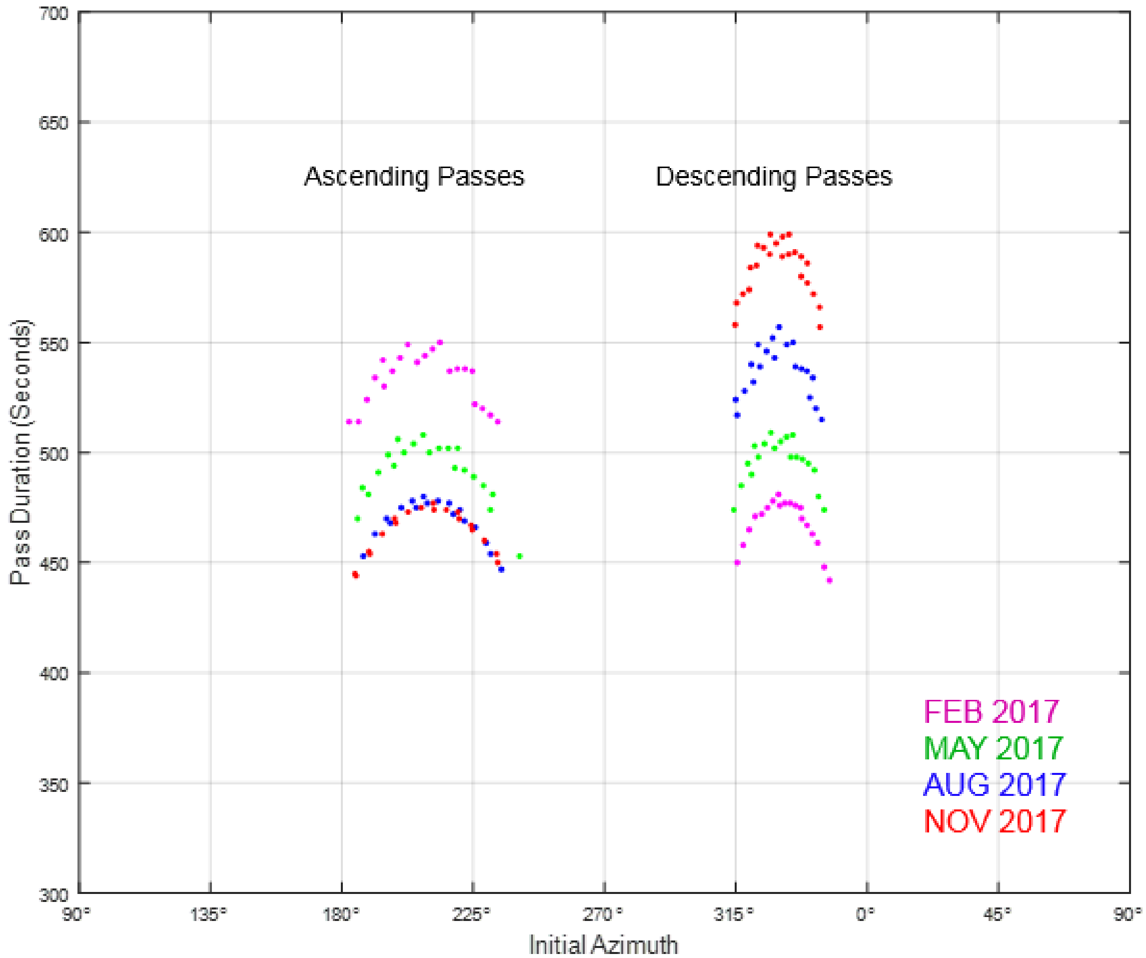


Figure 14. PTSUR-Flora J_2 Perigee Rotation Effect on Pass Duration

Using historical Flora TLE, the orbital parameters were modeled in STK to further illustrate this relationship. The custom report in Figure 15 depicts Flora's argument of perigee throughout its lifetime in orbit.

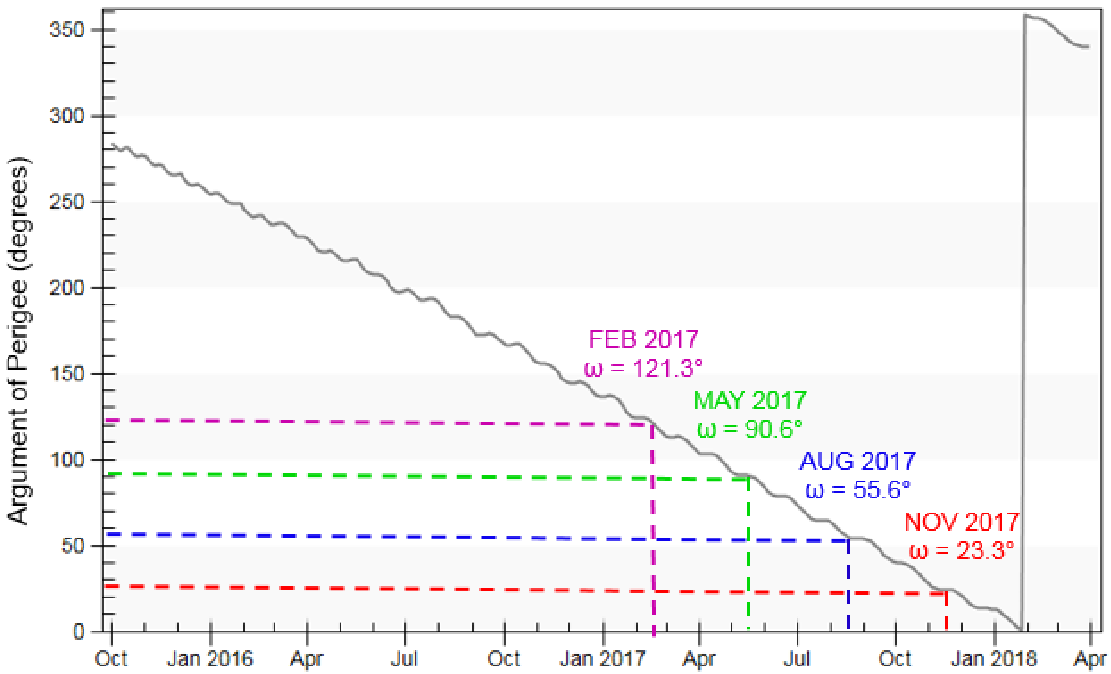


Figure 15. Flora Argument of Perigee Throughout CubeSat Lifetime

Correlating the data points from Figures 14 and 15, the pass durations in May 2017 are consistent with the argument of perigee value. Figure 14 depicts that the total pass time for both the ascending and descending passes is approximately 500 seconds. The significance of this equilibrium is that during this time period, based only on signal strength, the value of ascending and descending passes is the same. The pass durations can only be equal when perigee is equidistant from the ground station latitudes on both the ascending and descending side of the orbit. Therefore, the only two locations where the durations are the same length are when perigee is at its peak latitude in either the Northern or Southern Hemisphere. These two locations are associated with argument of perigee values of 90 and 270°, respectively. According to Figure 15, the argument of perigee in May 2017 is approximately 90°, consistent with the peak latitude in the Northern Hemisphere.

Although a longer pass duration provides additional time to communicate with a CubeSat, these passes are extended due to a higher average altitude. Consequently, this change reduces the downlink signal strength and link margin, which may result in an overall decrease in performance. Chapter III explores the implications of this increased distance between the ground station and satellite that results in greater free space signal loss.

B. THE MC3 DATA SET

In October 2015, the MC3 SOC began capturing data on the two PropCubes, Flora and Merryweather, from ground stations throughout the network. Following its launch in December 2017, a third PropCube, Fauna, also began to contribute its pass information to the MC3 data set. Throughout their orbits, as the satellites pass over nodes in the MC3 Network, the SOC records the track geometry along with the successful communication attempts via transmission preambles and decodes.

The MC3 PropCubes utilize the AX.25 standard communication protocol to transmit data to the network ground stations [12]. As shown in Figure 16, each transmission begins with a preamble that consists of a series of alternating ones and zeros to synchronize the ground station receiver with the satellite in preparation for data downlink. The preamble concludes with a 0x7E flag that signifies the start of data transmission [12]. Prior to another 0x7E flag that indicates the end of the message, a cyclic redundancy check (CRC) value, calculated based on the contents of the transmission itself, is appended to the data [12].

Preamble	Flag	Data	CRC	Flag	Extra
0xAA	0x7E	0xXX	0xXX	0x7E	0xFF
400 Bytes	1 Byte	X Bytes	2 Bytes	1 Byte	5 Bytes

Figure 16. AX.25 Standard Packet Structure. Adapted from [12].

Upon receiving the transmission, the ground station checks the validity of the message by calculating its own CRC based on the contents of the received data [12]. In order to pass the CRC and record the transmission as a valid decode, the calculated ground station and received CRC values must match, providing high statistical probability that the sent and received messages are identical [12]. In the event that the CRCs do not match, the transmission is recorded as a successful preamble, but a failed decode [12]. As shown in

Figure 17, a possible source of CRC mismatch is sporadic radio-frequency (RF) noise received by the ground station antenna, which has potential to disturb the incoming signal and increase the link bit error rate.

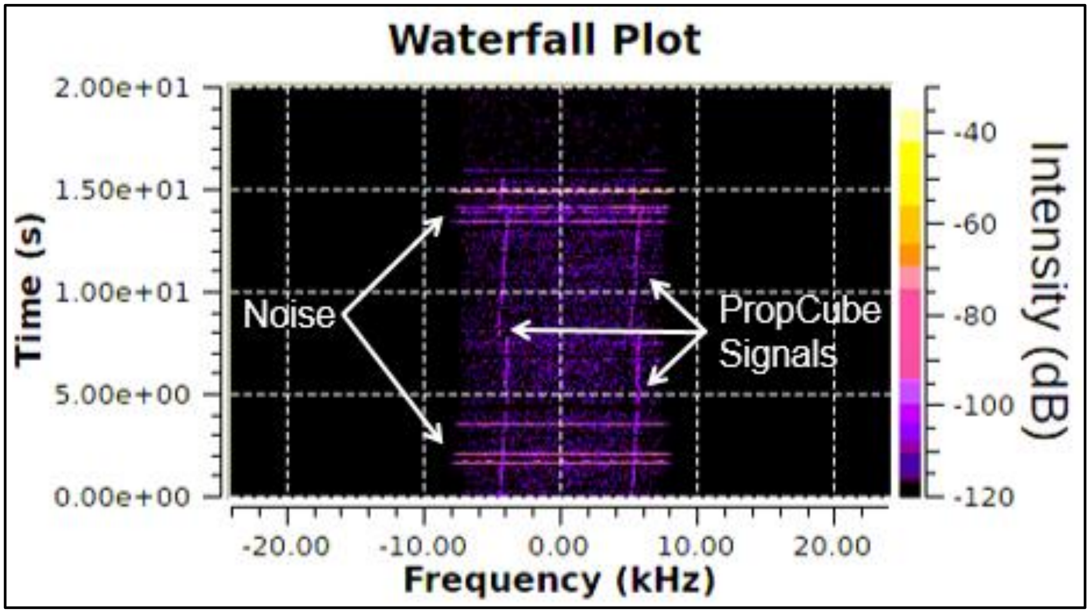


Figure 17. MC3 SOC Waterfall Plot

As an MC3 PropCube executes a ground station pass, the SOC software displays visual verification of received downlink signals via a waterfall plot. Centered on the approximate downlink frequency of 914 MHz, the plot displays the previous 20 seconds of received energy by the ground station antenna. MC3 operators can observe the energy of incoming transmissions and then retrieve the pass performance measured via successful preambles and decodes. The thesis uses these performance metrics to analyze the MC3 data set in three ways.

1. Whole Pass Analysis

The first analysis of the MC3 data set measures the overall success rate of the specific ground station-satellite pairs by evaluating each pass as a single datum, referred to as *whole pass analysis*. This approach investigates whether an entire pass succeeded or failed in order to provide a holistic view in which quality of the pass is measured by the completion of at least one preamble or transmission decode.

As shown in Figure 18, the data set, collected by the SOC for this macro view of MC3 pass quality, includes the ground station name, satellite identifier, AOS time, LOS time, AZo, AZf, Max El, pass duration, and the number of successful communication preambles and decodes.

ID	AOS [US/Pacific]	LOS [US/Pacific]	Start AZ	Stop AZ	Max EL	Pass Time [secs]	Ground Station	Satellite	Catalog Number	Dec	Pre
70630	2018-04-06 02:22:28 PDT	2018-04-06 02:33:15 PDT	330	151	88	647	NPS	13FLORA	90736	0	0
70637	2018-04-06 02:22:28 PDT	2018-04-06 02:33:15 PDT	330	151	88	647	PTSUR	13FLORA	90736	0	12
70626	2018-04-06 03:05:57 PDT	2018-04-06 03:12:55 PDT	330	129	55	418	HSFL	FAUNA	43052	4	17
70624	2018-04-06 04:42:10 PDT	2018-04-06 04:52:22 PDT	352	134	42	612	HSFL	10MERRYW	90738	0	7
70682	2018-04-06 05:43:17 PDT	2018-04-06 05:53:52 PDT	326	170	58	635	HSFL	13FLORA	90736	8	58
70672	2018-04-06 09:16:23 PDT	2018-04-06 09:26:25 PDT	331	150	88	602	NZL	13FLORA	90736	0	0

Figure 18. MC3 SOC Network Management Page

Using these parameters, the likelihood of successful data downlink is assessed based on past ground station-satellite performance. As outlined in Appendix A, the MC3 Network Management Page and associated structured query language (SQL) is utilized to extract the recorded data into a comma-separated values (CSV) format for follow-on analysis using MATLAB.

2. Segmented Pass Analysis

In addition to examining each CubeSat pass a whole, where it was categorized as either a success or failure, it is equally important to recognize that communication performance may differ throughout distinct pass segments, as a function of azimuth and elevation. While performing a pass, the satellites are intrinsically more likely to have successful data downlink when closer to the ground station due to decreased space loss and, therefore, increased signal strength. Optimum performance during a pass should occur near the CPA. Furthermore, each MC3 antenna is affected by local obstacles in the form of terrain, structures, and dense RF environments that can interfere with communication.

Defined as *segmented pass analysis*, this method utilizes a second data set to analyze the position of successful data downlink within individual passes in order to predict the azimuth-elevation segments that have the greatest likelihood of success. For each CubeSat pass, the MC3 SOC collects detailed information concerning the satellite's path geometry and data transmission times. The track file for each pass records, in increments of one-second, the azimuth, elevation, and range of the satellite with respect to the ground station, as shown in the example in Figure 19.

Date	Time	Az	El	Range	GS
2 Mar 2017	18:46:12.331930996	27.37	10.00	1770.98	PTSUR
2 Mar 2017	18:46:13.000000000	27.53	10.02	1769.66	PTSUR
2 Mar 2017	18:46:14.000000000	27.76	10.05	1767.71	PTSUR
2 Mar 2017	18:46:15.000000000	28.00	10.08	1765.79	PTSUR
2 Mar 2017	18:46:16.000000000	28.23	10.11	1763.89	PTSUR
2 Mar 2017	18:46:17.000000000	28.47	10.13	1762.01	PTSUR
2 Mar 2017	18:46:18.000000000	28.71	10.16	1760.16	PTSUR

Figure 19. Track File Data Format

In addition to the track information, the MC3 SOC creates a log file associated with each pass when the target ground station detects at least part of a successful data transmission. The communication preambles and decodes, recorded in the Raw and "keep it simple, stupid" (KISS) communication protocol logs, document the data received from the CubeSat along with an associated event timestamp, as seen in the example in Figure 20.

```
PTSUR 03/08/17 19:09:48.281 UTC pkt # = 5 of 7 KISS packet len = 201 bytes
201
C0109A8666606062008C989EA482620103CC16A76E7351FF4AEC394CBAA9323E0075821ACE...
PTSUR 03/08/17 19:09:54.335 UTC pkt # = 2 of 7 KISS packet len = 300 bytes
300
C0109A8666606062008C989EA482620103CC11A76E7351FF4AEC394CBAA9CBDE20C53A3C49...
PTSUR 03/08/17 19:09:54.501 UTC pkt # = 3 of 7 KISS packet len = 202 bytes
202
C0109A8666606062008C989EA482620103CC10A76E7351FF4AEC394CBAA93D922B72151EDB...
```

Figure 20. Log File Data Format

By correlating the track file position data with the instances of ground station-satellite communication from the log files, these combined data sets provide a means to determine the location of the satellite with respect to the ground station as significant events occur. This information can then be used to evaluate the specific segments of a pass that result in the best link performance for a ground station-satellite pair.

3. Supplemental CubeSat Data

Although the network track files provide the relative location of the satellite with respect to the ground station, the overall orbital parameters and pass geometry play a vital role in the probability of efficient communication. Therefore, the CubeSat TLE is required to supplement the data recorded by the MC3 SOC, as shown in the example in Figure 21. The historical TLE is extracted using the procedure outlined in Appendix B.

```
1 90736U          15288.40054719 +.00001463 +00000-0 +17624-3 0 0009
2 90736 064.7818 270.9147 0214888 279.5530 231.4662 14.7291480600084
1 90737U          15288.52585664 +.00001233 +00000-0 +14765-3 0 0009
2 90737 064.7850 270.5091 0217394 279.9270 192.5505 14.7360537900102
1 90738U          15288.45583022 +.00001718 +00000-0 +19874-3 0 0010
2 90738 064.7850 270.7209 0216853 279.7874 180.8416 14.7359200200085
1 90739U          15288.52685512 +.00003424 +00000-0 +37737-3 0 0007
2 90739 064.7844 270.5107 0216902 279.9037 196.9994 14.7358351100086
```

Figure 21. Space-Track.org TLE (Flora 90736, Merryweather 90738)

C. DATA COLLECTION LIMITATIONS AND CONSTRAINTS

Ideally, the MC3 Network would communicate with its satellites any time they are above the horizon and within line-of-sight. However, for many ground station passes, the capability to transmit and receive data is limited by factors such as physical obstacles, atmospheric attenuation, competing RF transmitters, and free space loss, all of which affect the communication link margin. As such, the MC3 Network software is configured with constraints intended to deliver the best opportunities to collect data from the satellites. Simultaneously, these constraints seek to protect the antenna assets from overuse and decrease unsuccessful, extraneous passes in the MC3 SOC data set.

As a CubeSat travels in its orbit, a ground station achieves theoretical line-of-sight any time the elevation angle to the satellite is greater than 0°. In a perfect world, at this earliest line-of-sight opportunity, the MC3 Network would begin to track and communicate with the asset. However, a Federal Communications Commission (FCC) operating rule dictates that "earth station antennas must not transmit at elevation angles less than five degrees" [13]. As an additional measure, in response to the aforementioned physical limitations, the network is constrained to only transmit and record track data when the satellite reaches a minimum of 10° above the horizon. Therefore, all track data used for this thesis begins and ends at this threshold value, annotated by the AOS and LOS times, respectively. These bounds ensure the target satellite reaches elevation angles at which obstructions are unlikely to have a substantial impact, although for the PropCube case, the lowest elevation to ever detect a preamble is 10.01° and the lowest elevation for a decode is 10.36°. Historically, however, the preponderance of successful communication occurs at higher elevation angles, with most preambles and decodes above the 40° elevation threshold, as shown in Table 6.

Table 6. Historical Probability of Preamble/Decode Below Threshold Elevation Angles

Elevation Threshold	Preambles	Decodes
10°	0.0 %	0.0 %
20°	6.5 %	0.9 %
30°	21.9 %	6.0%
40°	42.9 %	20.2 %
50°	65.4 %	46.4 %
60°	81.8 %	71.5 %
70°	92.7 %	88.6 %
80°	98.2 %	97.2 %

Furthermore, the software only schedules passes with a certain minimum elevation angle to reduce the number of passes that have little chance of communication. Data downlink is less likely at lower elevation angles due to physical obstacles and RF propagation losses associated with increased range. Additionally, while the PropCubes are near the horizon, local terrestrial emitters may interfere with the 914 MHz downlink signal. Therefore, as verified in Le Gaux's link budget calculations [14] and Table 6, in order to best utilize antenna assets with MC3's relatively small downlink power margin, the majority of the data collection has been constrained to only enable passes that reach at least 40° elevation angle. To illustrate the significance of increased elevation angle on range, Figure 22 depicts a nadir pass that reaches a maximum elevation angle of 90°, for a satellite at an altitude of 775 km in low earth orbit (LEO).

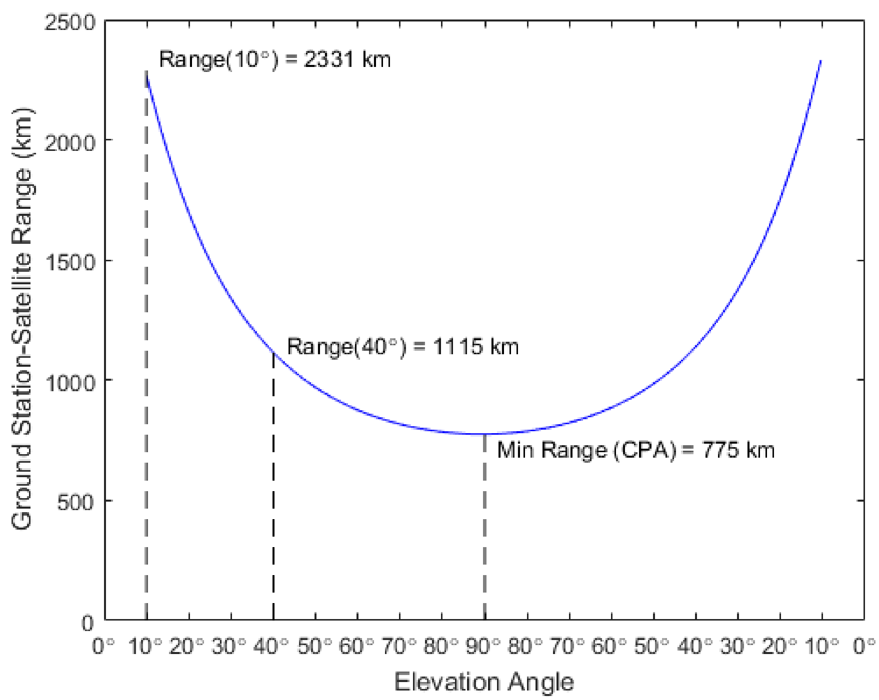


Figure 22. Pass Track Elevation Angle vs Range

Due to the relationship between elevation angle and range, once the satellite reaches the 40° threshold, it has significantly reduced the distance to the ground station, facilitating improved opportunity to communicate. In the case of the nadir pass in Figure 22, the transition from 10 to 40° elevation angle results in a range reduction of more than 50%. Consequently, as range is the only changing variable affecting the communication link margin, the result is a more than six decibel (dB) reduction in free space loss (L_S) from satellite AOS to the 40° minimum elevation angle constraint [15].

$$L_S = \left(\frac{4\pi d}{\lambda}\right)^2$$

$$\frac{L_{S10°}}{L_{S40°}} = \frac{\left(\frac{4\pi d_{10°}}{\lambda}\right)^2}{\left(\frac{4\pi d_{40°}}{\lambda}\right)^2} = \frac{{d_{10°}}^2}{{d_{40°}}^2}$$

$$10\log_{10}\left(\frac{2331^2}{1115^2}\right) = 6.4\ dB$$

The MC3 data set, combined with the constraints that drive its operation, is the foundation for the whole and segmented pass analysis conducted in Chapter III, ultimately resulting in a pass quality metric for ground station scheduling optimization.

THIS PAGE INTENTIONALLY LEFT BLANK

III. PASS QUALITY DETERMINATION

Chapter II reviewed several factors that may affect pass quality, such as argument of perigee rotation and the relationship between initial azimuth and maximum elevation. Additionally, it outlined the MC3 Network data set to include constraints and limitations in its collection. This chapter utilizes the data sets in order to measure pass quality for a specific ground station-satellite pair.

Using MATLAB and STK, Chapter III examines the historical passes of the PropCube satellite, Flora, as it travels overhead the PTSUR ground station. First, the thesis performs whole pass analysis, which analyzes each pass as a success or failure based on the presence of at least one preamble or decode. It then divides each PTSUR-Flora pass into segments to identify the locations within the pass where successful data downlink occurs.

A. PTSUR-FLORA WHOLE PASS ANALYSIS

In the previous chapter, this thesis verified the correlation between pass initial azimuth and maximum elevation angle. As a result, the whole pass analysis explored in this section utilizes these two metrics as the primary means to compare the link performance of each PTSUR-Flora pass. The MATLAB code used in the whole pass analysis for any MC3 ground station-satellite pair is found in Appendix C.

1. MC3 Data Set Analysis

To analyze a ground station-satellite pair's performance, the study first measures the probability of preambles and decodes to determine downlink success rate. As discussed in Chapter II, a preamble received during a pass signifies that the satellite is transmitting data. A pass with a successful decode indicates the satellite has not only transmitted data, but that some of that data has been successfully received by the target ground station and passed the CRC validation. Therefore, in order to record a transmission decode, a preamble must occur first.

Figure 23 depicts 547 PTSUR-Flora passes in the MC3 data set grouped by maximum elevation angle. Of these passes, 170 received only preambles and 198 received at least one successful preamble and decode. The remaining 179 passes did not record communication between the satellite and ground station in a Raw or KISS log file.

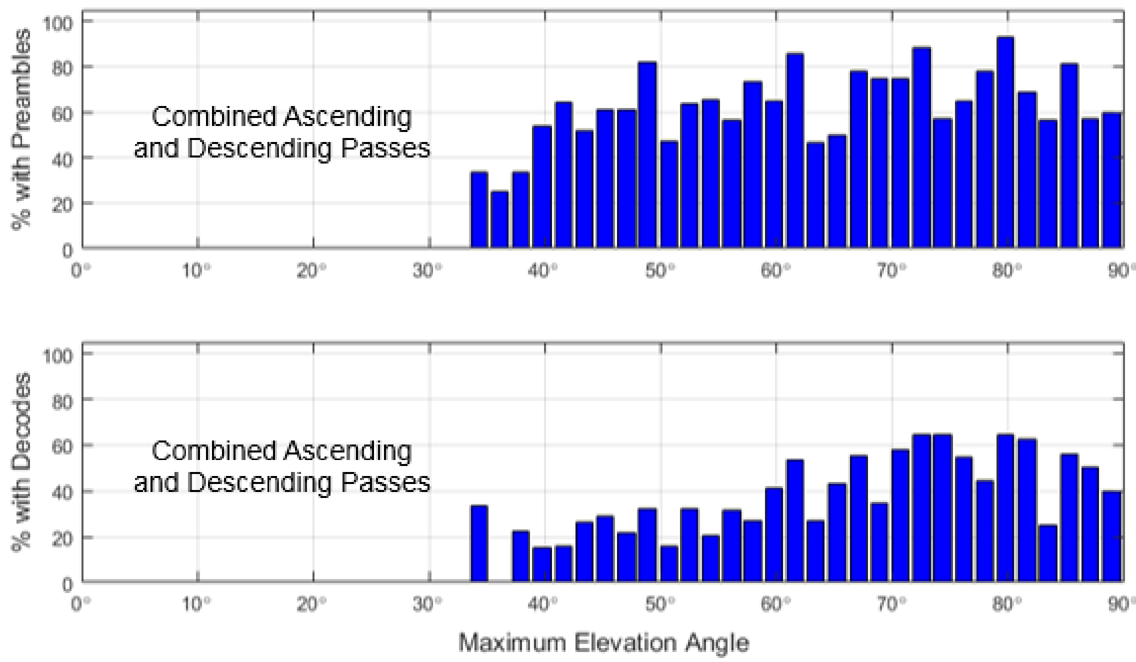


Figure 23. PTSUR-Flora Preamble/Decode Success Rate (Max Elevation)

Shorter ground station-satellite range should result in greater probability of preambles and decodes. However, Figure 23 depicts very little, if any, statistical correlation between these two metrics. This representation fails to differentiate the ascending from the descending passes as each of the elevation angles is shared by multiple initial azimuths. Therefore, to evaluate the performance of ascending and descending passes independently, the data set should be grouped by initial azimuth. Due to varying physical obstacles and noise environments surrounding a ground station, this distinction in pass direction is critical in overall pass quality assessment, as shown in Figures 24 and 25.

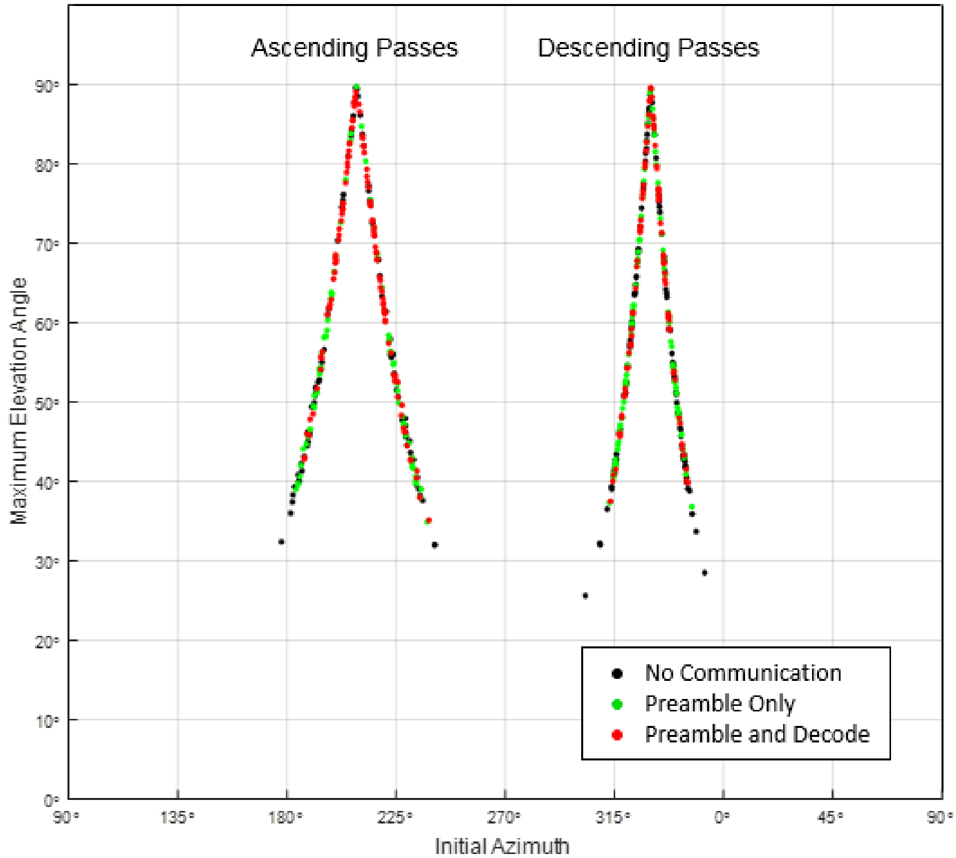


Figure 24. PTSUR-Flora Whole Pass Analysis Downlink Events

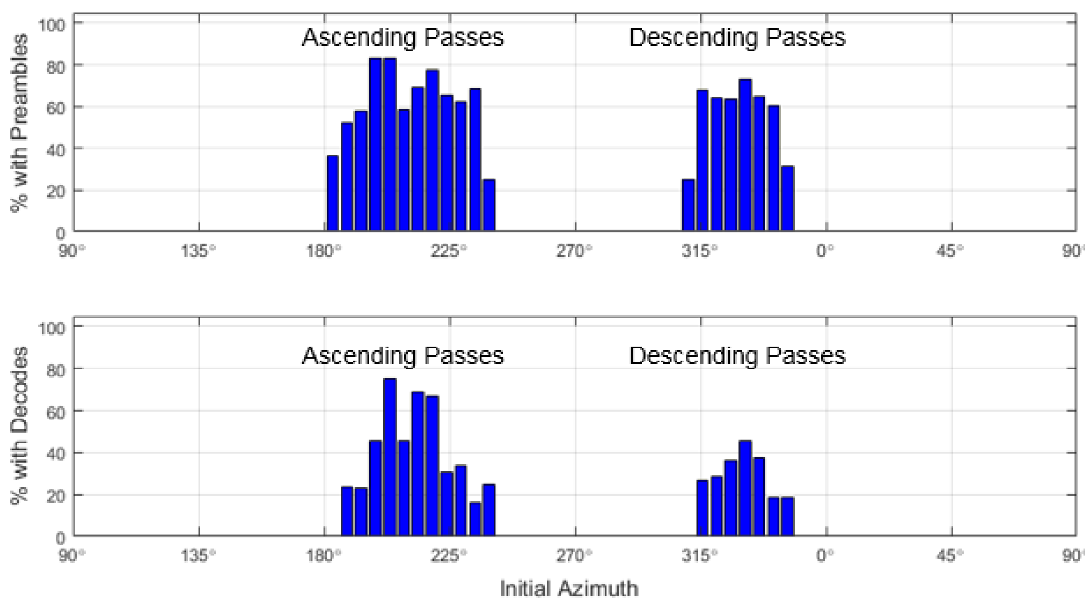


Figure 25. PTSUR-Flora Preamble/Decode Success Rate (Initial Azimuth)

The same 547 PTSUR-Flora passes are now represented by initial azimuth, rather than maximum elevation. The ascending pass group, located on the left side of the figures, shows a positive trend in data downlink as the maximum elevation angle increases. The majority of passes near the peak of this group contain successful decodes. As the maximum elevation angle decreases, the likelihood of decodes also declines. In general, the overall performance of PTSUR-Flora descending passes is poorer, which in part can be attributed to the physical obstacles surrounding the PTSUR antenna location, shown in Figure 26.

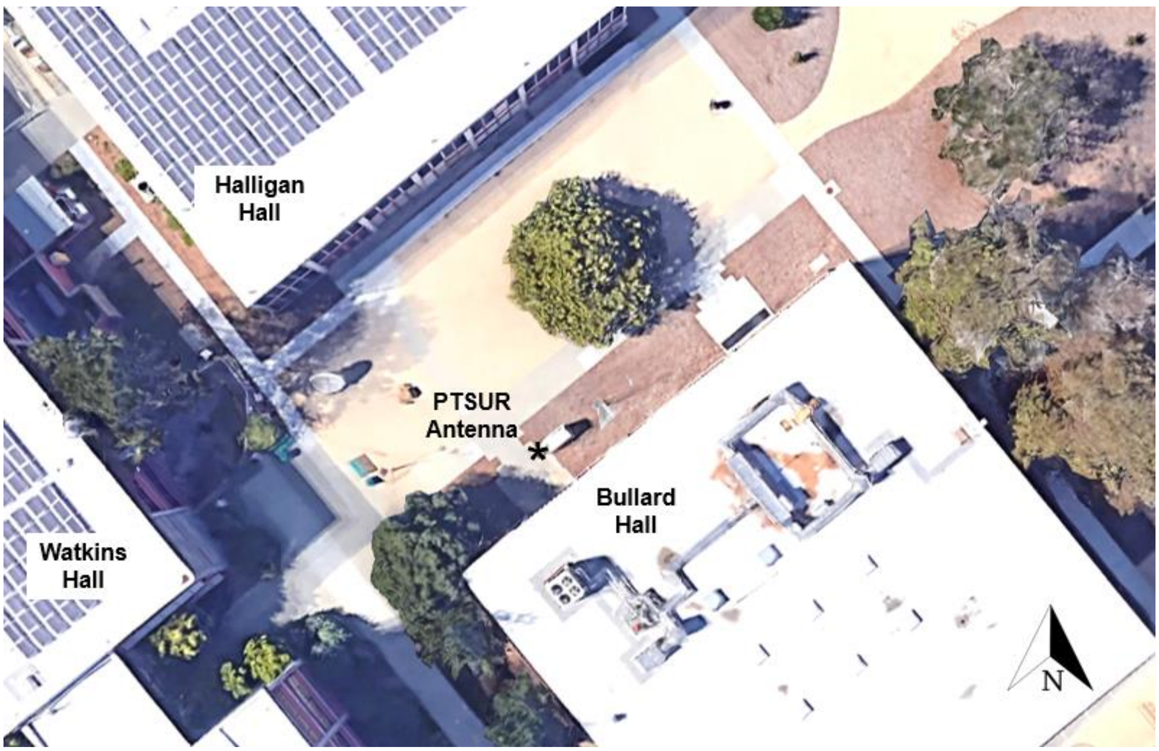


Figure 26. PTSUR Antenna Physical Obstructions. Adapted from [16].

Due to a noisy RF environment in the 914 MHz band surrounding the ground station, the MC3 operators moved the PTSUR antenna from its original rooftop location and repositioned it to an adjacent courtyard. Although the surrounding buildings now shield the antenna from much of the local 914 MHz noise, they also obstruct incoming satellite transmissions. For example, Bullard Hall, located less than 10 feet from the antenna to the southeast, is responsible for hindering communication between the PTSUR ground station and the egressing portion of descending passes. The complete effect of the surrounding physical barriers and noise environments on pass value is further analyzed in the PTSUR-Flora segmented pass analysis in Section B.

Although depicting valuable information concerning overall preamble and decode performance, Figures 24 and 25 show no visible disparity between passes that received a single decode and those that received multiple decodes. Since data downlink rate is directly linked to the number of decodes within a pass, the correlation between maximum elevation and pass performance is not evident in this format. In order to provide a metric to predict the quality of future PTSUR-Flora accesses, the final portion of the whole pass analysis calculates average decode rate to account for multiple events in a single pass. Again, the data set is grouped by pass initial azimuth, as shown in Figure 27.

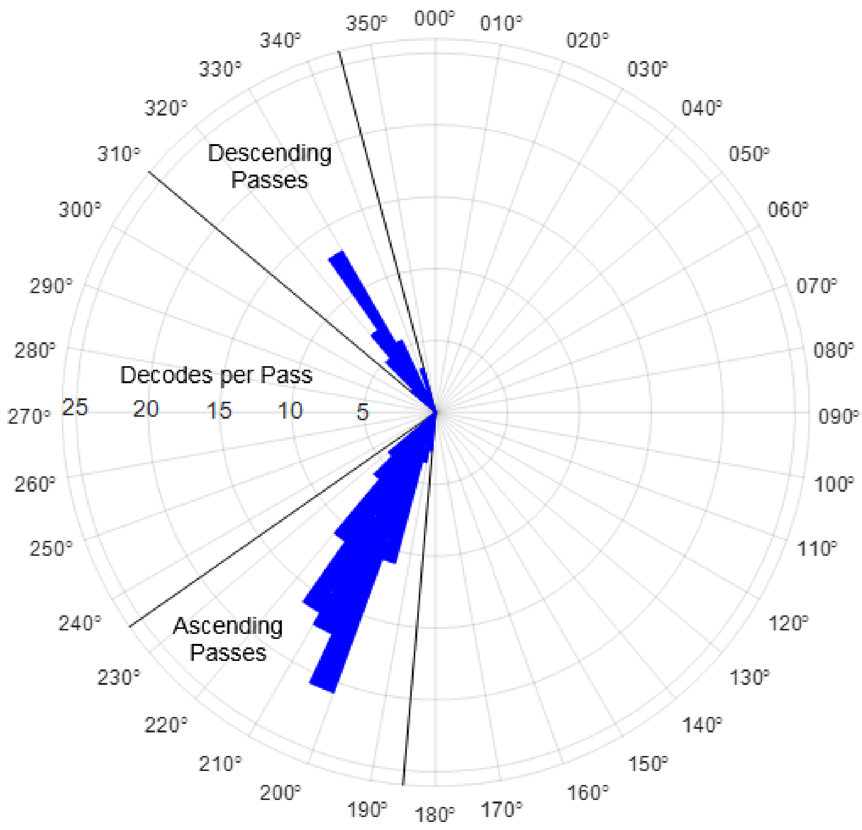


Figure 27. PTSUR-Flora Average Decodes per Pass

The lines adjacent to the ascending and descending pass groups represent the *acceptance region* in which passes can take place based on Flora's orbital inclination of 64.8°, PTSUR ground station latitude of 36.6°, and a 40° minimum pass elevation angle threshold. Within this acceptance region, the northern azimuth corresponds to a smaller linear distance when projected onto a spherical Earth and, therefore, appears narrower than that of the southern azimuth region.

As the initial azimuths in both the ascending and descending acceptance regions approach the corresponding nadir pass elevation of 90°, the ground station-satellite link achieves its smallest range. As a result, these particular passes are least affected by free space signal loss, and on average, provide the most decodes. Additionally, passes with higher elevation angles are less susceptible to the effects of physical obstacles that surround the ground station antenna. As the initial azimuths, and therefore maximum elevations, move further from the nadir pass, the average number of decodes per pass decreases.

Whole pass analysis provides two benefits to the CubeSat operator. First, it gives the user a means to predict the expected performance of upcoming passes. Second, whole pass analysis provides a holistic view of each ground station and its surrounding features that may obstruct downlinked data. With only the pass initial azimuth, the operator can now predict the likelihood of downlink and the number of decodes via whole pass analysis. By characterizing the value of future passes as a function of initial azimuth, CubeSat networks can optimize their antenna scheduling by allocating antenna assets to passes with greater likelihood of success. As an example, using the associated MATLAB script found in Appendix C, an upcoming PTSUR-Flora pass with a 220 ± 2.5° initial azimuth is predicted to perform as per the historical values shown in Table 7.

Table 7. PTSUR-Flora Whole Pass Analysis Example (AZo 220 ± 2.5°)

Pass Type	Ascending
Max El	65°
% Preamble	78%
% Decode	67%
Decodes / Pass	11.2

2. TLE Data Set Analysis

As described in Chapter II, perigee rotation causes the duration and average range of the MC3 CubeSat passes to change over time. As the pass duration increases, the ground stations have a longer opportunity to communicate with the satellite, but longer access comes at the expense of a higher average altitude and increased free space loss. Based on Curtis's two-body approach [11], this study approximates the change in CubeSat signal strength from perigee to apogee using an average eccentricity (e) and semi-major axis (a) of 0.0188 and 7,027 km, respectively, as calculated from Flora's historical TLE data set.

$$r = a\ \frac{1 - e^2}{1 + e\ cos\theta}$$

$$\theta_{APOGEE} = 180° \qquad \theta_{PERIGEE} = 0°$$

$$R_a = a\ (1 + e) \qquad R_p = a\ (1 - e)$$

$$R_a = 7027\ (1 + 0.0188) \qquad R_p = 7027\ (1 - 0.0188)$$

$$R_a = 7159\ km \qquad R_p = 6895\ km$$

$$Alt_a = 781\ km \qquad Alt_p = 517\ km$$

$$10 \log_{10} \left(\frac{781^2}{517^2}\right) = 3.6\ dB\ loss$$

In the case of Flora's orbit, the range difference for apogee passes results in less than half signal strength when compared to perigee. Throughout the orbital lifetime of the satellite, the magnitude of the signal strength loss will vary as Flora's perigee rotates. Figure 28 shows Curtis's method to determine average perigee rotation rate. Using the historical TLE mean semi-major axis, inclination, and eccentricity values of 7,027 km, 64.8°, and 0.0188, respectively, the study calculates Flora's perturbation rate to be 0.328° per day [11]. Therefore, a single 360° argument of perigee rotation cycle takes approximately three years and three days to complete.

$$\dot{\overline{\omega}} = -\left[\frac{3}{2}\frac{J_2\sqrt{\mu}R^2}{a^{7/2}(1-e^2)^2}\right]\left(\frac{5}{2}\sin^2 i - 2\right)$$

Figure 28. Average Argument of Perigee Rotation Rate. Source: [11].

When the argument of perigee places Flora perigee over the ground station, on either the ascending or descending side of the orbit, the passes are at minimum range and minimum free space loss. Accordingly, the passes during this period provide increased opportunity for successful data downlink. On the other hand, when apogee is coincident with the ground station latitude, 180° offset from the previous conditions, the pass range will be at a maximum. These accesses yield the greatest free space loss and a decreased chance of effective data flow, which reduces their pass value. To calculate the arguments of perigee associated with the minimum and maximum free space loss conditions, this study rotates the Cartesian coordinate frame about the x-axis to yield argument of perigee as a function of satellite inclination and ground station latitude, as shown in Figures 29 and 30.

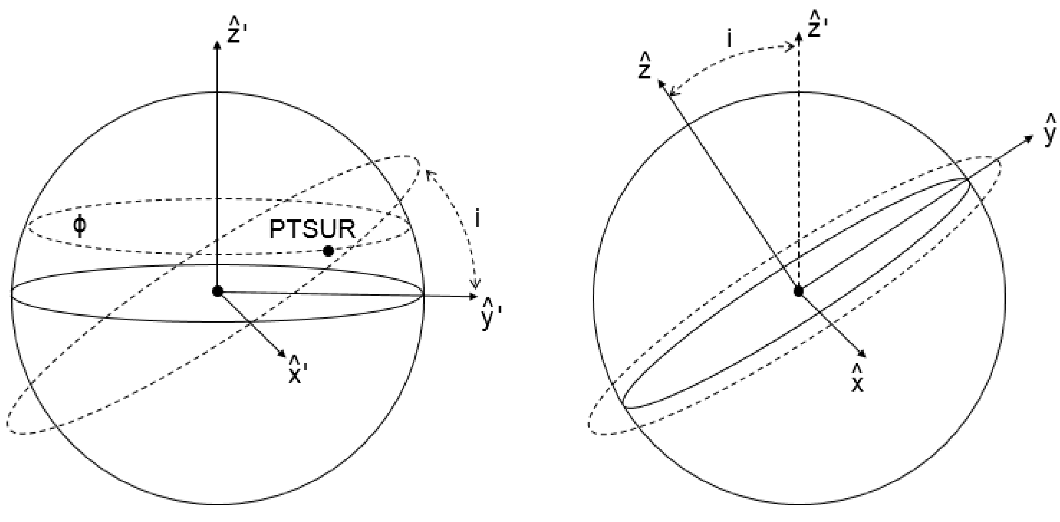
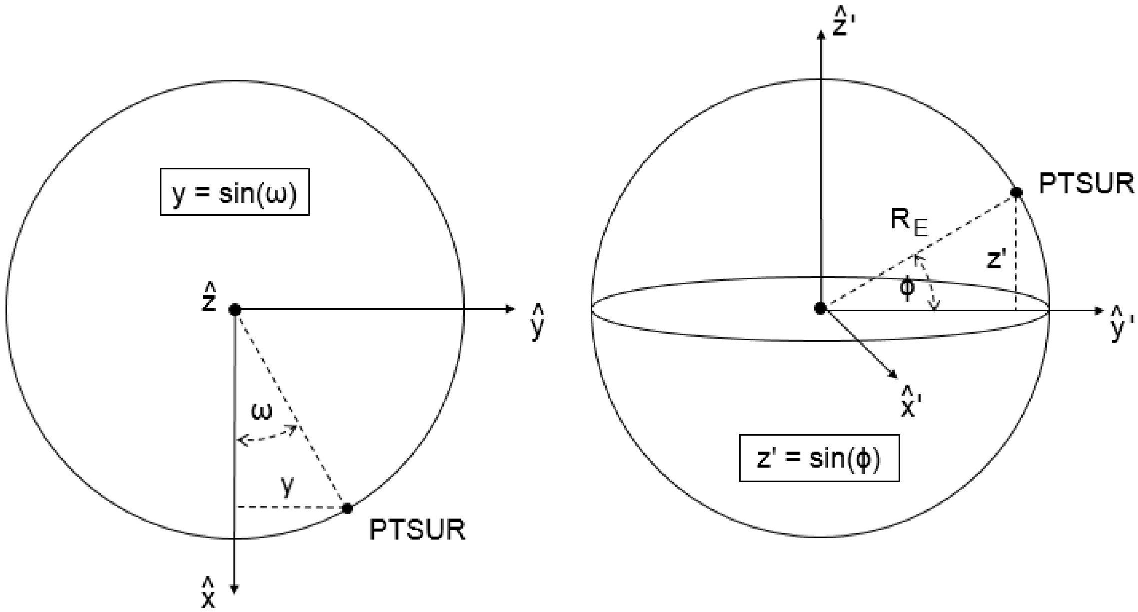


Figure 29. Cartesian Coordinate Frame Rotation



$$Rotation\ Matrix\ about\ x - axis\ to\ determine\ z'$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(i) & -\sin(i) \\ 0 & \sin(i) & \cos(i) \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \end{bmatrix}$$

$$z' = y\, sin(i)$$

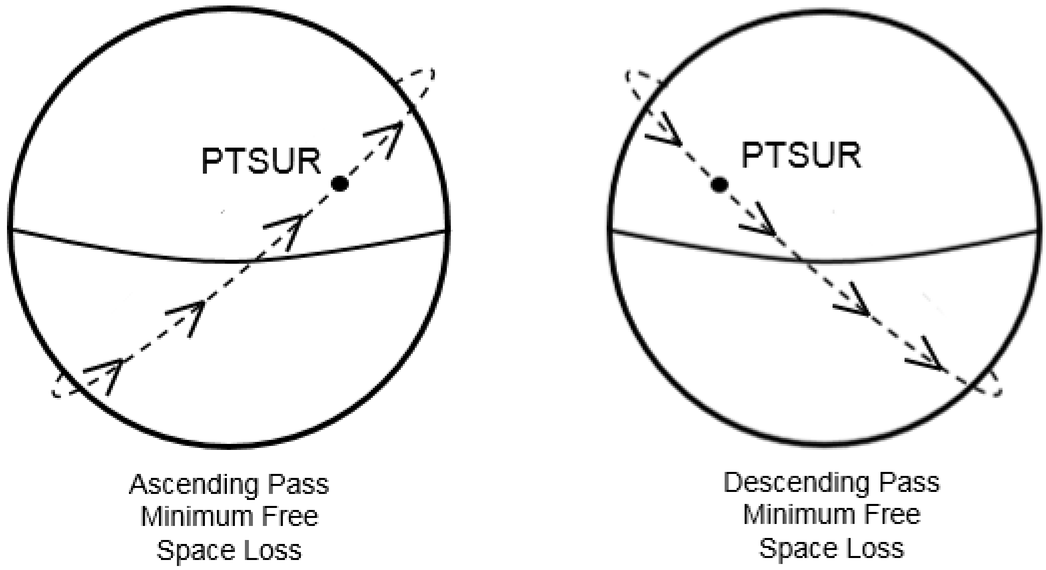
$$y = \sin(\omega) \qquad\qquad z' = \sin(\phi)$$

$$\sin(\phi) = \sin(\omega)\sin(i)$$

$$\omega = \sin^{-1}\left(\frac{\sin(\phi)}{\sin(i)}\right)$$

Figure 30. Coordinate Frame Geometric Relationships

Applying the PTSUR ground station latitude and Flora orbital inclination to the general equation, the study now calculates the minimum free space loss perigee locations for both ascending and descending PTSUR-Flora passes, as shown in Figure 31.



$$\omega_A = \sin^{-1}\left(\frac{\sin(\phi)}{\sin(i)}\right) \qquad \omega_D = 180 - \sin^{-1}\left(\frac{\sin(\phi)}{\sin(i)}\right)$$

$$\omega_A = \sin^{-1}\left(\frac{\sin(36.6°)}{\sin(64.8°)}\right) \qquad \omega_D = 180 - \sin^{-1}\left(\frac{\sin(36.6°)}{\sin(64.8°)}\right)$$

$$\omega_A = 41.2° \qquad \omega_D = 138.8°$$

Figure 31. PTSUR-Flora Minimum Free Space Loss Arguments of Perigee

Using the Flora TLE data set, this study calculates the historical location of perigee within the PropCube orbit, its relationship with the ground station latitude, and the associated pass CPA minimum range. Similar to the method used by Vertat, et al [9], communication distance, and, therefore, free space loss can be calculated based on a CubeSat orbital altitude and elevation angle. The MATLAB code, found in Appendix D, extracts the historical TLE date, semi-major axis, eccentricity, and argument of perigee information. Per Curtis's elliptical orbit calculations [11], the link range at varying elevation angles is calculated using the true anomalies corresponding to both ascending and descending ground station passes. The script then determines the historical signal loss as compared to a 500 km range, 0 dB baseline, as shown in Figures 32 and 33.

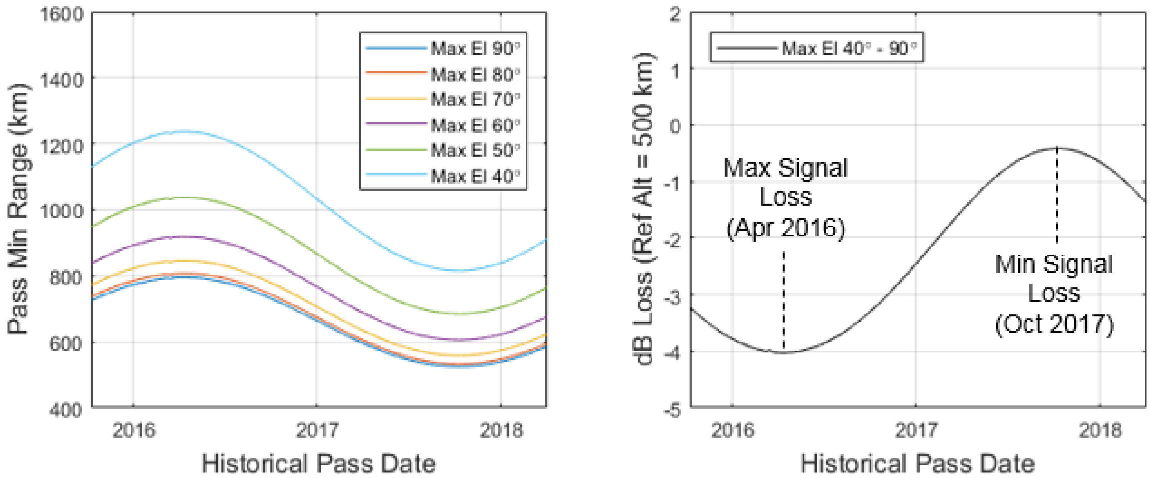


Figure 32. Historical PTSUR-Flora Ascending Pass Minimum Range and Free Space Loss

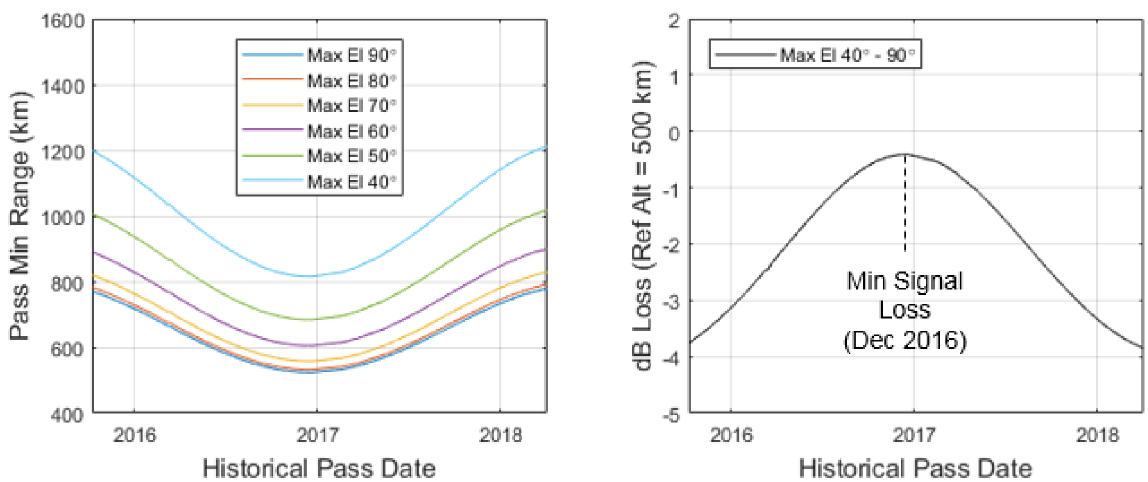


Figure 33. Historical PTSUR-Flora Descending Pass Minimum Range and Free Space Loss

Similar to the pass duration discussion in Chapter 2, as the range and free space losses vary with perigee rotation, there are two unique times when the argument of perigee is located equidistant from the ascending and descending pass locations. When perigee is located at the orbit's highest latitudes, in either the Northern Hemisphere (ω = 90°) or Southern Hemisphere (ω = 270°), there is no difference in free space loss between the two pass directions. Therefore, based only on predicted signal losses, one type of directional pass should not be favored over the other, as shown in Figure 34.

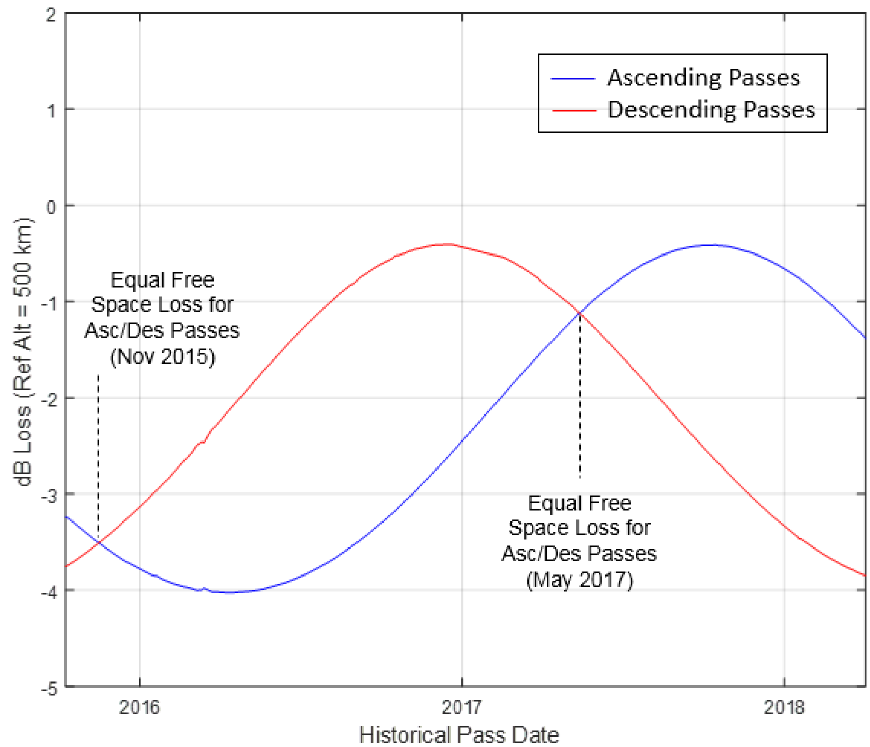


Figure 34. Historical PTSUR-Flora Ascending vs Descending Free Space Loss

Although the historical data provides an overview of past signal loss due to perigee rotation, in order to account for changes in future MC3 passes, this thesis must predict performance based on argument of perigee location. To more accurately calculate the minimum/maximum signal loss time periods, STK's propagator extrapolates the current Flora TLE and correlates the predicted argument of perigee to the PTSUR ground station latitude. As shown in Figure 35 and Table 8, when Flora's perigee is located over the ground station for either direction pass, the free space losses are at a minimum, as seen at 41.2 and 138.8°. On the other hand, when offset 180°, the free space losses are maximum at arguments of perigee of 221.2 and 318.8°. Instances of minimum free space loss occur when argument of perigee is between 0 and 180°, as the PTSUR ground station is located in the Northern Hemisphere. Conversely, the maximum free space loss conditions occur when argument of perigee is greater than 180°, located in the Southern Hemisphere.

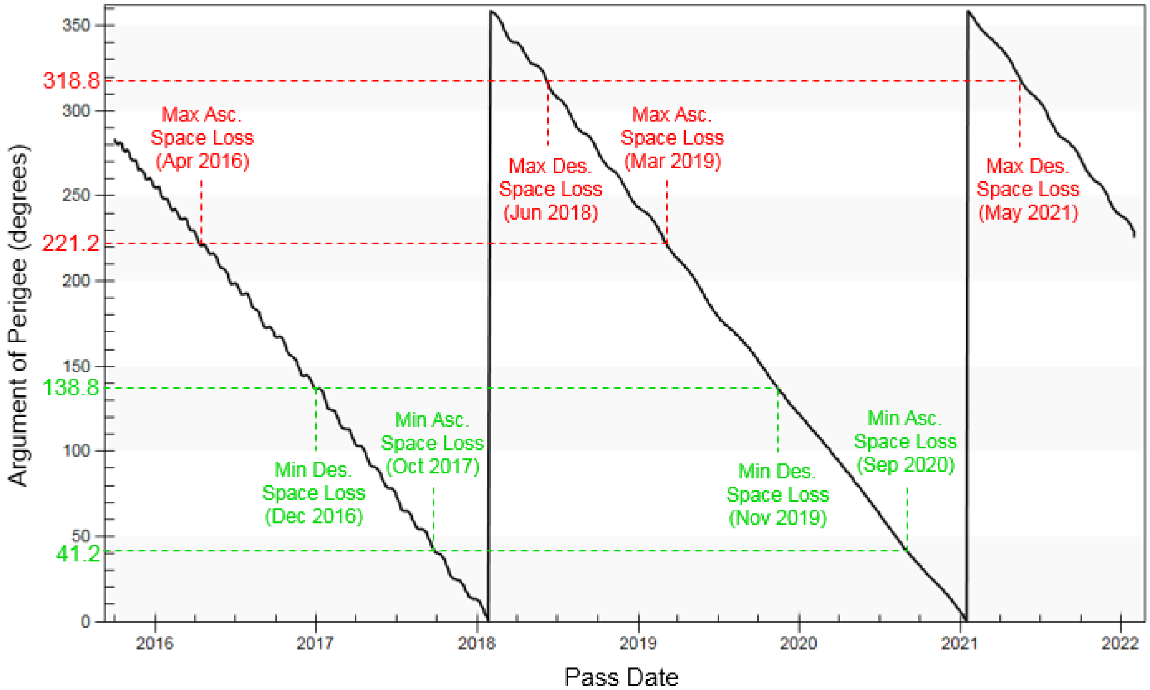


Figure 35. PTSUR-Flora Free Space Signal Loss due to Argument of Perigee

Table 8. PTSUR-Flora Free Space Signal Loss due to Argument of Perigee

Date	Pass Type	Arg. of Perigee (ω)	Free Space Signal Loss
April 2016	Ascending	221.2°	Maximum
December 2016	Descending	138.8°	Minimum
May 2017	Asc. & Des.	90.0°	Equal
October 2017	Ascending	41.2 °	Minimum
June 2018	Descending	318.8°	Maximum
October 2018	Asc. & Des.	270.0°	Equal
March 2019	Ascending	221.2°	Maximum
November 2019	Descending	138.8°	Minimum
April 2020	Asc. & Des.	90.0°	Equal
September 2020	Ascending	41.2°	Minimum
May 2021	Descending	318.8°	Maximum
September 2021	Asc. & Des.	270.0°	Equal

As shown in Figures 36 and 37, the ascending and descending pass signal strength predictions are represented as a percent of maximum value based on location of perigee during particular months. A signal strength of 100% represents perigee location over the ground station, and, therefore, the minimum free space loss condition. Conversely, a signal strength of 0% represents apogee location over the ground station resulting in maximum free space loss. All intermediate values are based on a linear scale from minimum to maximum loss throughout the analysis period. The historical preamble and decode rates for PTSUR-Flora are overlaid for each pass type in order to associate the theoretical signal strength with link performance, and, therefore, correlate pass value with perigee location.

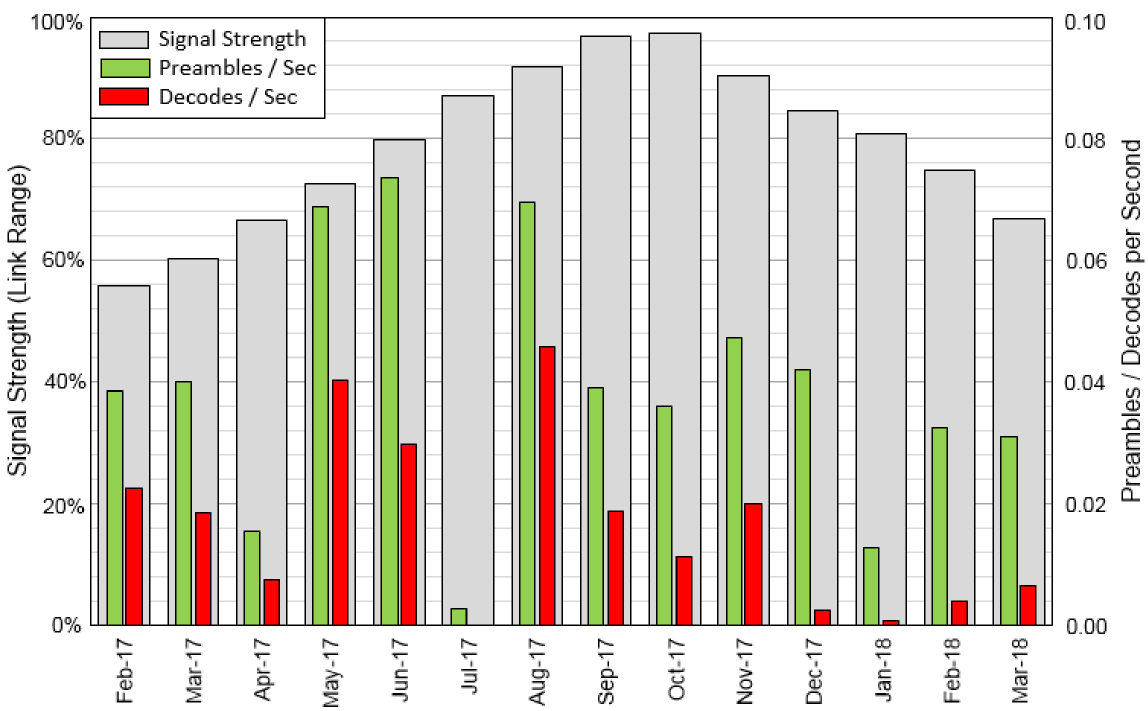


Figure 36. PTSUR-Flora Ascending Pass Perigee Location Signal Strength

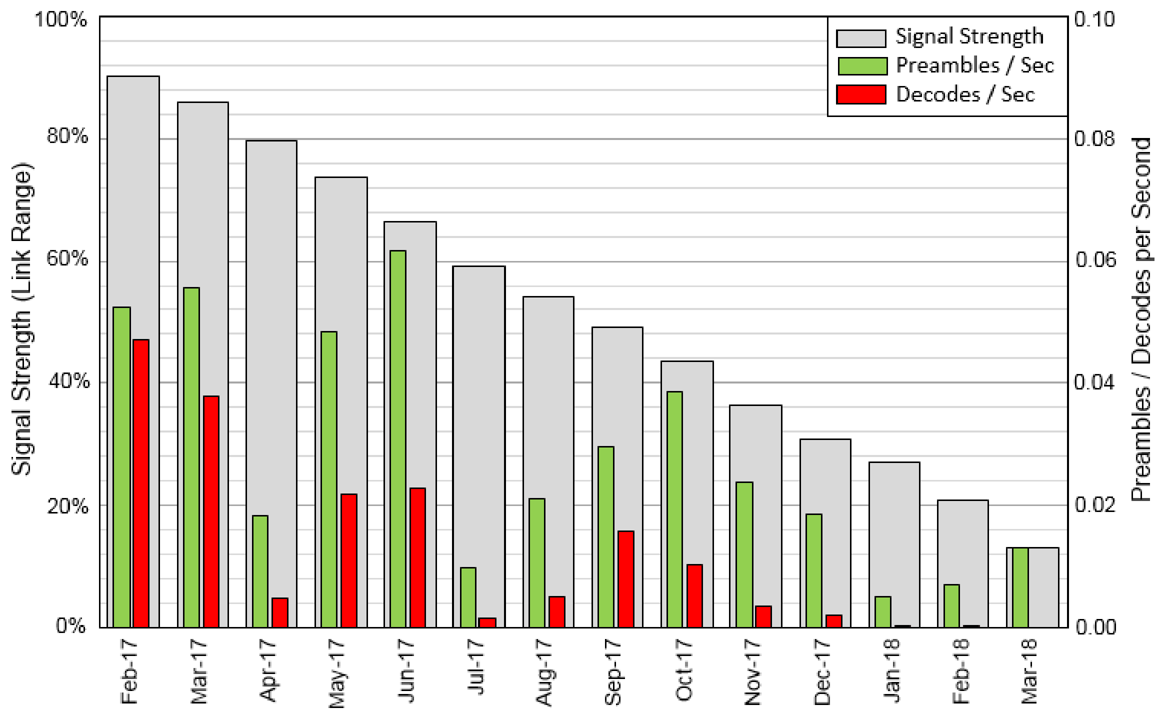


Figure 37. PTSUR-Flora Descending Pass Perigee Location Signal Strength

In theory, if an operator has a specific satellite and time of year, this thesis's pass value metric provides information on which direction pass, ascending or descending, has the best opportunity for communication. This value metric can be used as a multiplier for pass quality, where passes nearer to the argument of perigee with lower altitudes are given a higher pass value as compared to passes near apogee. However, the current data set does not provide enough statistical significance to develop this pass value multiplier. Although link distance is one of the factors affecting pass quality, it is clear that it is not the only variable, and more data is required to assess the true influence of this factor.

Whole pass analysis of preamble and decode likelihood provides the foundation for the pass value function. However, in order to best optimize the ground station network, this thesis must also analyze individual pass segments and determine the most influential factors for successful data reception.

B. PTSUR-FLORA SEGMENTED PASS ANALYSIS

In Section A's whole pass analysis, each access was deemed a success or failure based on the presence of a preamble or decode. Although offering insight into the PTSUR-Flora downlink performance, the results only offer general trends of entire passes and fail to differentiate the unique characteristics of each pass segment. This thesis, in order to develop a pass value function, requires a more granular analysis of the pass segments and their data downlink opportunity. Prior to analysis, using the MATLAB code in Appendix E, the track and log file data set is purged of extraneous files. The MATLAB code used in the segmented pass analysis for any MC3 ground station-satellite pair is found in Appendix F.

The first step to analyze specific segments of a pass is to determine the satellite azimuth and elevation during successful downlink events. As described in Chapter II, the log file for each pass contains the preamble and decode times, and the track file records position information relative to the ground station. By correlating the downlink event timestamp to the azimuth and elevation, the study amassed a list of satellite locations during every recorded preamble and decode throughout the MC3 Network. Three passes by the PropCube Flora over the PTSUR ground station are shown in Figure 38, displaying the locations where successful communication from satellite to ground occurred within each pass.

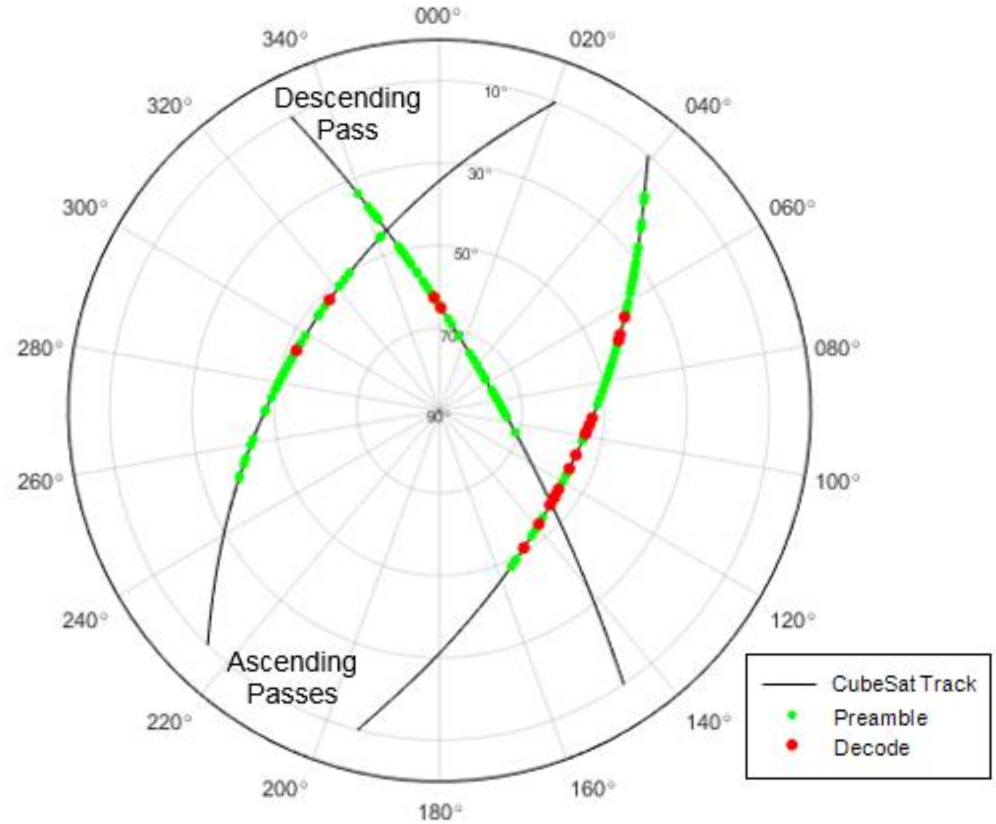


Figure 38. PTSUR-Flora Sample Pass Preamble and Decode Locations

As expected, the majority of downlink events occur when the PropCube is at higher elevation angles. Figure 38 also highlights the difficulty in establishing data transfer at low elevation angles as there are few preambles below 30° and no decodes below 40°. When the satellite is near the horizon, data flow is unlikely due to free space loss and physical obstacles. As explored in Chapter II, and as the basis for the MC3 Network pass constraints, the likelihood of successful communication with the PropCube increases as the elevation angle increases.

By compiling all preambles and decodes for a ground station-satellite pair on a single plot, this study begins to distinguish azimuth-elevation regions with greater likelihood of link success, as shown in Figure 39. It also begins to identify specific regions where data transfer is unlikely. These measured communication gaps can be correlated to physical barriers or suspected noise environments surrounding the ground station antenna.

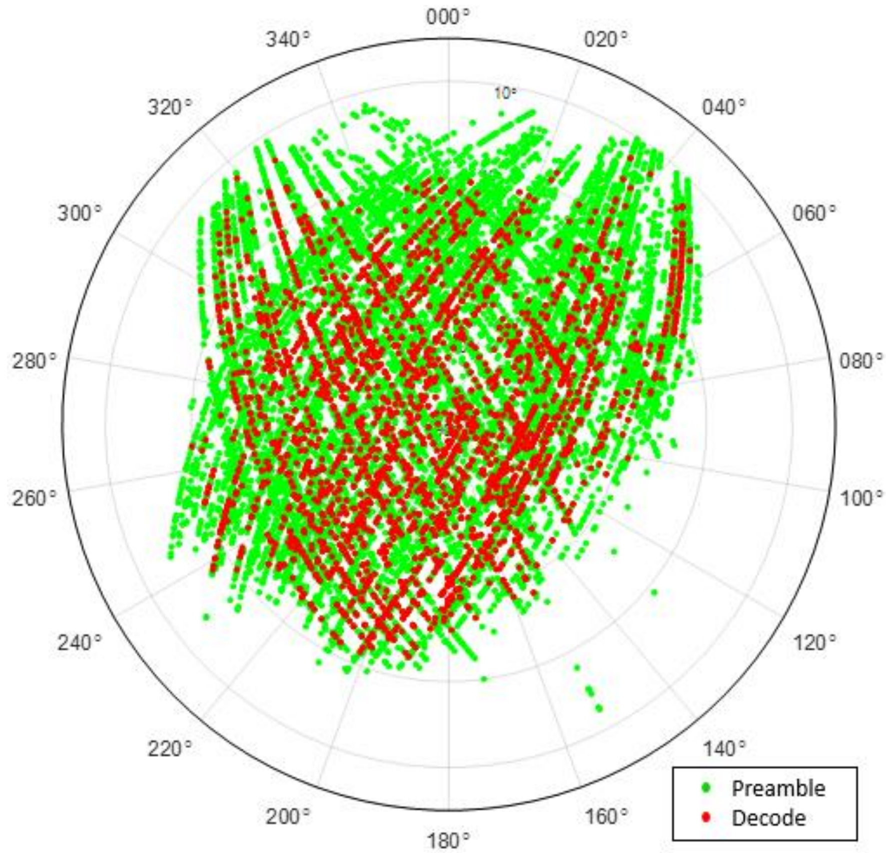


Figure 39. PTSUR-Flora All Historical Pass Preamble and Decode Locations

With all historical preambles and decodes represented on a single figure, it is evident that there are multiple locations where the surrounding environment obstructs communication. Bullard Hall, to the southeast of the PTSUR antenna, blocks most transmissions in the direction typically associated with descending pass egress. In addition, a tree to the west of Bullard Hall obstructs communication in the direction of ascending ingress, and a magnolia tree within the courtyard creates a transmission notch in the ascending egress segment. When assigning pass value for schedule prioritization, operators must consider the aggregate effects of all obstacles, as shown in Figures 40, 41, and 42 for the PTSUR case.

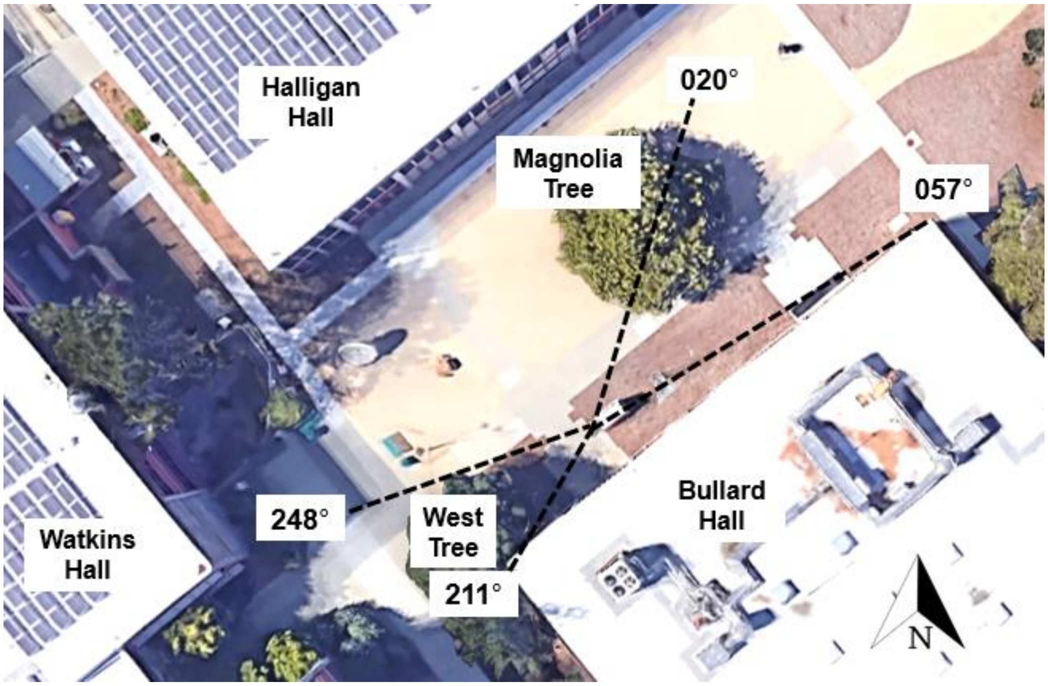


Figure 40. PTSUR Ground Station Obstacle Overlay. Adapted from [16].

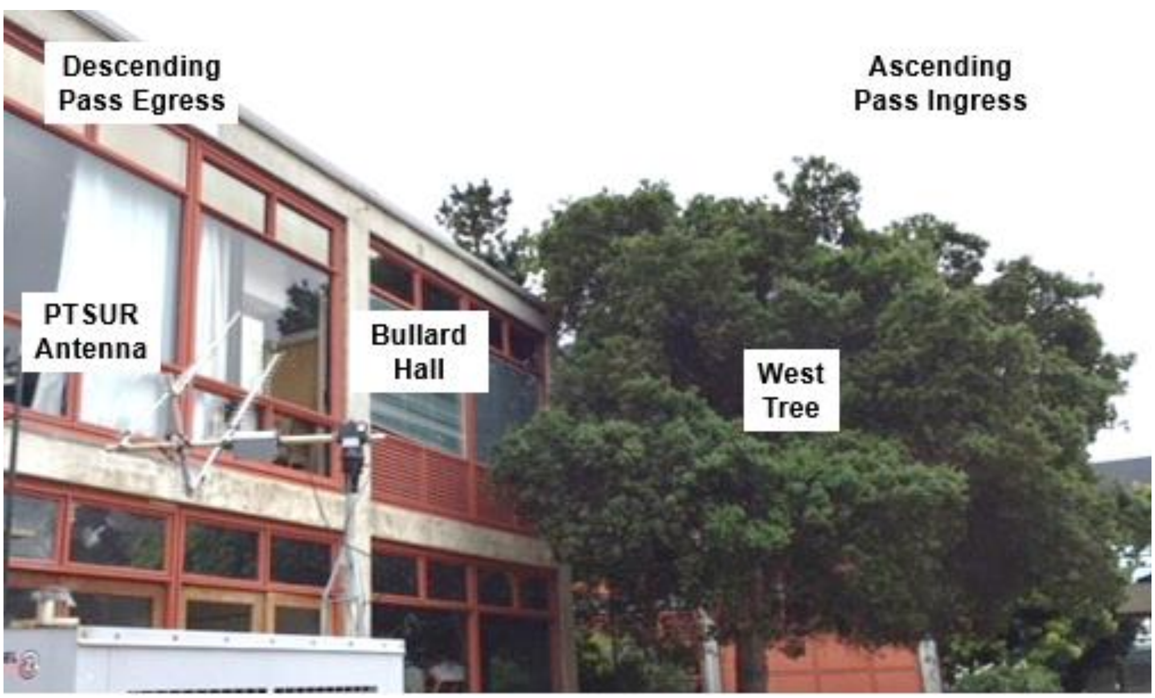


Figure 41. PTSUR Ground Station Obstacles

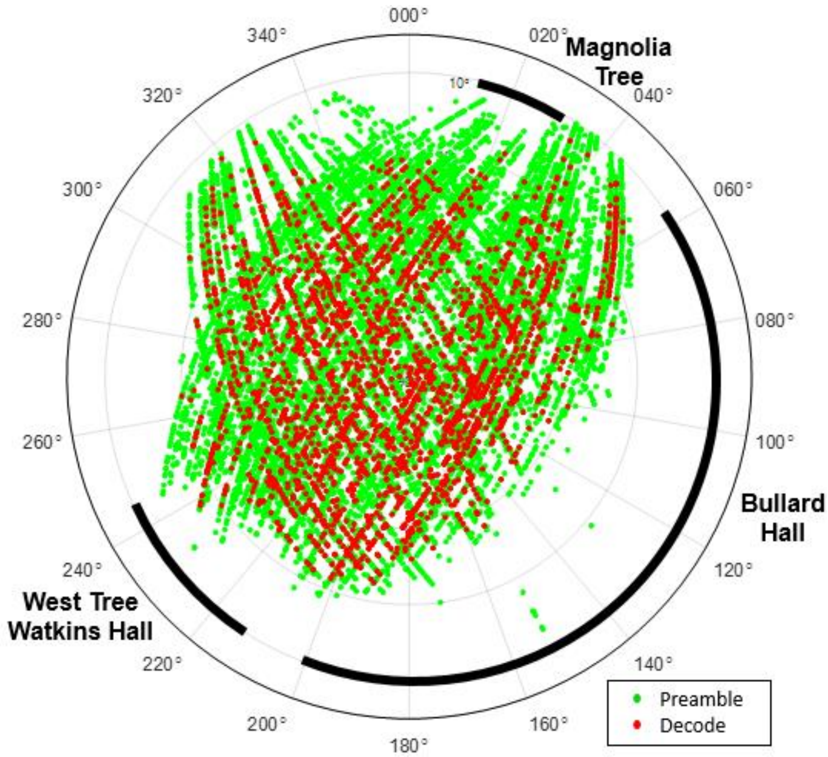


Figure 42. PTSUR-Flora Effects of Known Interference

Although Figure 42 represents the azimuth-elevation of all downlink events, it does not differentiate locations that received a single transmission from those that received many. In order to account for multiple events at the same azimuth-elevation, more value should be given to pass segments with denser data flow. Therefore, the remainder of the segmented pass analysis measures pass quality via the density of successful decodes, as shown in Figure 43.

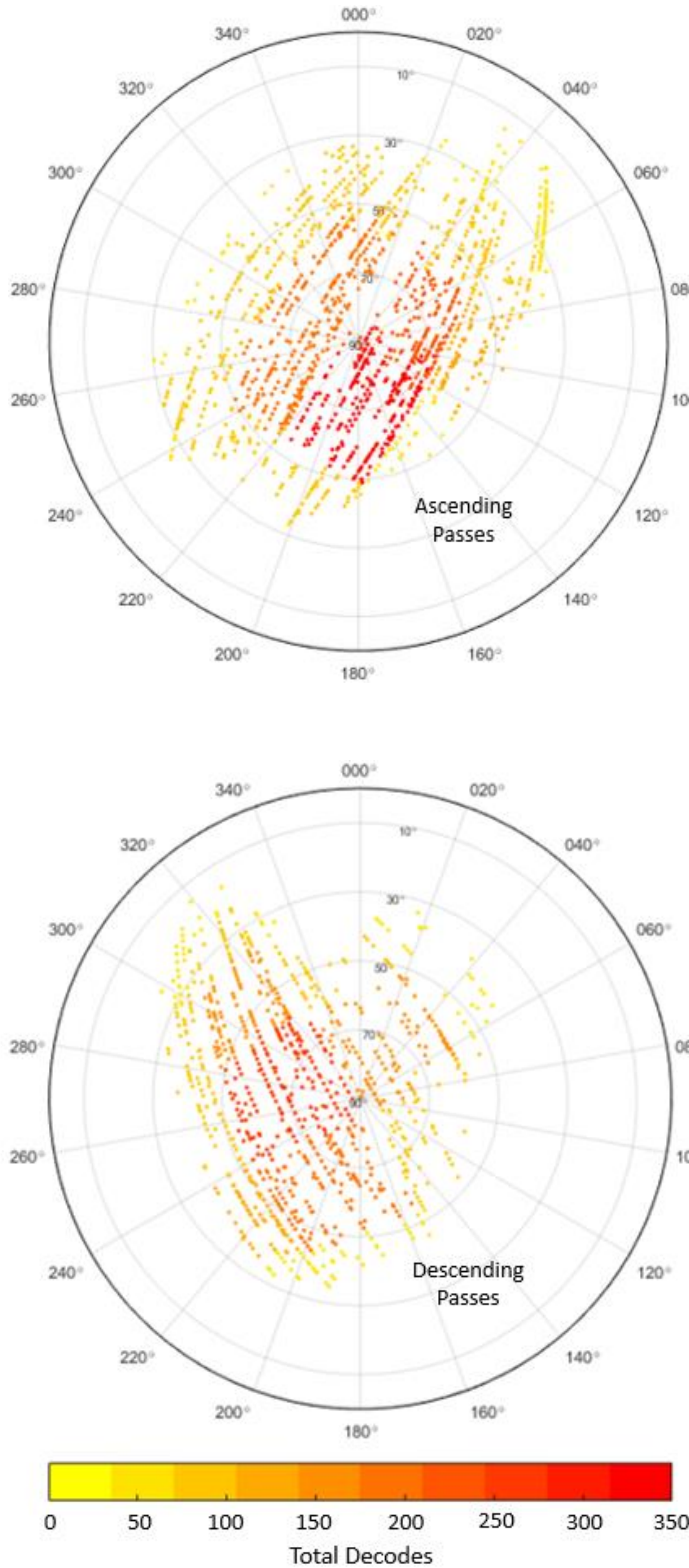


Figure 43. PTSUR-Flora Ascending and Descending Pass Decodes

Each PTSUR-Flora pass is divided into 10 segments of equal duration. Since access times vary throughout the data set, the segments represent 10% of their respective total pass duration. The individual data points within each pass now depict the decode density for that particular segment, thus accounting for multiple decodes within the same region. The received transmission density depicted still, however, only measures the total historical decodes in those particular pass segments. Since not all pass segments are equally represented in the data set, that is, some segments have a larger sample size, these heat maps must be normalized to a measure of decodes per second. Additionally, the polar representation depicts the decode density at particular locations, rather than at specific times within the pass. An operator may be aware that there is a higher likelihood of receiving transmissions at a certain azimuth-elevation, but, in order to optimize scheduling for conflicted satellite passes, a more practical measure of segment quality is decodes per second as a function of elapsed time.

Minelli [10] predicts the theoretical segment value via time by modeling the free space loss throughout a pass and correlating expected data flow to the available signal strength. To better understand actual value of passes as a function of pass segment time and initial azimuth, let us first produce a representation of theoretical signal strength by combining Minelli's approach with the initial azimuth and pass segment time binning, as shown in Figure 44. From all historical PTSUR-Flora MC3 track data, the average range from satellite to ground station, as compared to a 500 km reference altitude, is used to calculate expected free space loss. These ranges and their associated losses are binned first by their initial azimuths and then by time segment within the pass. As expected, the initial azimuths corresponding to nadir passes benefit from decreased range, and therefore, have the least free space loss near the mid-point of their pass. As the initial azimuths move towards lower elevation passes, the free space loss increases due to longer distances between satellite and ground station.

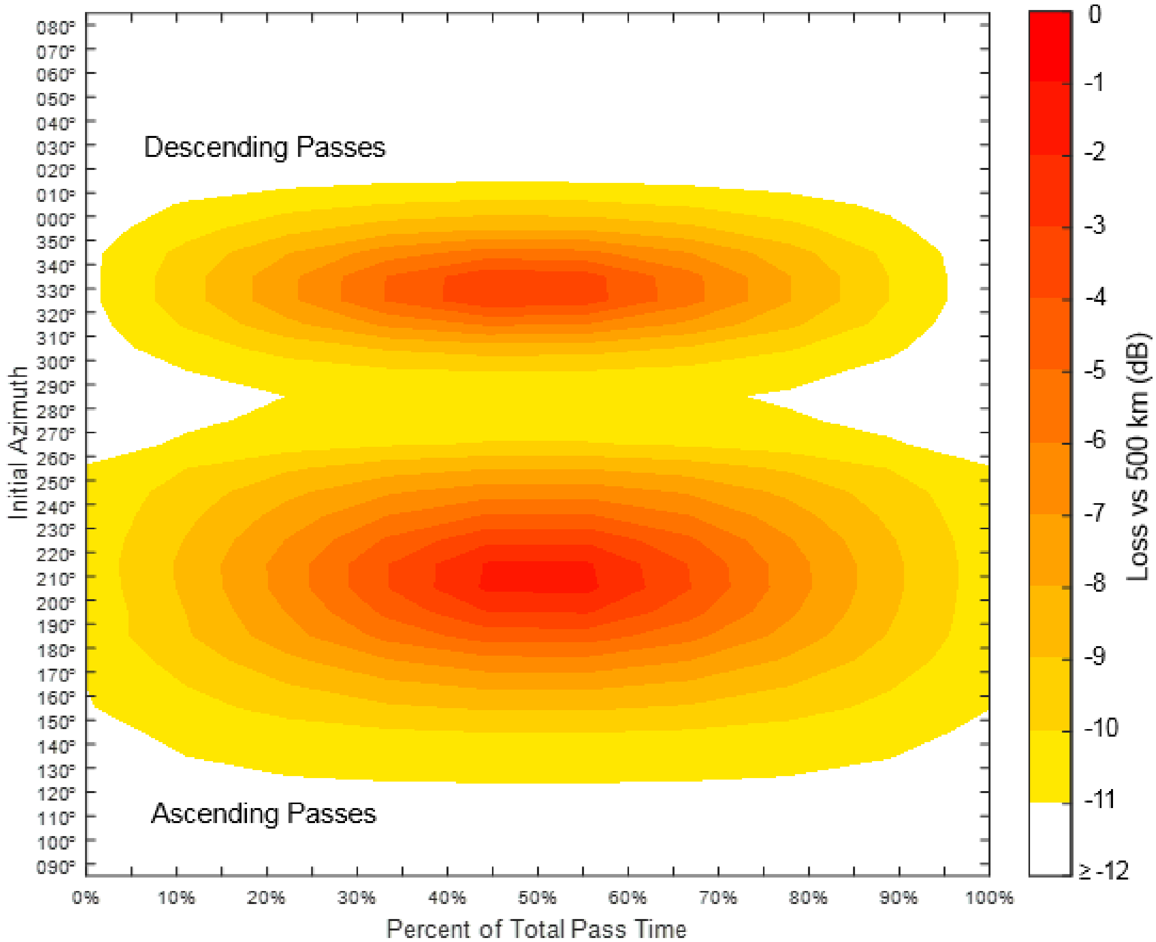


Figure 44. PTSUR-Flora Theoretical Free Space Loss without 40° Minimum Elevation Angle Constraint Heat Map

As discussed in Chapter II, the software only schedules passes with a certain minimum elevation angle to reduce the number of accesses with low likelihood of communication. Figure 44 represents the entire set of PTSUR-Flora pass tracks, to include access data prior to implementation of this software constraint. As such, it depicts various pass segments that reside outside the current pass acceptance regions. To more accurately represent the data for future passes, considering current MC3 ground station collection constraints, Figure 45 shows only the data that satisfies the 40° minimum elevation angle threshold.

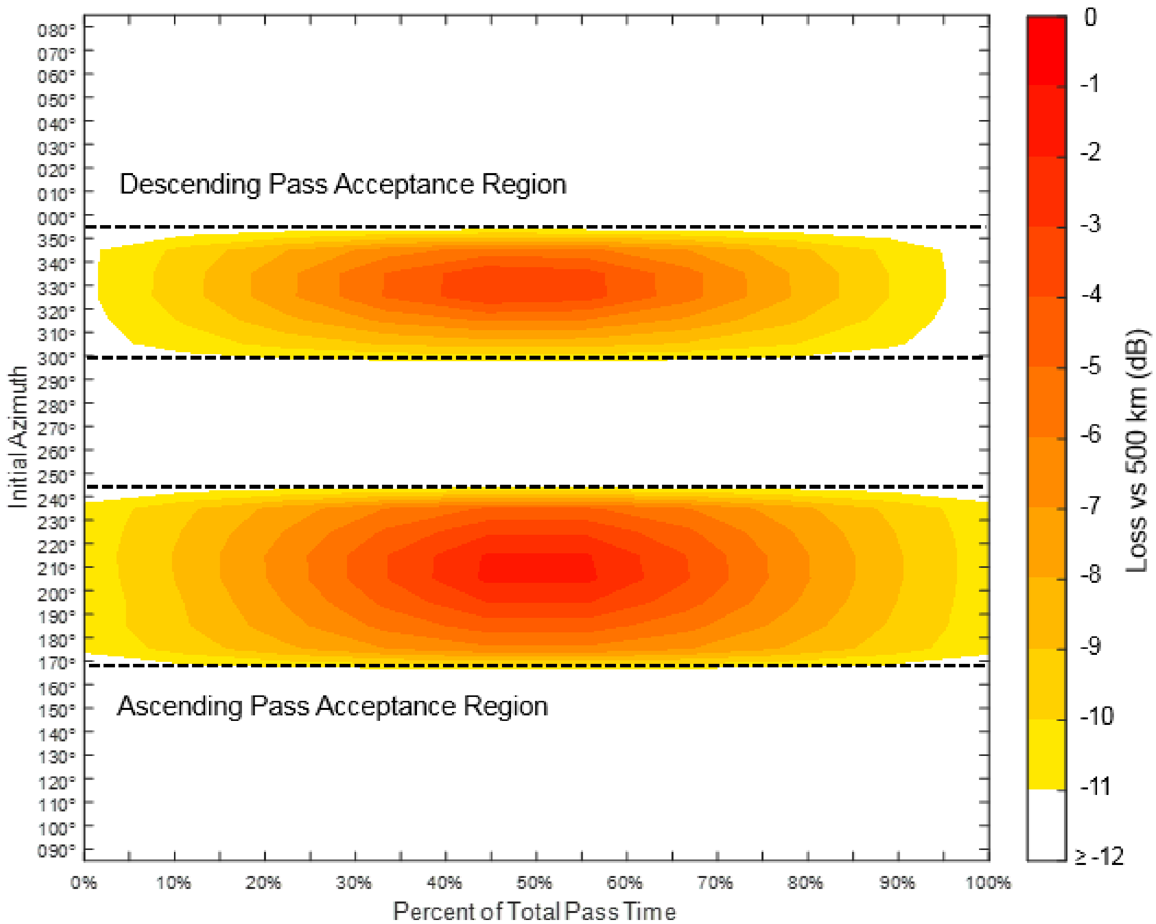


Figure 45. PTSUR-Flora Theoretical Free Space Loss with 40° Minimum Elevation Angle Constraint Heat Map

By presenting the data in this format, with only the ground station, satellite, and initial azimuth, the MC3 operator can predict the segments of a pass with the greatest downlink likelihood. As expected, based on the theoretical approach, these regions are near the pass CPA. In the case of conflicting satellite passes, the predicted signal losses at various times throughout the pass should influence the appropriate time to shift antenna tracking priority. By taking a transect of this plot at the appropriate pass initial azimuth, the signal losses are predicted based on the position of the satellite as a function of elapsed time, a concept that will be further explored in Chapter IV.

Figure 45 provides a theoretical representation of the pass quality based on link budget. However, these calculations do not account for noise environments, physical obstructions, satellite orientation, and other real-world factors that may affect the ability to receive data from the PropCubes. Therefore, using historical performance data is the only way to accurately predict pass quality via initial azimuth and pass elapsed time. By substituting the signal loss metric with that of decodes per second from the MC3 data set, past link performance from real-world information that accounts for noise environments and obstructions can be plotted, as shown in Figure 46, and used to predict future link performance. The effects of physical obstacles surrounding the PTSUR ground station are clearly seen in Figure 46. Again, depending on the direction of the pass, Watkins Hall, Bullard Hall, and the surrounding foliage play a role in the ability to receive transmissions from the overhead PropCubes.

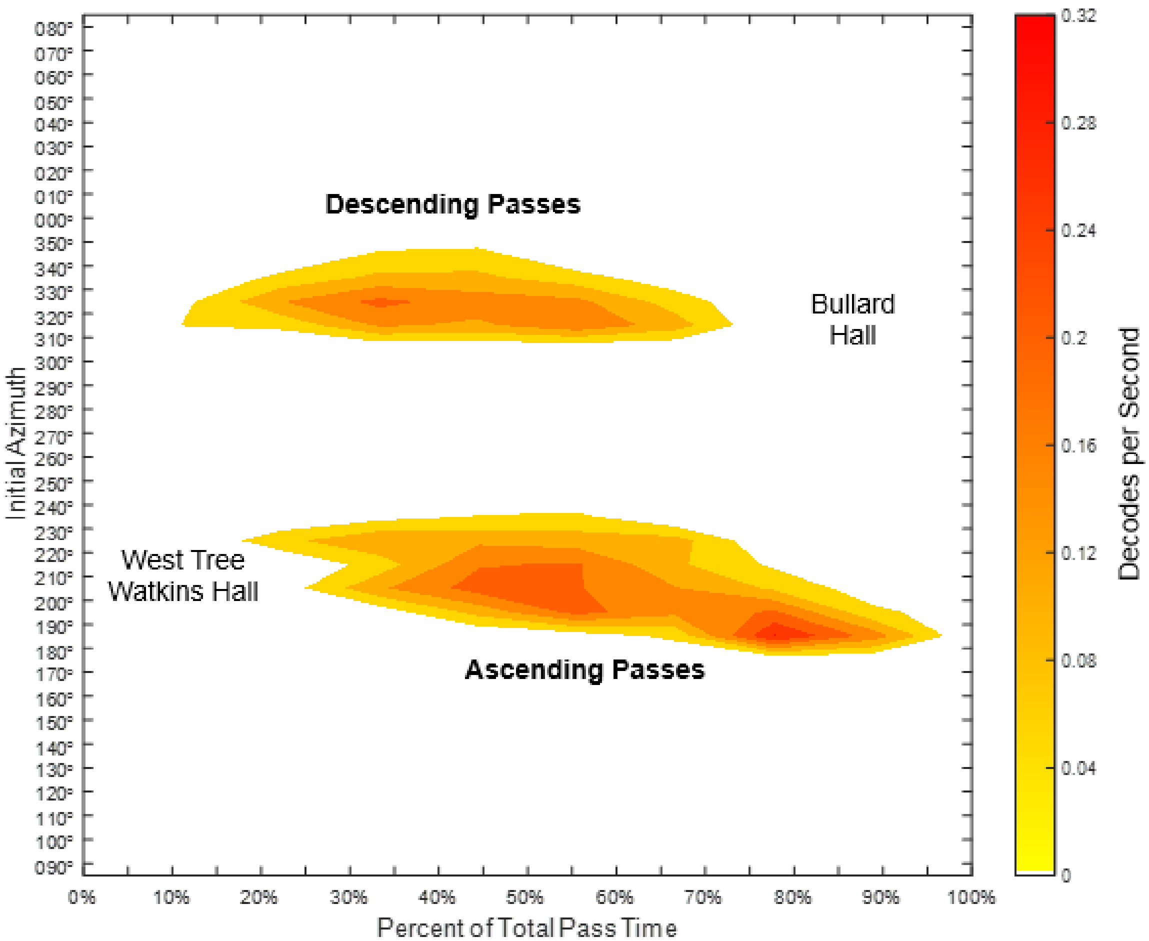


Figure 46. PTSUR-Flora Historical Downlink Performance Heat Map

The example shown in Figure 46 is the segmented pass analysis for PTSUR-Flora from March 2017 to April 2018. A transect of this data at a particular initial azimuth provides a metric for pass value based on likelihood of successful data downlink. The aggregation of whole pass analysis, TLE argument of perigee rotation effects, and segmented pass analysis provides a means to grade upcoming passes and optimize antenna scheduling in the case of conflicting passes, as explored in Chapter IV.

IV. PASS VALUE FUNCTION AND PREDICTIVE MODEL

Chapter III described the process to measure pass quality for a specific ground station-satellite pair via whole and segmented pass analysis. Additionally, it explained potential effects of argument of perigee rotation on satellite pass distance and associated signal losses. In order to optimize ground station network scheduling, this chapter utilizes the signal strength, whole pass, and segmented pass quality metrics to predict communication link performance of upcoming accesses.

Chapter IV not only provides a means to grade future passes independently and predict optimal communication times within a pass, but also delivers a method for one-to-one comparison of conflicting passes. Using historical performance, operators can now assess pass quality and reduce attempts at downlinking data when communication is unlikely or help pick between two passes for the more likely to produce data.

A. PASS VALUE FUNCTION

As a satellite travels from its initial to final azimuth, the range from the satellite to the ground station dictates the theoretical free space loss. For PTSUR-Flora, as shown in Chapter III, this thesis models the expected link performance as a function of pass initial azimuth and elapsed time via a signal loss heat map. The heat map in Figure 47 is made up from the calculated cross-sections of the signal loss values. As one moves along the y-axis of the figure, each cross-section can be considered a single pass at that particular initial azimuth. Moving from left to right along the cross-section signifies the elapsed time of that pass. Therefore, each cross-section represents the link performance as a function of initial azimuth and elapsed pass time for that particular ground station-satellite pair.

Figure 48 shows the extracted cross-section for a PTSUR-Flora pass occurring at 215° initial azimuth. By taking this heat map cross-section, and interpolating by a piecewise polynomial to provide a smooth function, operators can determine the signal quality metric for any discrete time interval from AOS to LOS. As expected, midway through the pass, the satellite reaches its smallest range at CPA, which minimizes free space signal loss.

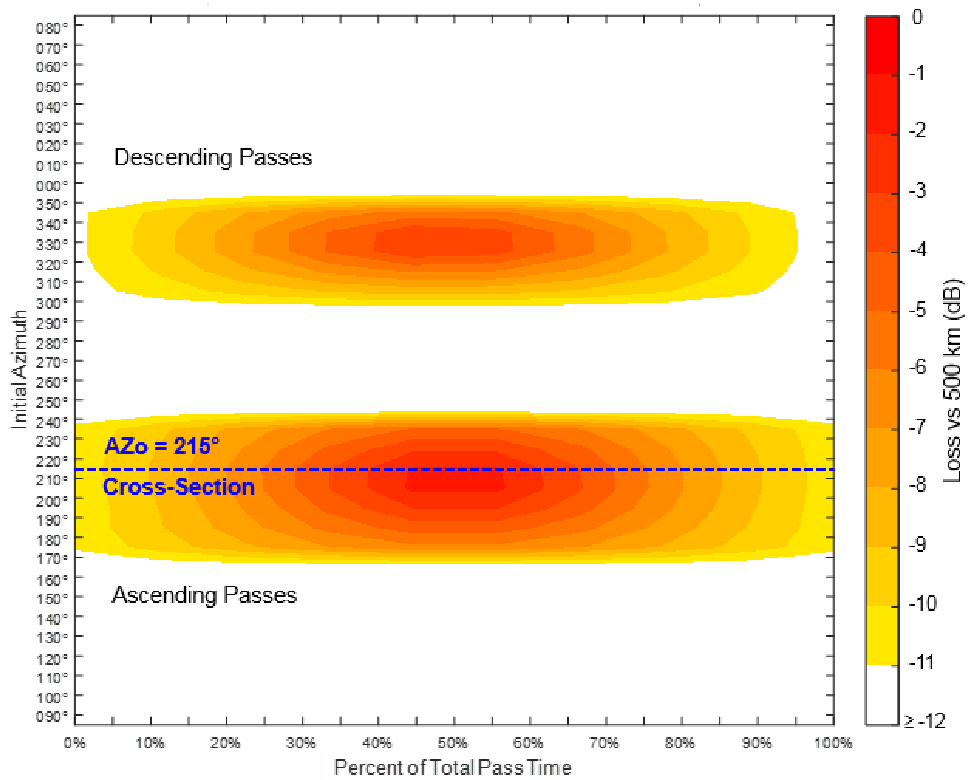


Figure 47. PTSUR-Flora Free Space Loss Heat Map (AZo = 215° ± 5°)

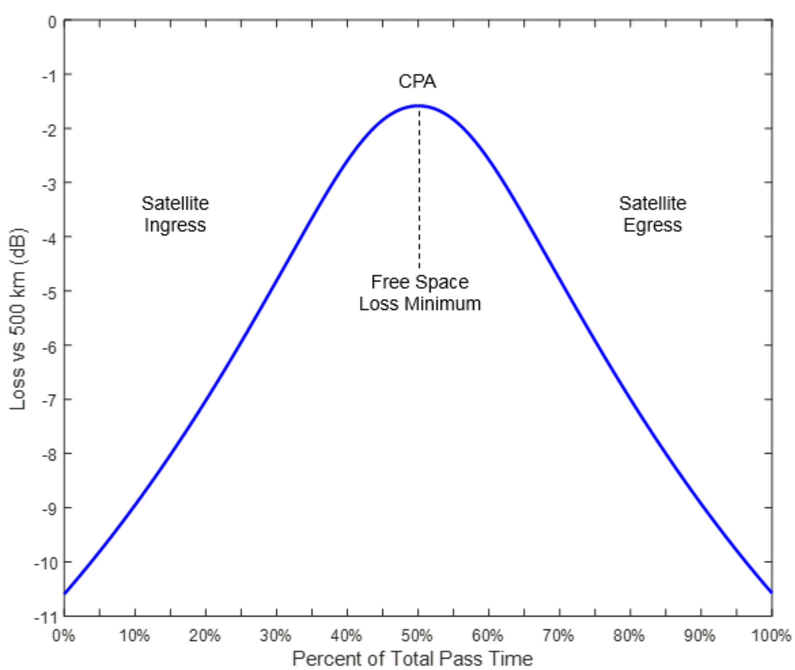


Figure 48. PTSUR-Flora Theoretical Value Function (AZo = 215° ± 5°) using Piecewise Polynomial Interpolation

Just as this thesis models communication likelihood at specific times based on range, Minelli [1] uses a similar approach by representing satellite location with respect to the ground station as an exponential (Gaussian) function. The size of the Gaussian, which represents the downlink signal strength from an overhead satellite, is scaled by a Benefit Value Function (BVF) [1]. In Minelli's work, the parameters σ_x and σ_y define the spread of the Gaussian whereas the scaling value (*V*) determines the height. The scaling value is directly related to the link margin, therefore, BVF height varies as the range from the satellite to ground station receiver changes throughout a pass, as shown in Figure 49.

$$BVF = Ve^{-\left(\frac{(x-x_n)^2}{2\sigma_x^{\,2}}+\frac{(y-y_n)^2}{2\sigma_y^{\,2}}\right)}$$

Figure 49. Benefit Value Function. Source: [1].

In Minelli's model, a brighter color Gaussian signifies a taller BVF and, therefore, a stronger downlink signal [1]. A representative descending pass with initial azimuth of approximately 315° is shown in Figure 50. The three Gaussians represent the downlink signal strength at three distinct times along the satellite path. As the ground station-satellite range lessens, the signal strength of the link increases, reaching a maximum value at the pass CPA. Following CPA, the BVF intensity decreases as the satellite travels away from the ground station to a final azimuth of approximately 170° at the minimum elevation of 10°.

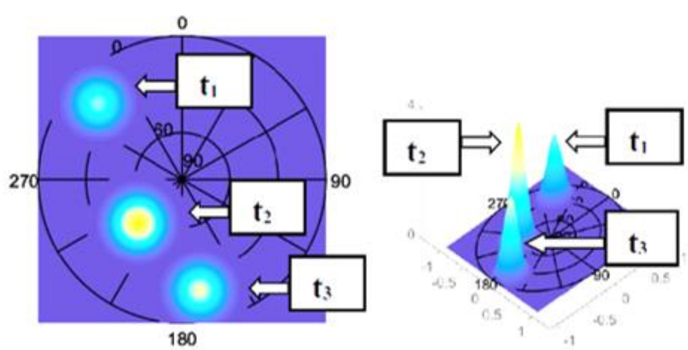


Figure 50. Single Satellite Pass Gaussians. Adapted from [1].

Earlier in this section, a heat map cross-section at a specific initial azimuth represented the signal loss values at discrete times throughout the pass. To liken the results to Minelli's pass Gaussians, this thesis repeats this method to predict the theoretical free space loss for a descending pass at 315° initial azimuth, as shown in Figure 51. Since MATLAB calculates the signal loss throughout the pass using the satellite-to-ground station range, the values are proportional to the scaling value from Minelli's model. As such, the thesis's model has the same relation to range as Minelli's where the height of the Gaussians at each of the three times are proportional to the signal loss values.

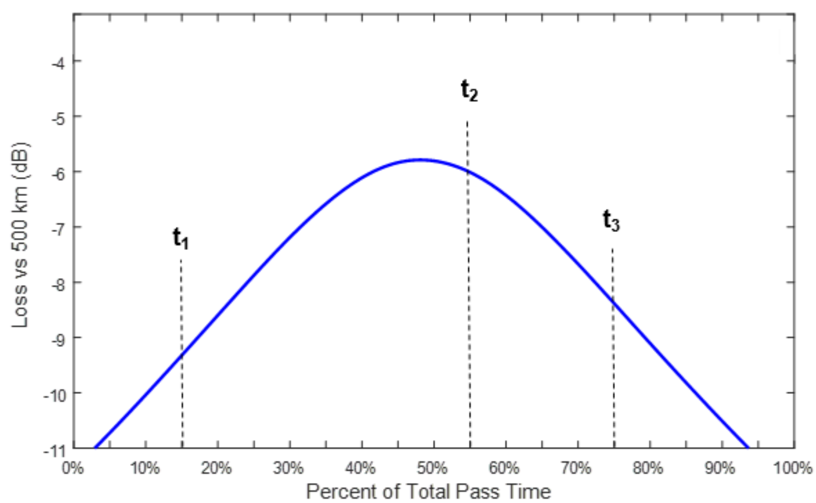


Figure 51. PTSUR-Flora Approximate Gaussian Times (AZo = 315° ± 5°)

Minelli recognizes the shortcomings of the current model for ground station optimization since it does not consider "spacecraft orientation, varying data rates, environmental noise, etc." [1]. However, creating a value function based on historical pass performance accounts for real-world factors that affect downlink success. In the same manner as the theoretical signal loss value function, a cross-section at a selected initial azimuth of the historical data provides expected communication performance during the pass. Each value function represents the performance of similar AZo passes throughout the access duration. Again, MATLAB interpolates the values with a piecewise polynomial for a smooth signal quality metric. The historical performance value function as compared to the theoretical signal quality based on free space loss is shown in Figures 52 and 53.

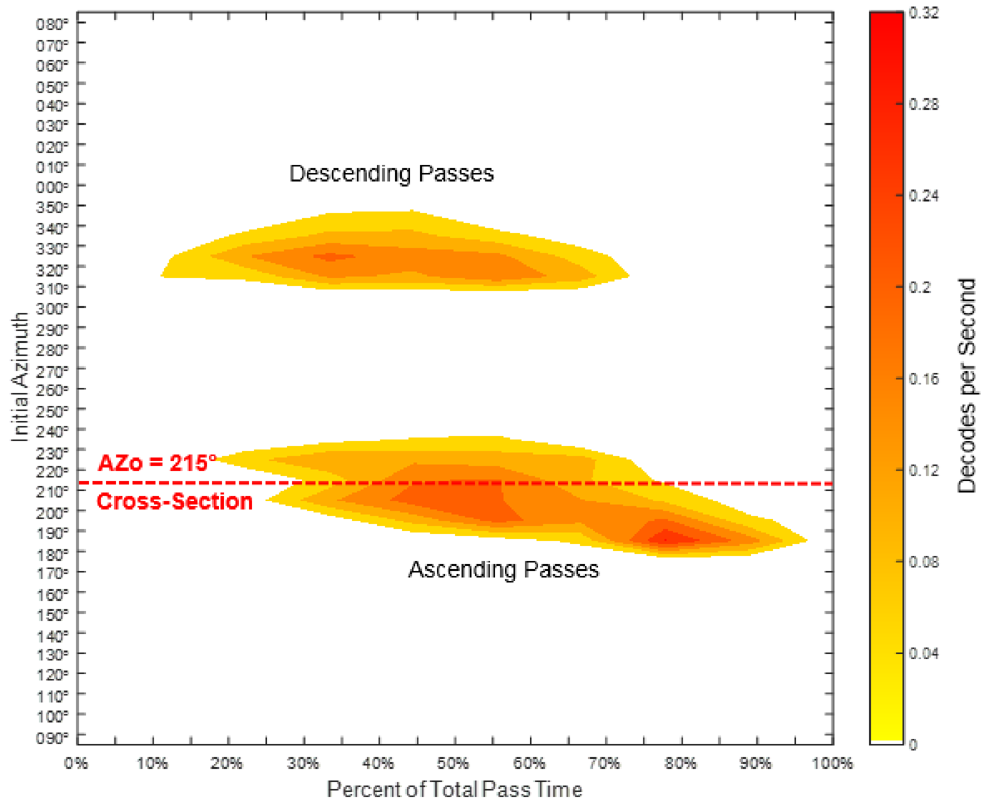


Figure 52. PTSUR-Flora Historical Performance (AZo = 215° ± 5°)

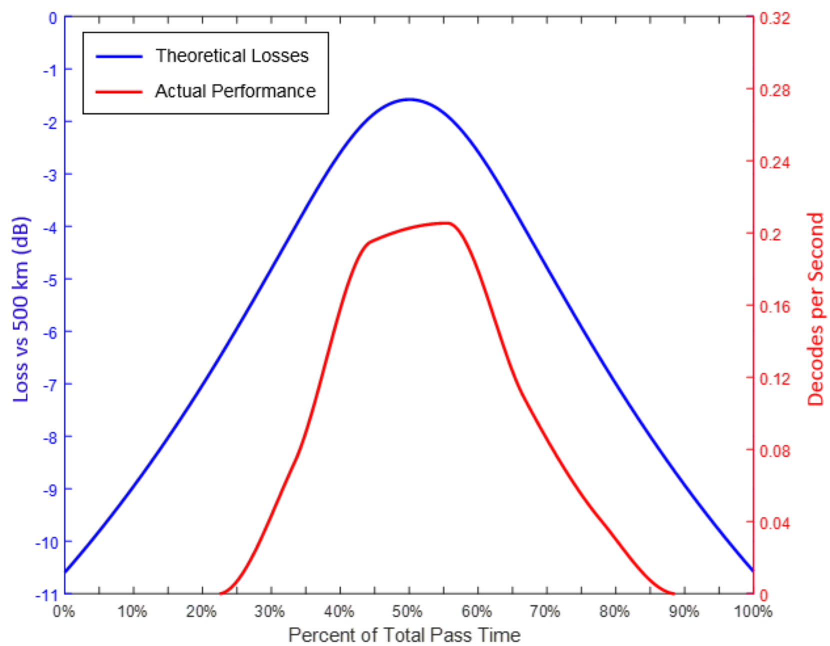


Figure 53. PTSUR-Flora Theoretical vs Historical Performance Value Functions (AZo = 215° ± 5°)

For the initial azimuth of 215° in Figure 53, and consistent in both the theoretical loss and historical performance curves, the best signal quality occurs mid-pass, as the satellite reaches its maximum elevation of 70° and free space loss is minimized. There is a substantial difference, however, at the start and end of the actual performance value function. The first opportunity for decodes occurs more than 20% into the pass, and received transmissions fall off upon reaching the final fifth of the access. For a typical pass duration at this initial azimuth, these combined gaps consist of nearly three minutes with low likelihood of successful communication. Referring back to the PTSUR-Flora segmented pass analysis, the adjacent west tree and Watkins Hall obstruct line-of-sight at the beginning of an ascending pass, such as one with initial azimuth of 215°. Similarly, the magnolia tree on the northeast side of the courtyard obstructs the egressing portion. Operators can expect minimal data downlink when these physical obstacles impede the satellite-to-ground station line-of-sight.

This thesis performs another heat map cross-section analysis for an initial azimuth of 185°, as shown in Figure 54. This group corresponds to lower maximum elevation passes, longer link ranges, and, therefore, a theoretical curve that depicts greater overall losses.

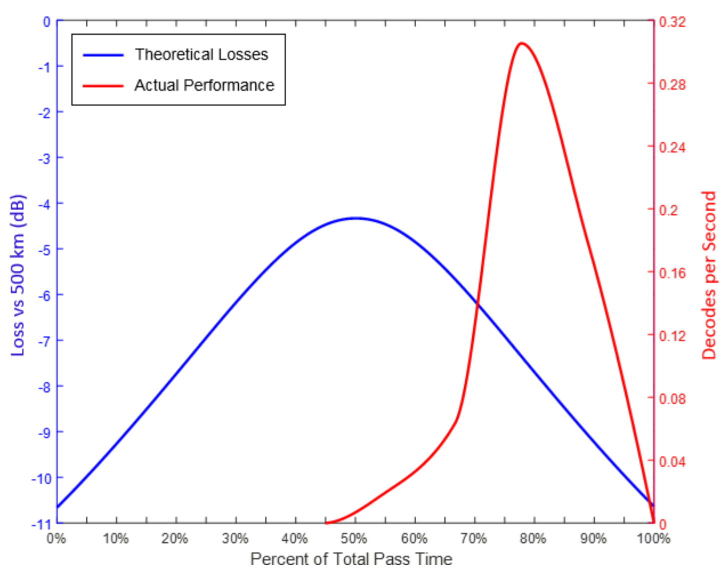


Figure 54. PTSUR-Flora Theoretical vs Historical Performance Value Functions (AZo = 185° ± 5°)

The historical performance curve in Figure 54 differs significantly from that of the theoretical results. Again, physical obstacles surrounding the PTSUR antenna impede signals during specific PropCube pass segments. The pass originates in the south where Bullard Hall obstructs incoming signals. Due to the proximity of the building to the antenna, lower track elevation angles, and satellite position to the southeast of the ground station for the first half of the pass, there are no historical received transmissions in these segments. At approximately the midpoint of the pass, the satellite track clears the north corner of Bullard Hall and communication begins. For the remainder of the pass, the ground station maintains line-of-sight for the majority of the egressing portion as the satellite track remains clear of obstructions between Bullard Hall and the northeast magnolia tree.

For ascending PTSUR passes, as shown in the previous example, the ground station collects more decodes as Flora tracks away from the antenna during satellite egress. In contrast, the majority of descending pass decodes occur as the PropCube moves inbound to the ground station. Figure 55 shows the pass value function for an initial azimuth of approximately 325°, a descending Flora pass. The beginning of the access is characterized by a clear line-of-sight above Halligan Hall whereas the egress is obstructed by Bullard Hall to the southeast.

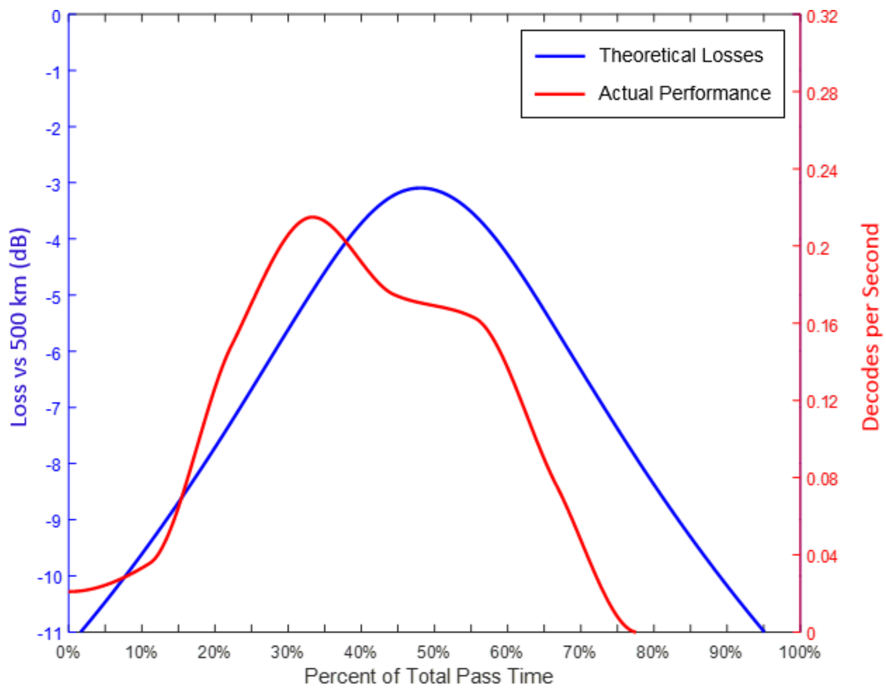


Figure 55. PTSUR-Flora Theoretical vs Historical Performance Value Functions (AZo = 325° ± 5°)

B. CONFLICTING PASS COMPARISON

The purpose of the pass quality value function is to predict the performance of upcoming passes and assess the likelihood of successful communication. Ground station operators can then use these predictions to aid in antenna scheduling and maximize data transfer in response to the "many satellites, few ground stations" problem. If a ground station simultaneously has two satellites within line-of-sight, with all other considerations equal, operators should attempt communication with the one that gives the best opportunity to transmit and receive data from the ground station.

In the previous section, this thesis begins pass grading with theoretical signal strength based on free space loss. MATLAB uses the same calculations to initially compare two generic satellite passes with conflicting ground station access times, as shown in Figure 56. The analysis period begins at the ascending pass AOS of the first satellite and ends at the descending pass LOS of the second satellite. Sat_2 AOS occurs exactly halfway through the pass duration of Sat_1, thus creating a scheduling conflict where antenna priority must be given to one satellite at the expense of the other.

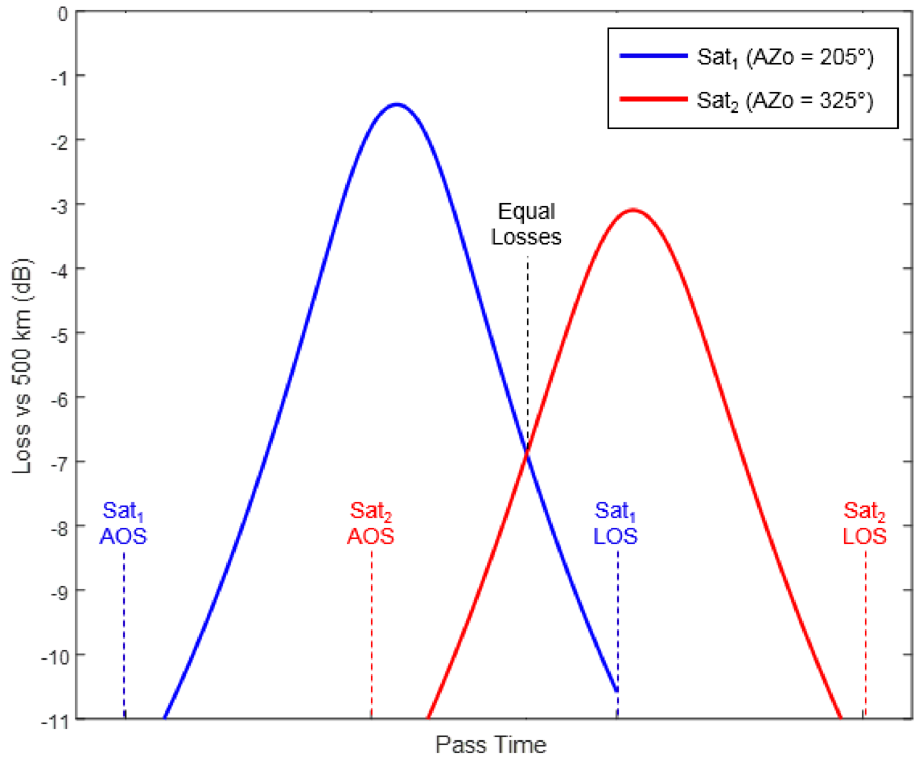


Figure 56. Conflicting Pass Theoretical Value Functions

As explored in the previous section, the difference in signal loss magnitudes between these two passes is due to differing elevation angles throughout each access. Although both are relatively high maximum elevation angles passes, Sat_1 experiences decreased overall losses at 80° maximum elevation angle whereas Sat_2's elevation peaks at 70°. If the passes were to reach CPA simultaneously, Sat_2's expected signal strength would not surpass Sat_1's at any time during the pass duration. However, due to the time offset from the first pass to the second, Figure 56 shows a distinct cross-over point when the two accesses have equal losses based on their respective satellite-to-ground station ranges. If using the expected signal strength as the only grading criteria, the operator would transition antenna priority from the first to the second satellite at the value function intersection.

Minelli's optimization research comes to a similar conclusion when analyzing two conflicting passes based on their signal strength Gaussians [1]. Figure 57 shows the tracks of two satellites, both performing descending passes over a ground station. The signal strength Gaussians for these two satellites are shown at four distinct times based on their range to the ground station and associated dB cost. Again, a brighter color Gaussian represents a taller BVF and stronger downlink signal.

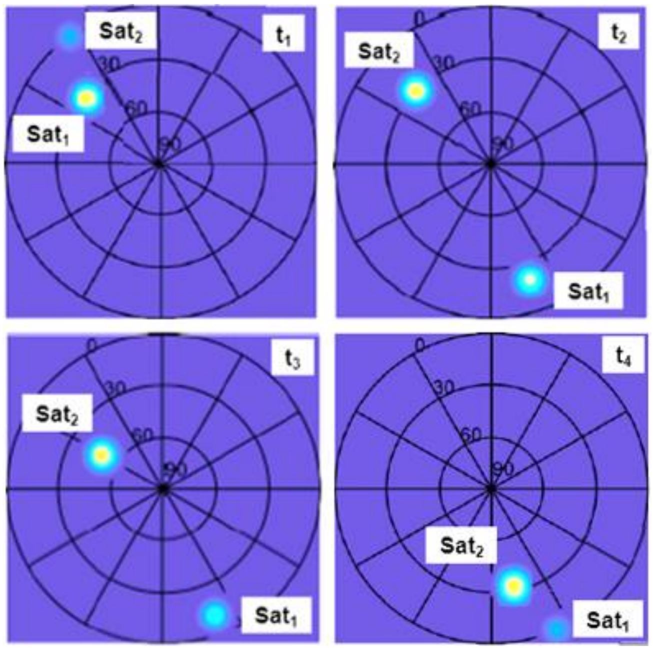


Figure 57. Conflicting Pass Satellite Gaussians. Adapted from [1].

The second satellite's AOS is offset from the first. Therefore, as Sat_1 ingresses towards its pass CPA, it receives antenna priority based on greater overall signal strength. After it reaches its CPA, Sat_1 begins the egressing portion of its track where Sat_2 then becomes the priority as its signal strength surpasses Sat_1. Throughout the remainder of the scenario, Sat_2 maintains the greater signal dB and associated Gaussian height based on free space loss and link margin. Understanding that this analysis does not consider antenna gain patterns and the time taken to transition the antenna from one satellite track to the other, Minelli's assessment depicts a similar crossover point when operators should shift priority to maximize data downlink, as shown in Figure 58 [1].

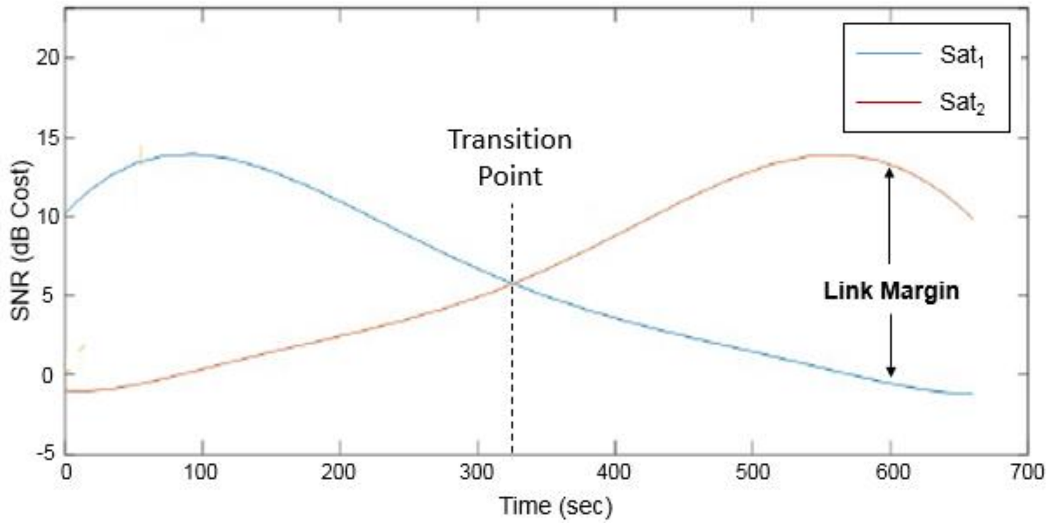


Figure 58. Link Margin for Conflicting Pass Satellites. Adapted from [1].

As explored earlier in this chapter, the theoretical signal loss curves do not consider the effects of terrain, manmade obstacles, or local noise environments in the areas surrounding MC3 antennas. As such, historical link performance provides a more accurate representation to predict signal quality and optimize data transfer. Figures 59 and 60 model the pass value relationship between conflicting accesses of the PropCubes Flora and Merryweather. The scenario begins at Flora's AOS, when the satellite commences a pass with an initial azimuth of 225° overhead PTSUR. After Flora completes 40% of its pass, Merryweather begins a descending access with initial azimuth of 315°.

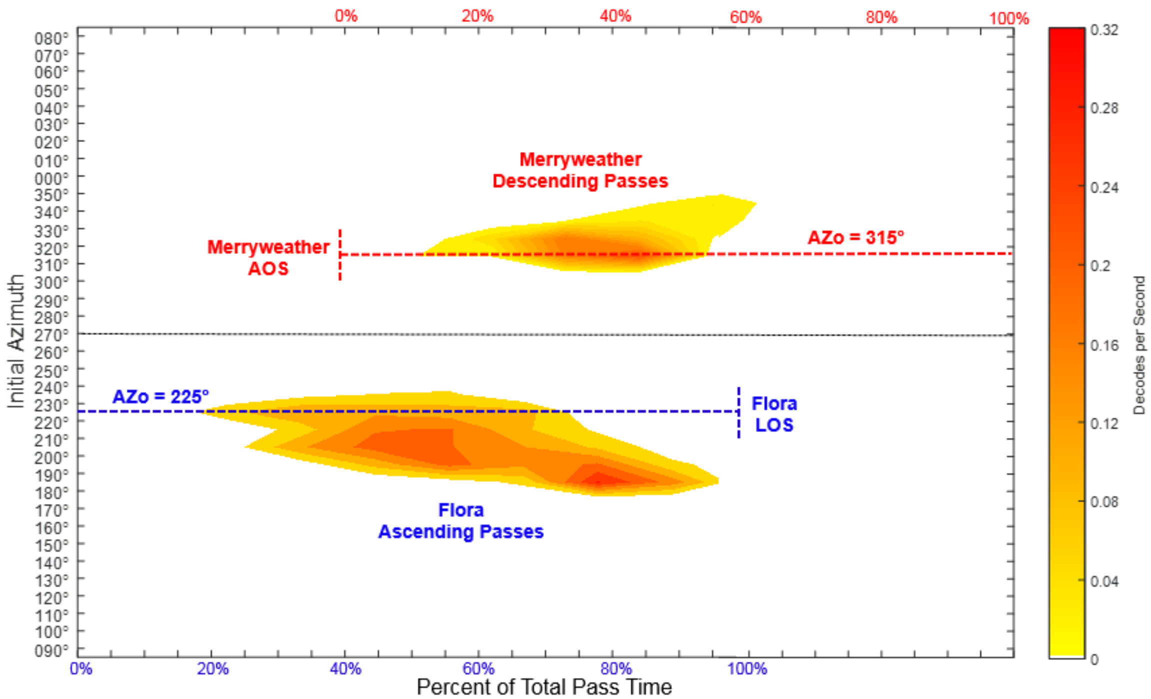


Figure 59. PTSUR-Flora/Merryweather Conflicting Historical Performance

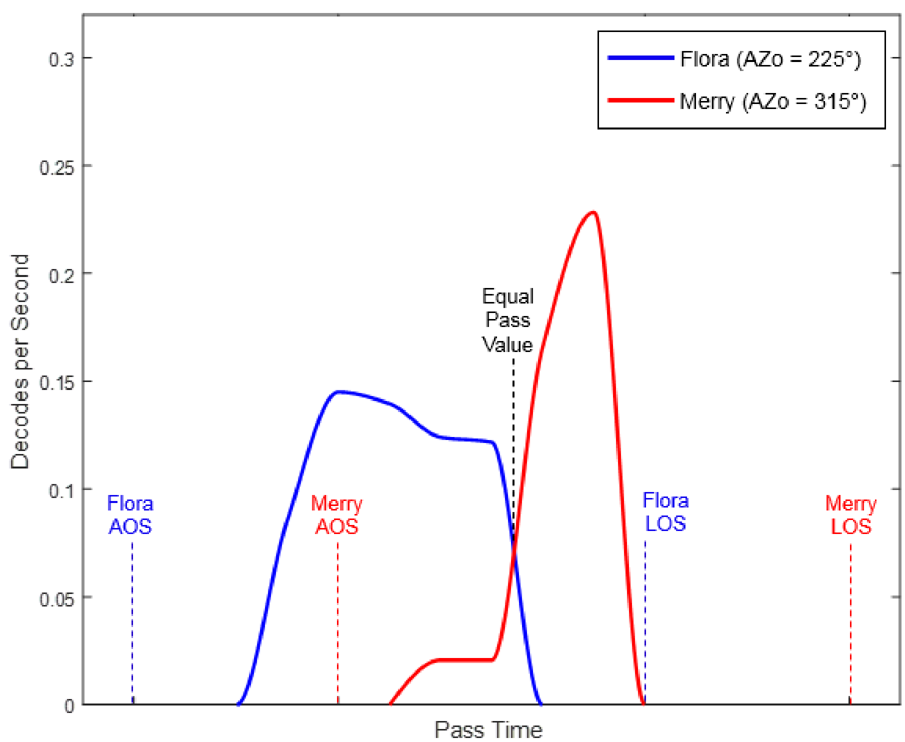


Figure 60. PTSUR-Flora/Merryweather Conflicting Historical Value Functions

The historical performance cross-sections for each satellite, along with the pass start time offset, provide a forecast of data transfer probability as a function of elapsed time. As Flora ingresses towards the ground station during the early portions of its pass, it is the only satellite in view and should receive antenna priority. Flora's performance reaches a maximum value at the approximate time of Merryweather's AOS. However, even though these passes now conflict, there is no necessity to shift priority due to Merryweather's historically poor performance during early pass segments. As the scenario progresses, Flora's pass quality drops, and Merryweather's performance increases as it clears physical obstacles surrounding PTSUR. At the pass value intersection where the historical performance of both satellites is equal, antenna priority should shift to Merryweather for the remainder of its pass to optimize network decode rate.

A one-to-one conflicting pass comparison for Flora and Fauna over the PTSUR ground station is shown in Figure 61. In this scenario, Flora is executing a descending pass with initial azimuth of 315°. Simultaneously, Fauna begins an ascending pass with initial azimuth of 225°. Due to concurrent AOS times, physical obstacles and pass geometry become significant factors in assessing antenna prioritization. Flora receives initial priority as it achieves line-of-sight first after clearing the roof of Halligan Hall during ingress. As Flora reaches the segment of its pass where it performs best, however, Fauna's historical downlink rate increases rapidly to its peak. If the pass value function is the sole metric driving antenna steering, the intersection approximately halfway into the scenario is the optimal antenna transition point. Based on historical performance and the simultaneous nature of these passes, the operator may choose to forgo the best performing portion of Flora's pass in favor of a more beneficial Fauna downlink, as shown in Figure 61.

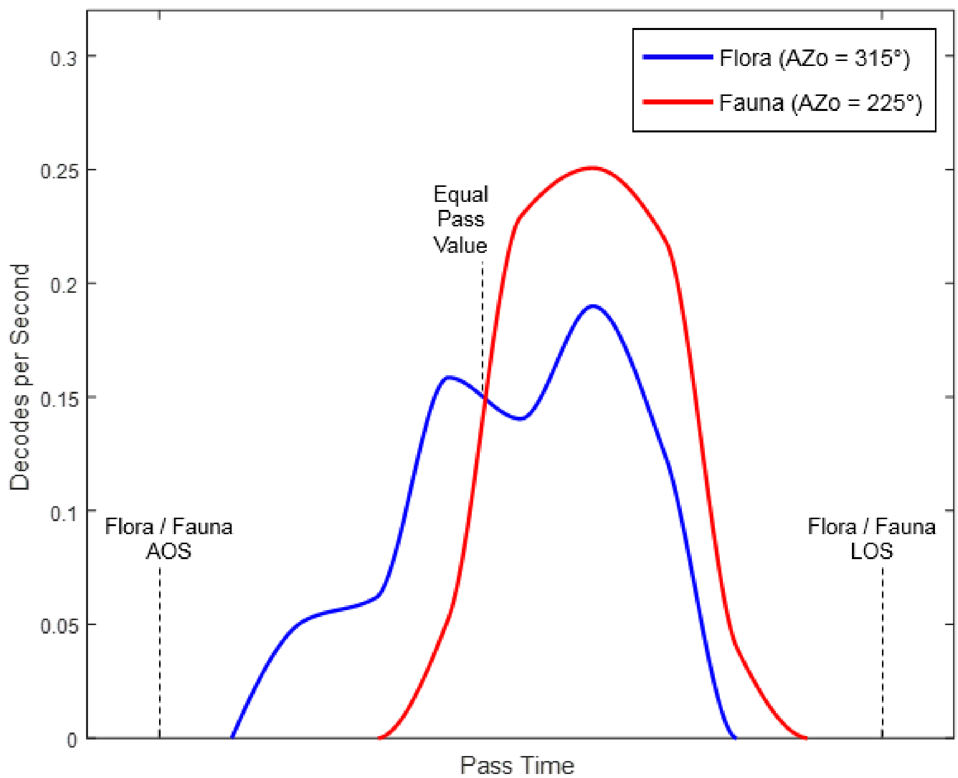


Figure 61. PTSUR-Flora/Fauna Conflicting Pass Historical Value Functions

C. SATELLITE PASS OPTIMIZATION

Researchers can use this thesis's pass quality metric with CubeSat network optimization programs to drive antenna schedule prioritization. By referencing the initial azimuth of an upcoming pass and the historical satellite performance at that azimuth, operators can determine pass value as a function of elapsed access time and predict the likelihood of successful data downlink. In the event of a conflicting pass, where satellites are competing for limited antenna assets, operators can perform a one-to-one pass benefit comparison to efficiently schedule antenna assets.

Minelli's MC3 Network optimization program, which he is currently developing in the NPS Small Satellite Laboratory, relies on scaled Gaussians defined by their respective BVF [1], [10]. For each pass, the optimizer should be configured to use the value function polynomial approximation from this thesis as a multiplier for the height scaling metric *V*. As satellites transit access regions with greater historical data rates, the Gaussian height will increase accordingly showing the pass segments with greater downlink benefit. The program will no longer grade passes only based on their link range and resulting signal strength, but also evaluate their likelihood of success via the MC3 data set. These improved Gaussians will now include the complex, real-world effects of satellite orientation, physical obstacles, and local noise environments.

As explored in Chapter III, the relationship between PropCube argument of perigee location and downlink performance was not statistically significant to establish a correlation for pass grading. However, with a more robust data set, the effect of a changing average altitude throughout the life of a satellite provides an additional measure of overall pass quality. Much like the scaling value *V* multiplier can adjust the Gaussian height, the location of perigee in the orbit with respect to the ground station latitude can scale the spread of the Gaussian, defined by the parameters σ_x and σ_y. By spreading the Gaussian via this metric, the optimizing code will account for the temporal changes in argument of perigee and give more value to passes nearer the ground station.

Chapter V provides a more thorough analysis of future work and considerations to further optimize the MC3 Network, or any other similarly configured CubeSat ground stations, to take full advantage of limited antenna assets.

THIS PAGE INTENTIONALLY LEFT BLANK

V. FUTURE WORK AND CONCLUSION

Chapter IV developed a measure of MC3 PropCube pass quality as a function of initial azimuth and elapsed time. Additionally, it utilized ground station-satellite value functions to predict communication link performance and perform one-to-one comparisons of simultaneous conflicting passes. Optimizing programs, such as the one currently being developed for the MC3 Network [10], can utilize these metrics to better forecast downlink likelihood and efficiently schedule antenna assets.

This chapter summarizes the results of this thesis and synthesizes the importance of optimizing networks in response to the growing "many satellites, few ground stations" problem. It provides recommendations for future work concerning the MC3 Network data set and optimization program. It also presents additional factors, not fully explored in this thesis, that affect pass quality and satellite-to-ground station communication.

A. SUMMARY

Due to CubeSat proliferation and the requirement for successful data downlink, the number of conflicting passes within ground station networks, such as MC3, will continue to increase [1]–[5]. To strategically plan for potential ground station saturation, operators should give antenna priority to the satellite that has the best chance of successful data downlink based on the historical performance of similar geometry passes [1], [2], [4], [5].

This thesis, using the MC3 PropCube data set as an example, not only provides a means to analyze the downlink likelihood of upcoming passes as a whole, but also evaluates the quality of individual pass segments. The current MC3 SOC software prioritizes the first satellite in view regardless of its opportunity for success and remains tracking that satellite for the entirety of its pass. An optimizing program, currently in development in the NPS Small Satellite Laboratory, prioritizes satellite passes based on expected signal strength calculated from satellite-to-ground station distance [10]. However, the analysis performed in this thesis provides pass value considering real-world factors that affect communication such as physical obstacles and directional noise environments.

Based on the initial azimuth of an upcoming pass, the results of analysis provide a value function that predicts the decode rate during each pass segment. In the event of conflicting passes, this thesis facilitates value function comparison in order to assess the pass that provides the most benefit in each time segment. Additionally, using the pass quality as a function of time, operators and their network optimization programs could use the metric to determine a transition point where antenna priority should be shifted during a pass based on higher likelihood of data transfer. If implemented into an optimization program, this thesis's pass value function can be used to grade each access and appropriately schedule antenna assets to maximize the overall network data rate.

B. FUTURE WORK

The pass quality results presented in this thesis improve upon the current pass grading method by evaluating ground station-satellite pairs not based solely on link distance, but also accounting for the combined effects of physical obstacles and local noise environments. More work could be done, however, to consider other differentiating factors within the data set that may influence successful communication. A more detailed analysis of these variables could improve upon the pass quality metric and provide a more accurate performance prediction for the MC3 Network and antenna scheduling tools.

1. Argument of Perigee Temporal Effects

As explored in Chapters II and III, argument of perigee location changes throughout the orbital lifetime of a satellite due to J_2 perturbation effects [11]. As perigee rotates, the MC3 CubeSat pass duration and average range with respect to the ground station vary, which, in turn, affects link performance. Depending on perigee location with respect to ground station latitude at the time of an access, one pass type, ascending or descending, may be more valuable than the other based on lower average altitude and decreased link distance.

In Chapter III, this thesis calculated that a full 360° perigee rotation for the PropCube, Flora, takes approximately three years to complete. With just over a year and a half of pass information contained in the current data set, the correlation between link performance and temporal changes in average range is not statistically significant.

Therefore, to better assess the value of decreased link distance and incorporate it into the pass value function, future studies could utilize a more robust data set including satellites that have progressed through a full perigee rotation cycle. Using the methods outlined in this thesis, perigee rotation effects could provide an additional measure of access quality to be implemented into the pass value function and optimizing program Gaussians.

2. Local Noise Environments

By performing segmented pass analysis on various ground station-satellite pairs, where performance was assessed based on the location of preambles and decodes within the individual passes, this thesis explored the effects of local noise environments and their potential impact on successful data downlink. As an example, the preamble and decode location plot for UNM-Flora shows less successful downlink events to the north of the ground station in the direction of the RF dense city of Albuquerque [17]. However, this analysis does not differentiate daytime passes from those at night, when noise environments may be less influential. In general, the amount of RF noise produced by an urban environment becomes less dense as nighttime approaches and remains low throughout the evening hours. Therefore, to better distinguish the quality of individual passes, future work could analyze the diurnal changes within the local noise environments.

3. PropCube Orientation

Permanent magnets equipped on each MC3 PropCube act as a rudimentary attitude control system. Throughout their orbits, these magnets align the satellites with the Earth's magnetic field, which results in a predictable orientation and prevents spacecraft tumbling. However, this orientation may not result in optimal communication attitude for any particular ground station. More advanced CubeSats, to include future additions to the MC3 Network, will utilize star trackers and horizon sensors as well as active pointing systems to improve communication. Using these systems, future researchers can determine the orientation of satellite transmit antennas during a MC3 Network pass. Off-nadir pointing may be the cause of varying signal strength and inconsistent downlinks. By correlating antenna direction with instances of successful preambles and decodes throughout a pass, researchers can assess the likelihood of data downlink based on a satellites expected orientation as it passes over the ground station.

C. CONCLUSION

As of May 2018, the MC3 Network monitors and controls three CubeSats using a system of seven ground stations. This network can sufficiently manage the existing number of CubeSats, as pass conflicts rarely occur due to the relatively small number of downlinking satellites. However, no earlier than (NET) the end of 2018, the MC3 Network is projected to service an additional six satellites, shown in Table 9. Other networks are experiencing similar growth in CubeSat constellation size that will create frequent conflicting passes and inevitable ground station saturation.

Table 9. Projected CubeSat Additions to MC3 Network

CubeSat Name	Estimated Launch Date (NET)
GOERGEN SHFT	May 2018
RSat	July 2018
NPSAT1	October 2018
FalconSat-7	October 2018
Anna (Polar Scout)	June - October 2018
Elsa (Polar Scout)	June - October 2018

As CubeSat networks continue to grow, the ground stations that service their downlink capabilities must become more robust. In order to strategically plan before reaching downlink saturation, operators and software that schedule antenna assets should prioritize satellites based on their likelihood of link success and consider factors such as satellite-to-ground station distance, physical obstacles, and local noise environments. Integrating historical performance of ground station-satellite pairs into schedule optimization programs will more accurately prioritize the satellite that has the best chance to communicate with the ground station and maximize network performance.

APPENDIX A. PASS DATA SET TO EXCEL CSV FILE

From the MC3 Access Data Homepage, select the SQL tab to navigate to the MC3 SQL Query Page, as shown in Figures 62 and 63.

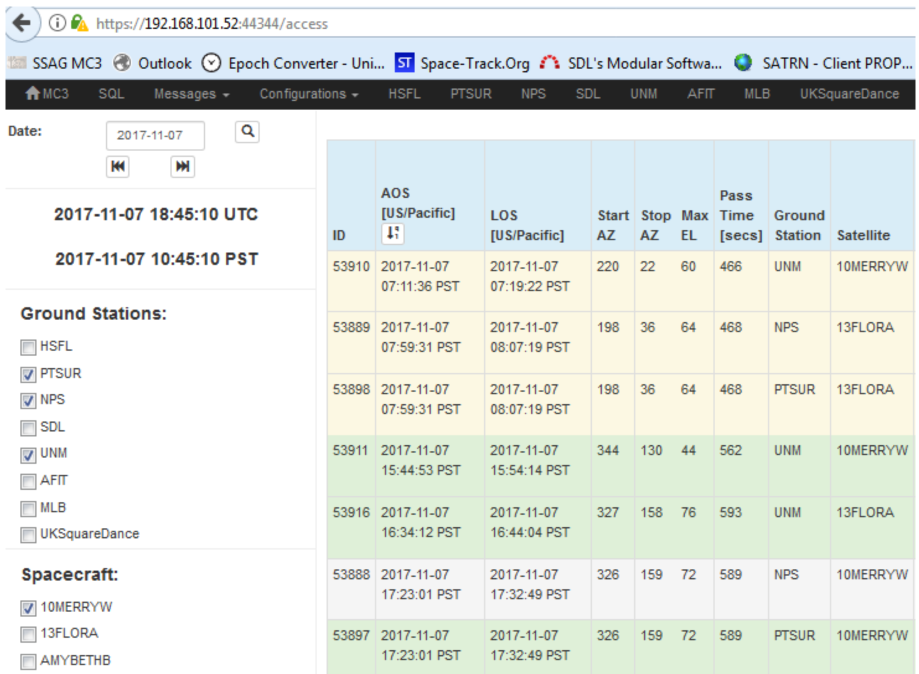


Figure 62. MC3 SOC Data Homepage https://192.168.101.52:44344/access

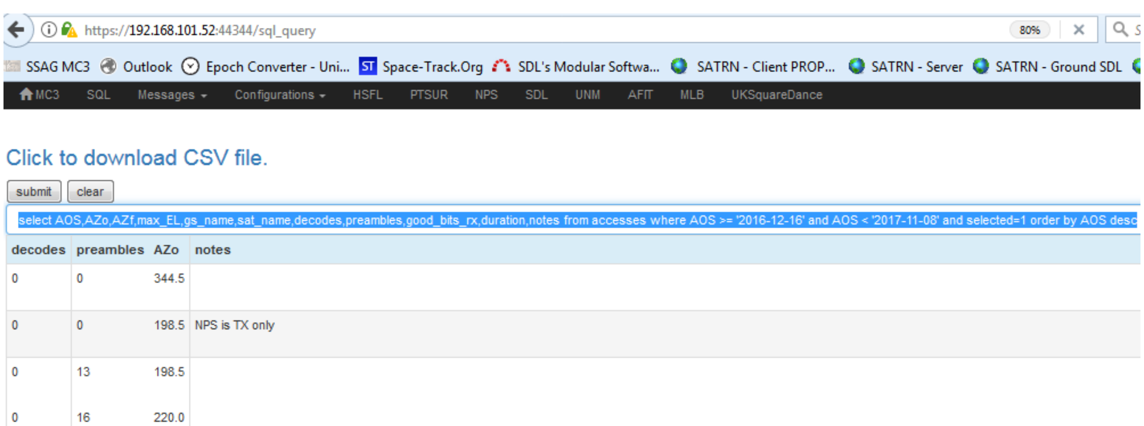


Figure 63. MC3 SOC SQL Query Page https://192.168.101.52:44344/sql_query

Enter the following query into the command line:

Select AOS,AZo,AZf,max_EL,gs_name,sat_name,decodes,preambles,good_bits_rx,duration, notes from accesses where AOS >= '2016-12-16'and selected=1 order by AOS desc

As needed, modify the command line input to export specific data.

Define column headers for CSV:

select AOS,AZo,AZf,max_EL,gs_name,sat_name, decodes,preambles,good_bits_rx,duration,notes

Define all saved accesses from MC3 database:

from accesses

Define starting date:

where AOS >= '2016-12-16'

Include enabled passes only:

and selected=1

Sort CSV by AOS Date/Time in descending order:

order by AOS desc

After the query is submitted, select "Click to Download CSV File." Save the file to the desktop as the default location. The CSV will be saved to the desktop with a filename format resembling *sql_1515176417.568000*. If the CSV is not saved to the desktop, all saved CSVs are archived into the directory below. Locate the file based on the saved filename and disregard the creation date as it may be incorrect. Rename the file GSdataCSV to be compatible with the MATLAB scripts in this thesis.

\\192.168.101.67 (Y:)\MC3\Software\web\csv

APPENDIX B. TLE EXPORT AND COMPILE USING MATLAB

Access the Space-Track website, shown in Figure 64, at https://www.space-track.org/auth/login and login with the following credentials:

Username: mc3ops@gmail.com

Password: ---Redacted for MC3 OPSEC---

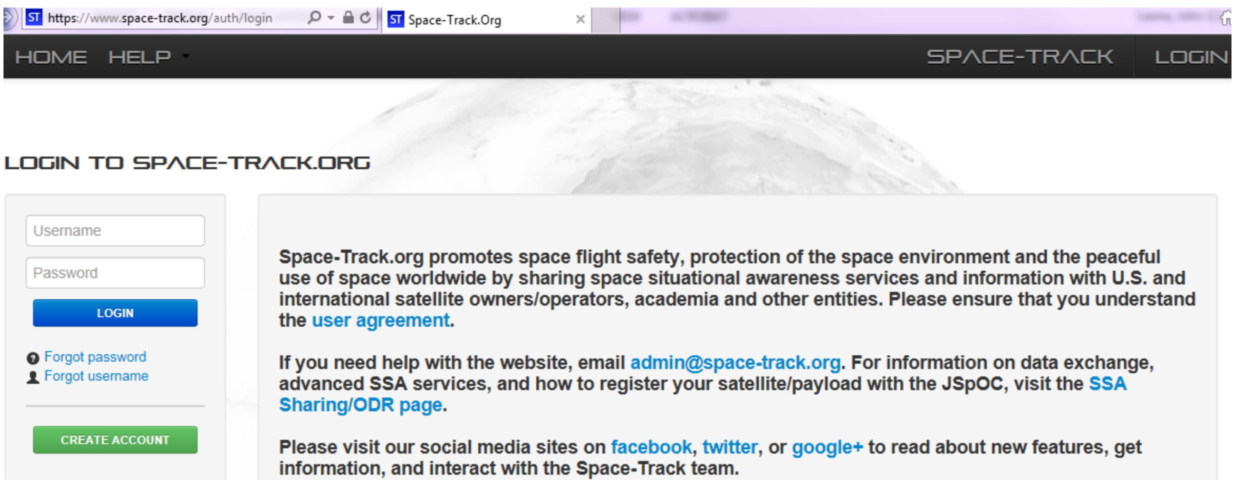


Figure 64. Space-Track.org Homepage

Select “Files >> Download” at the top of the screen to obtain historical TLE for Flora and Merryweather. Expand the directory labelled “L55” and download the desired TLE files to the TLE directory on the computer.

In order to obtain historical TLE for Fauna, select “Retrieve TLE Data by Satellite Catalog Number” under the “Two Line Element (TLE) Data” heading from the Space-Track.org main menu. As shown in Figure 65, enter Fauna’s Catalog Number (43052) and set any desired constraints. Date range starting at 2017–12-01 and ending on the current date obtains all historical Fauna TLE. Copy and paste TLE information for Fauna into the 1RawFauna .txt file in the TLE directory on the computer.

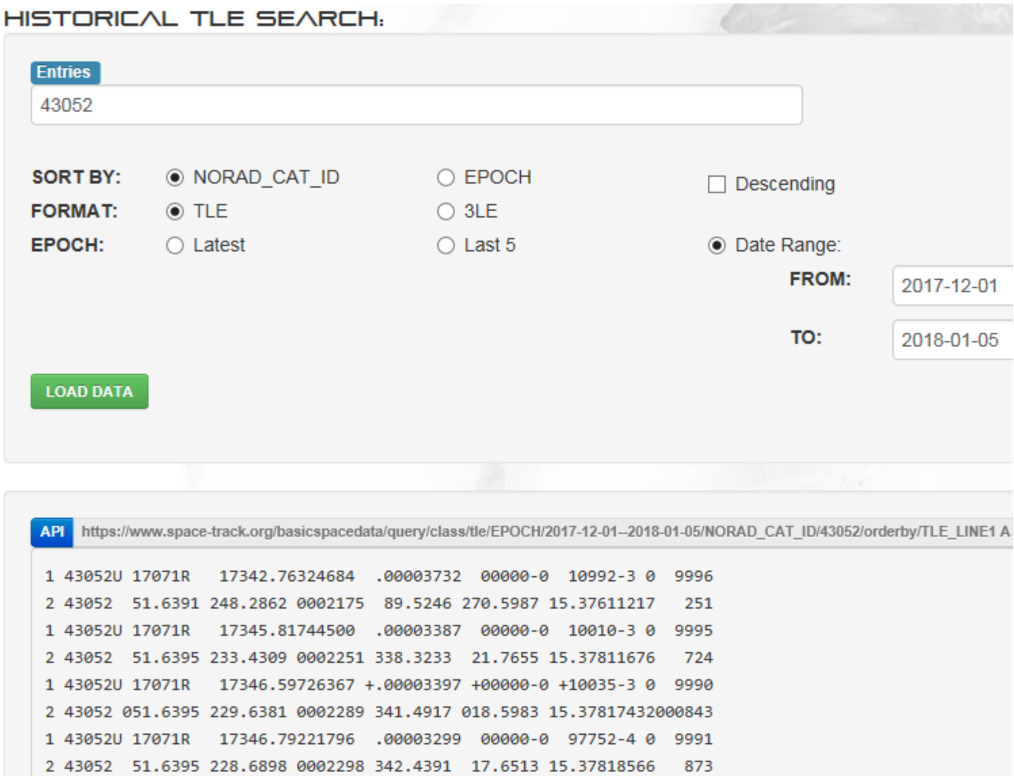


Figure 65. Space-Track.org TLE Search

After the TLE files for Flora/Merryweather and TLE text information for Fauna is located in the TLE folder on the computer, run the TLE MATLAB script, found in Appendix D, to extract TLE for each satellite into its own TLE file. Once the script is complete, the TLE directory will have files labelled FloraTLE, MerryTLE, and FaunaTLE for use in STK and MATLAB.

The TLE files from Space-Track.org may download without a file extension. Prior to running the TLE compile script, ensure that all downloaded TLE have the .txt extension. Renaming them one-by-one is an option, but for a large number of files, the following steps conduct a batch rename extension of the files.

- Find the FilesTLE folder

- Hold Shift + Right-Click the folder. Click “Open Command Window Here”

- Enter the following command. It will rename any file without an extension to the .txt extension. Note the two spaces within the command. `ren *. *.txt`

APPENDIX C. WHOLE PASS ANALYSIS MATLAB SCRIPT

```
% ===== APPENDIX C. WHOLE PASS ANALYSIS =====

% Portions of script generated from MATLAB Data Import Script Generator

% ===== 1. IMPORT GROUND STATION DATA FROM EXCEL CSV =====
% Prereqs: CSV Export in Appendix A

%%
% Import Ground Station data from Excel CSV into MATLAB table for
% follow-on analysis.  CSV must be organized in the following columns:
%      decodes, preambles, AZo, notes, gsname, AZf,
%      duration, AOS, satname, gootbitsrx, maxEL
clear all; close all; clc
% Initialize variables for filename, start row, and format string
% Extract Pass Data from CSV file
filename = 'GSdataCSV.csv';
delimiter = ',';
startRow = 2;
formatSpec = '%q%q%q%q%q%q%q%q%q%q%q%[^\n\r]';
% Open CSV file, read columns of data by format string, close file
fileID = fopen(filename,'r');
textscan(fileID, '%[^\n\r]', startRow-1, 'WhiteSpace', '',...
    'ReturnOnError',false, 'EndOfLine', '\r\n');
dataArray = textscan(fileID, formatSpec, 'Delimiter', delimiter,...
    'TextType','string', 'ReturnOnError', false);
fclose(fileID);
% Converting numeric text to numbers, date formats to MATLAB date/time
% Replaces non-numeric text with NaN
raw = repmat({''},length(dataArray{1}),length(dataArray)-1);
for col=1:length(dataArray)-1
    raw(1:length(dataArray{col}),col) = mat2cell(dataArray{col},...
        ones(length(dataArray{col}), 1));
end
numericData = NaN(size(dataArray{1},1),size(dataArray,2));
for col=[1,2,3,6,7,10,11]
    rawData = dataArray{col};
    for row=1:size(rawData, 1)
        % Create a expression to detect/remove non-numeric
        % prefixes and suffixes.
        regexstr = '(?<prefix>.*?)(?<numbers>([-
]*(\d+[\,]*)+[\.]{0,1}\d*[eEdD]{0,1}[-+]*\d*[i]{0,1})|([-
]*(\d+[\,]*)*[\.]{1,1}\d+[eEdD]{0,1}[-+]*\d*[i]{0,1}))(?<suffix>.*)';
        try
            result = regexp(rawData(row), regexstr, 'names');
            numbers = result.numbers;
            % Detected commas in non-thousand locations.
            invalidThousandsSeparator = false;
            if numbers.contains(',')
                thousandsRegExp = '^\d+?(\,\d{3})*\.{0,1}\d*$';
                if isempty(regexp(numbers, thousandsRegExp, 'once'))
                    numbers = NaN;
```

```
                    invalidThousandsSeparator = true;
                end
            end
            % Convert numeric text to numbers.
            if ~invalidThousandsSeparator
                numbers = textscan(char(strrep(numbers, ',', '')),...
                    '%f');
                numericData(row, col) = numbers{1};
                raw{row, col} = numbers{1};
            end
        catch
            raw{row, col} = rawData{row};
        end
    end
end
% Convert the contents of columns with dates to
% MATLAB datetimes using the specified date format.
try
    dates{8} = datetime(dataArray{8}, 'Format', 'yyyy-MM-dd
HH:mm:ss',...
        'InputFormat', 'yyyy-MM-dd HH:mm:ss');
catch
    try
        % Handle dates surrounded by quotes
        dataArray{8} = cellfun(@(x) x(2:end-1), dataArray{8},...
            'UniformOutput', false);
        dates{8} = datetime(dataArray{8}, 'Format',...
            'yyyy-MM-dd HH:mm:ss', 'InputFormat', 'yyyy-MM-dd
HH:mm:ss');
    catch
        dates{8} = repmat(datetime([NaN NaN NaN]), size(dataArray{8}));
    end
end
dates = dates(:,8);
% Split data into numeric and string columns.
rawNumericColumns = raw(:, [1,2,3,6,7,10,11]);
rawStringColumns = string(raw(:, [4,5,9]));
% Replace non-numeric cells with NaN
R = cellfun(@(x) ~isnumeric(x) && ~islogical(x),rawNumericColumns);
% Find non-numeric cells
rawNumericColumns(R) = {NaN}; % Replace non-numeric cells
% Create output variable
GSdata = table;
GSdata.decodes = cell2mat(rawNumericColumns(:, 1));
GSdata.preambles = cell2mat(rawNumericColumns(:, 2));
GSdata.AZo = cell2mat(rawNumericColumns(:, 3));
GSdata.notes = rawStringColumns(:, 1);
GSdata.gs_name = rawStringColumns(:, 2);
GSdata.AZf = cell2mat(rawNumericColumns(:, 4));
GSdata.duration = cell2mat(rawNumericColumns(:, 5));
GSdata.AOS = dates{:, 1};
GSdata.sat_name = rawStringColumns(:, 3);
GSdata.good_bits_rx = cell2mat(rawNumericColumns(:, 6));
GSdata.max_EL = cell2mat(rawNumericColumns(:, 7));
% Clear temporary variables
```

```
clearvars -except GSdata

%%
% ===== 2. GROUND STATION AZo VS EL PLOT =====
% Prereqs: CSV import from Script 1

%%
% Following import of CSV data, use the following script to plot
% AZo/EL data for specific ground station, satellite, and time
close all; clearvars -except GSdata; clc
% User selects Ground Station name
UserGS = input...
    ('\nAFIT / HSFL / PTSUR / SDL / UNM\nSelect Ground Station: ','s');
% User selects satellite name
UserSat = input...
    ('\n13FLORA / 10MERRYW / FAUNA\nSelect Satellite: ','s');
% User selects month
UserDate = input('\nMM-YYYY or ALL\nSelect Month: ','s');
% Extract data for User Sat/GS/Date selection
Count=1;
formatOut = 'mm-yyyy';
for i=1:length(GSdata.AZo);
    if GSdata.gs_name(i)==UserGS
        if GSdata.sat_name(i)==UserSat
            if strcmp(strread(UserDate,'%3s'),'ALL')
                AZoGS(Count)=GSdata.AZo(i);
                AZfGS(Count)=GSdata.AZf(i);
                ELGS(Count)=GSdata.max_EL(i);
                decodesGS(Count)=GSdata.decodes(i);
                preamblesGS(Count)=GSdata.preambles(i);
                Count=Count+1;
            elseif datestr(GSdata.AOS(i),formatOut)==UserDate
                AZoGS(Count)=GSdata.AZo(i);
                AZfGS(Count)=GSdata.AZf(i);
                ELGS(Count)=GSdata.max_EL(i);
                decodesGS(Count)=GSdata.decodes(i);
                preamblesGS(Count)=GSdata.preambles(i);
                Count=Count+1;
            end
        end
    end
end
% Plot AZo vs EL with different symbols for Preambles/Decodes/None
CountPre = 1; CountDec = 1; CountNone = 1;
hold on
% Separate "for" loops ensure that the plot order is None, Pre, Dec
for j=1:length(AZoGS);
    % Correct for data points that cause "discontinuity" in plot
    if AZoGS(j) < 90;
        AZoGS(j) = AZoGS(j)+360;
    end
    if decodesGS(j)==0;
        if preamblesGS(j)==0;
            plot(AZoGS(j),ELGS(j),'k*','MarkerSize',3)
            CountNone = CountNone + 1;
```

```
        end
    end
end
for j=1:length(AZoGS);
    if preamblesGS(j)>0;
        if decodesGS(j)==0;
            plot(AZoGS(j),ELGS(j),'g*','MarkerSize',3)
            CountPre=1+CountPre;
        end
    end
end
for j=1:length(AZoGS);
    if decodesGS(j)>0;
        plot(AZoGS(j),ELGS(j),'r*','MarkerSize',3)
        CountDec=1+CountDec;
    end
end
XTickVal = 90:45:450;
xticks([XTickVal]);
xticklabels({'90\circ','135\circ','180\circ','225\circ','270\circ',...
    '315\circ','0\circ','45\circ','90\circ'});
xlim([90 450]);
ylim([0 100]);
yticklabels({'0\circ','10\circ','20\circ','30\circ','40\circ',...
    '50\circ','60\circ','70\circ','80\circ','90\circ'})
xlabel('Initial Azimuth','FontSize',13);
ylabel('Maximum Elevation Angle','FontSize',13);
title(sprintf('%s - %s', UserGS, UserSat),'FontSize',14);
grid on
set(gcf,'color','w')
clearvars -except GSdata UserDate UserGS UserSat

%%
% ===== 3. GROUND STATION AZo VS PASS DURATION PLOT =====
% Prereqs: CSV import from Script 1

%%
% Following import of CSV data, use the following script to plot AZo vs
% duration data for specific ground station, satellite, and time
% period.
close all; clearvars -except GSdata; clc
% User selects Ground Station name
UserGS = input(...
    '\nAFIT / HSFL / PTSUR / SDL / UNM\nSelect Ground Station: ','s');
% User selects satellite name
UserSat = input...
    ('\n13FLORA / 10MERRYW / FAUNA\nSelect Satellite: ','s');
% User selects month
UserDate = input('\nMM-YYYY or ALL\nSelect Month: ','s');
% Extract data for User Sat/GS/Date selection
Count=1;
formatOut = 'mm-yyyy';
for i=1:length(GSdata.AZo);
    if GSdata.gs_name(i)==UserGS
        if GSdata.sat_name(i)==UserSat
```

```
            if strcmp(strread(UserDate,'%3s'),'ALL')
                AZoGS(Count)=GSdata.AZo(i);
                DurGS(Count)=GSdata.duration(i);
                DateGS(Count)=GSdata.AOS(i);
                Count=Count+1;
            elseif datestr(GSdata.AOS(i),formatOut)==UserDate
                AZoGS(Count)=GSdata.AZo(i);
                DurGS(Count)=GSdata.duration(i);
                DateGS(Count)=GSdata.AOS(i);
                Count=Count+1;
            end
        end
    end
end
% Plot AZo vs Duration
for j=1:length(AZoGS);
    % Correct for AZo data points that cause "discontinuity" in plot
    if AZoGS(j) < 90;
        AZoGS(j) = AZoGS(j)+360;
    end
end
plot(AZoGS,DurGS,'b.','MarkerSize',10)
XTickVal = 90:45:450;
xticks([XTickVal]);
xticklabels({'90\circ','135\circ','180\circ','225\circ','270\circ',...
    '315\circ','0\circ','45\circ','90\circ'});
xlim([90 450]);
ylim([300 700]);
xlabel('Initial Azimuth','FontSize',13);
ylabel('Pass Duration (Seconds)','FontSize',13);
title(sprintf('%s - %s', UserGS, UserSat),'FontSize',14);
grid on
set(gcf,'color','w')
clearvars -except GSdata UserDate UserGS UserSat AZoGS

%%
% ===== 4. PREAMBLE AND DECODE ANALYSIS BY AZo BINS (BAR PLOTS) =====
% Prereqs: CSV import from Script 1

%%
% Following import of CSV data, use the following script to determine
% AZo bins for for specific ground station, satellite, and time period.
close all; clearvars -except GSdata; clc
% User selects Ground Station name
UserGS = input(...
    '\nAFIT / HSFL / PTSUR / SDL / UNM\nSelect Ground Station: ','s');
% User selects satellite name
UserSat = input...
    ('\n13FLORA / 10MERRYW / FAUNA\nSelect Satellite: ','s');
% User selects month
UserDate = input('\nMM-YYYY or ALL\nSelect Month: ','s');
% Extract data for User Sat/GS/Date selection
Count=1;
formatOut = 'mm-yyyy';
for i=1:length(GSdata.AZo);
```

```
    if GSdata.gs_name(i)==UserGS
        if GSdata.sat_name(i)==UserSat
            if strcmp(strread(UserDate,'%3s'),'ALL')
                AZoGS(Count)=GSdata.AZo(i);
                AZfGS(Count)=GSdata.AZf(i);
                ELGS(Count)=GSdata.max_EL(i);
                decodesGS(Count)=GSdata.decodes(i);
                preamblesGS(Count)=GSdata.preambles(i);
                Count=Count+1;
            elseif datestr(GSdata.AOS(i),formatOut)==UserDate
                AZoGS(Count)=GSdata.AZo(i);
                AZfGS(Count)=GSdata.AZf(i);
                ELGS(Count)=GSdata.max_EL(i);
                decodesGS(Count)=GSdata.decodes(i);
                preamblesGS(Count)=GSdata.preambles(i);
                Count=Count+1;
            end
        end
    end
end
% Correct for AZo data points that cause "discontinuity" in plot
for j=1:length(AZoGS);
    if AZoGS(j) < 90;
        AZoGS(j) = AZoGS(j)+360;
    end
end
% Create AZo bins (evenly spaced by degrees) for data analysis
Bins = 40;
DataTable = [AZoGS;AZfGS;ELGS;decodesGS;preamblesGS]';
DataTable = sortrows(DataTable,1);
[N,edges,bin] = histcounts(DataTable(:,1),Bins);
% Extract data from bins, probability of decode and avg decode per bin
% Col 1 - Bin Number
% Col 2 - Left AZo
% Col 3 - Right AZo
% Col 4 - Average Max El for Bin
% Col 5 - Total Number of Passes in Bin
% Col 6 - Number of Passes in Bin with atleast 1 Decode
% Col 7 - Total Number of Decodes in Bin
% Col 8 - Number of Passes in Bin with atleast 1 Preamble
% Col 9 - Successful Decode Passes / Total Number of Passes
% Col 10 - Total Decodes / Total Passes (Avg decodes per pass)
% Col 11 - Successful Preamble Passes / Total Number of Passes
BinStats(:,1) = 1:Bins;
BinStats(:,5) = N';
for k=1:Bins;
    BinStats(k,2) = edges(k);
    BinStats(k,3) = edges(k+1);
    BinInd = find(bin==k);
    BinStats(k,4) = mean(DataTable(min(BinInd):max(BinInd),3));
    BinStats(k,6) = length(find(DataTable(BinInd,4)>=1));
    BinStats(k,7) = sum(DataTable(BinInd,4));
    BinStats(k,8) = length(find(DataTable(BinInd,5)>=1));
end
BinStats(:,9) = BinStats(:,6)./BinStats(:,5).*100;
```

```matlab
BinStats(:,10) = BinStats(:,7)./BinStats(:,5);
BinStats(:,11) = BinStats(:,8)./BinStats(:,5).*100;
% Convert NaN from dividing by zero into zero value
BinStats(isnan(BinStats))=0;
% Plot the histograms of the pass data for given satellite/GS
subplot(3,1,1)
bar((BinStats(:,2)+BinStats(:,3))/2,BinStats(:,11),'b');
title(sprintf('%s - %s', UserGS,UserSat),'FontSize',14);
XTickVal = 90:45:450;
xticks([XTickVal]);
xticklabels({'90\circ','135\circ','180\circ','225\circ','270\circ',...
    '315\circ','0\circ','45\circ','90\circ'});
xlim([90 450]);
ylim([0 105]);
ylabel('% with Preambles','FontSize',13);
grid on
subplot(3,1,2)
bar((BinStats(:,2)+BinStats(:,3))/2,BinStats(:,9),'b');
XTickVal = 90:45:450;
xticks([XTickVal]);
xticklabels({'90\circ','135\circ','180\circ','225\circ','270\circ',...
    '315\circ','0\circ','45\circ','90\circ'});
xlim([90 450]);
ylim([0 105]);
ylabel('% with Decodes','FontSize',13);
grid on
subplot(3,1,3)
bar((BinStats(:,2)+BinStats(:,3))/2,BinStats(:,10),'b');
XTickVal = 90:45:450;
xticks([XTickVal]);
xticklabels({'90\circ','135\circ','180\circ','225\circ','270\circ',...
    '315\circ','0\circ','45\circ','90\circ'});
xlim([90 450]);
ylim([0 max(BinStats(:,10)+5)]);
ylabel('Average Decodes per Pass','FontSize',13);
xlabel('Initial Azimuth (Degrees)','FontSize',13);
grid on
set(gcf,'color','w')
clearvars -except GSdata BinStats UserDate UserGS UserSat

%%
% ===== 5. DECODE ANALYSIS BY AZo BINS (POLAR PLOT) =====
% Prereqs: CSV import from Script 1

%%
% Following import of CSV data, use the following script to determine
% AZo bins for for specific ground station, satellite, and time period.
close all; clearvars -except GSdata; clc
% User selects Ground Station name
UserGS = input(...
    '\nAFIT / HSFL / PTSUR / SDL / UNM\nSelect Ground Station: ','s');
% User selects satellite name
UserSat = input...
    ('\n13FLORA / 10MERRYW / FAUNA\nSelect Satellite: ','s');
% Extract data for User Sat/GS selection
```

```
Count=1;
for i=1:length(GSdata.AZo);
    if GSdata.gs_name(i)==UserGS
        if GSdata.sat_name(i)==UserSat
            AZoGS(Count)=GSdata.AZo(i);
            AZfGS(Count)=GSdata.AZf(i);
            ELGS(Count)=GSdata.max_EL(i);
            decodesGS(Count)=GSdata.decodes(i);
            preamblesGS(Count)=GSdata.preambles(i);
            Count=Count+1;
        end
    end
end
% Correct for AZo data points that cause "discontinuity" in plot
for j=1:length(AZoGS);
    if AZoGS(j) < 90;
        AZoGS(j) = AZoGS(j)+360;
    end
end
% Create AZo bins (evenly spaced by degrees) for data analysis
DataTable = [AZoGS;AZfGS;ELGS;decodesGS;preamblesGS]';
DataTable = sortrows(DataTable,1);
edges = 0:5:360;
Bins = length(edges)-1;
[N,edges,bin] = histcounts(DataTable(:,1),edges);
% Extract data from bins, probability of decode and avg decode per bin
% Col 1 - Bin Number
% Col 2 - Left AZo
% Col 3 - Right AZo
% Col 4 - Average Max El for Bin
% Col 5 - Total Number of Passes in Bin
% Col 6 - Number of Passes in Bin with atleast 1 Decode
% Col 7 - Total Number of Decodes in Bin
% Col 8 - Number of Passes in Bin with atleast 1 Preamble
% Col 9 - Successful Decode Passes / Total Number of Passes
% Col 10 - Total Decodes / Total Passes (Avg decodes per pass)
% Col 11 - Successful Preamble Passes / Total Number of Passes
BinStats(:,1) = 1:Bins;
BinStats(:,5) = N';
for k=1:Bins;
    BinStats(k,2) = edges(k);
    BinStats(k,3) = edges(k+1);
    BinInd = find(bin==k);
    BinStats(k,4) = mean(DataTable(min(BinInd):max(BinInd),3));
    BinStats(k,6) = length(find(DataTable(BinInd,4)>=1));
    BinStats(k,7) = sum(DataTable(BinInd,4));
    BinStats(k,8) = length(find(DataTable(BinInd,5)>=1));
end
BinStats(:,9) = BinStats(:,6)./BinStats(:,5).*100;
BinStats(:,10) = BinStats(:,7)./BinStats(:,5);
BinStats(:,11) = BinStats(:,8)./BinStats(:,5).*100;
% Convert NaN from dividing by zero into zero value
BinStats(isnan(BinStats))=0;
% Determine Acceptance Region of AZo for Asc/Des Passes (Rounded)
EL40 = intersect(find(ELGS>40),find(ELGS<45));
```

```
AscHi = pi/180*(5*ceil(max(AZoGS(EL40(find...
    ((AZoGS(EL40))<270))))/5)+5);
AscLo = pi/180*(5*floor(min(AZoGS(EL40))/5)-5);
DesHi = pi/180*(5*ceil(max(AZoGS(EL40))/5)+5);
DesLo = pi/180*(5*floor(min(AZoGS(EL40(find...
    ((AZoGS(EL40))>270))))/5)-5);
% Plot Polar Histogram
theta = linspace(0,2*pi);
polarplot(theta,zeros(size(theta)),'k')
thetaticks(0:10:350)
pax = gca;
pax.FontSize = 10;
pax.GridLineStyle = '-';
pax.ThetaDir = 'clockwise';
pax.ThetaZeroLocation = 'top';
pax.RAxisLocation = 271;
set(gcf,'color','w');
rlim([0 round(max(BinStats(:,10)/5))*5+6])
rticks([5:5:100])
rticklabels({'5','10','15','20','25','30','35','40','45','50','60',...
    '65','70','75','80','85','90','95','100'})
thetaticklabels({'\fontsize{8}000\circ','\fontsize{8}010\circ',...
   '\fontsize{8}020\circ','\fontsize{8}030\circ',...
   '\fontsize{8}040\circ','\fontsize{8}050\circ',...
   '\fontsize{8}060\circ','\fontsize{8}070\circ',...
   '\fontsize{8}080\circ','\fontsize{8}090\circ',...
   '\fontsize{8}100\circ','\fontsize{8}110\circ',...
   '\fontsize{8}120\circ','\fontsize{8}130\circ',...
   '\fontsize{8}140\circ','\fontsize{8}150\circ',...
   '\fontsize{8}160\circ','\fontsize{8}170\circ',...
   '\fontsize{8}180\circ','\fontsize{8}190\circ',...
   '\fontsize{8}200\circ','\fontsize{8}210\circ',...
   '\fontsize{8}220\circ','\fontsize{8}230\circ',...
   '\fontsize{8}240\circ','\fontsize{8}250\circ',...
   '\fontsize{8}260\circ','\fontsize{8}270\circ',...
   '\fontsize{8}280\circ','\fontsize{8}290\circ',...
   '\fontsize{8}300\circ','\fontsize{8}310\circ',...
   '\fontsize{8}320\circ','\fontsize{8}330\circ',...
   '\fontsize{8}340\circ','\fontsize{8}350\circ'})
title(sprintf('%s - %s - Decodes/Pass vs Initial Azimuth',...
    UserGS, UserSat),'FontSize',12)
hold on
% Acceptance Regions for Azo vs MaxEl = 40 degrees
polarplot([AscHi AscHi AscLo AscLo],[0 1000 1000 0],'-k')
polarplot([DesHi DesHi DesLo DesLo],[0 1000 1000 0],'-k')
polarhistogram('BinEdges',[edges/180*pi],'BinCounts',BinStats(:,10),...
    'EdgeColor','b','FaceColor','b','FaceAlpha',1)
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D. TLE DATA SET ANALYSIS MATLAB SCRIPT

```
% ===== APPENDIX D. TLE DATA SET ANALYSIS =====

% Portions of script generated from MATLAB Data Import Script Generator

% ===== 1. TLE COMPILATION FOR EACH CUBESAT =====
% Prereqs: TLE Files for Flora, Merry, Fauna from Appendix B

%%
% This script extracts the historical TLE data for Flora, Merry, and
% Fauna from multiple TLE files. All data extracted from
% Space-Track.org.
% Flora and Merry - Extracted from "bulk" TLE files for L55 mission
% Fauna - Extracted from single TLE list file from Satellite Catalog
% Number before running script, ensure all TLE files are .txt extension
% Refer to Appendix B for procedures obtain TLE and convert to .txt
close all; clear; clc
% Delete any existing Flora and Merry TLE files so data is not appended
fclose('all');
delete('FilesTLE\!FloraTLE.txt')
delete('FilesTLE\!MerryTLE.txt')
delete('FilesTLE\!FaunaTLE.txt')
% Create listing of all TLE Files in TLE directory
dirTLE = dir('FilesTLE\*.txt');
% Establish counter so data appended to extracted file each iteration
FloraCount=1;
MerryCount=1;
FaunaCount=1;
for i = 1:length(dirTLE)
    % Create working filename for current TLE extract
    TLEname = dirTLE(i).name;
    TLEfolder = dirTLE(i).folder;
    filename = [TLEfolder,'\',TLEname];
    % Format for each text line, open text file, read columns of data
    formatSpec = '%s%[^\n\r]';
    fileID = fopen(filename,'r');
    dataArray = textscan(fileID, formatSpec, 'Delimiter', '',...
        'WhiteSpace', '', 'TextType', 'string','ReturnOnError', false);
    % Close the text file, create output variable, clear temp variables
    fclose(fileID);
    TLE = [dataArray{1:end-1}];
    clearvars TLEname TLEfolder filename formatSpec fileID dataArray...
        ans;
    % Begin loop to search for Flora, Merry, Fauna data
    for j = 1:length(TLE);
        TLErow = TLE{j};
        SatNum = TLErow(3:7);
        % Extract Flora (90736) TLE lines from file
        if strcmp(SatNum,'90736') % Flora
            FloraTLE{FloraCount,1} = TLErow;
            FloraCount = FloraCount+1;
        end
```

```
        % Extract Merry (90738) TLE lines from file
        if strcmp(SatNum,'90738') % Merry
            MerryTLE{MerryCount,1} = TLErow;
            MerryCount = MerryCount+1;
        end
        % Extract Fauna (43052) TLE lines from file
        if strcmp(SatNum,'43052') % Fauna
            FaunaTLE{FaunaCount,1} = TLErow;
            FaunaCount = FaunaCount+1;
        end
    end
end
% Export extracted Flora, Merry, Fauna data to exclusive txt files
FloraFileID = fopen('FilesTLE\!FloraTLE.txt','w');
[nrows,ncols] = size(FloraTLE);
for row = 1:nrows
    fprintf(FloraFileID,'%s\n',FloraTLE{row,:});
end
MerryFileID = fopen('FilesTLE\!MerryTLE.txt','w');
[nrows,ncols] = size(MerryTLE);
for row = 1:nrows
    fprintf(MerryFileID,'%s\n',MerryTLE{row,:});
end
FaunaFileID = fopen('FilesTLE\!FaunaTLE.txt','w');
[nrows,ncols] = size(FaunaTLE);
for row = 1:nrows
    fprintf(FaunaFileID,'%s\n',FaunaTLE{row,:});
end
fclose('all');
clearvars -except dirTLE FloraTLE MerryTLE FaunaTLE

%%
% ===== 2. HISTORIC RANGE AND LOSSES DUE TO PERIGEE ROTATION =====
% Prereqs: TLE Files created by Script 1

%%
% Script extracts the COE from historic TLE information and displays
% range change and associated free space losses due to perigee rotation
close all; clear; clc
% User selects Ground Station name
UserGS = input...
    ('\nAFIT / HSFL / PTSUR / SDL / UNM\nSelect Ground Station: ','s');
% User selects satellite name
UserSat = input...
    ('\n13FLORA / 10MERRYW / FAUNA\nSelect Satellite: ','s');
% Ground Station information table (latitude) for future calculations
GSinfo = table({'AFIT';'HSFL';'PTSUR';'SDL';'UNM'},[39.7821;21.2983;...
    36.5949;41.7606;35.0539]);
GSinfo.Properties.VariableNames = {'GS_name','Latitude'};
% Satellite information table to choose correct TLE file
Satinfo = table({'13FLORA';'10MERRYW';'FAUNA'},{'!FloraTLE';...
    '!MerryTLE';'!FaunaTLE'});
Satinfo.Properties.VariableNames = {'Sat_name','TLE'};
% Pull applicable GS, Sat, TLEname data from tables for user selections
UserLat = GSinfo.Latitude(find(strcmp(UserGS,GSinfo.GS_name)));
```

```
UserTLE = Satinfo.TLE(find(strcmp(UserSat,Satinfo.Sat_name)));
UserTLE = UserTLE{1:end};
% Extract all TLE from file into a MATLAB variable
filename = ['FilesTLE\',UserTLE,'.txt'];
formatSpec = '%s%[^\n\r]';
fileID = fopen(filename,'r');
dataArray = textscan(fileID, formatSpec, 'Delimiter', '',...
    'WhiteSpace','', 'TextType', 'string',  'ReturnOnError', false);
fclose(fileID);
TLE = [dataArray{1:end-1}];
% Input Constants
muE = 398600;
RE = 6378;
% Determine Dates and COEs for historic TLE files
count = 1;
for i=1:2:(length(TLE)-1)
    n = str2num(TLE{i+1}(53:63));
    a(count) = (muE^(1/3))/((2*n*pi/86400)^(2/3));
    e(count) = str2num(['0.',TLE{i+1}(27:33)]);
    inc(count) = str2num(TLE{i+1}(9:16));
    omega(count) = str2num(TLE{i+1}(35:42));
    doy = TLE{i}(21:23);
    year = ['20',TLE{i}(19:20)];
    date(count) = datetime(str2num(year),1,str2num(doy));
    count = count + 1;
end
% Determine True Anomaly corresponding to GS pass (asc and des)
% Value dependent on Inc, argument of perigee, and GS latitude
UserInc = mean(inc);
UserMinLoss = asind(sind(UserLat)/sind(UserInc));
for j=1:length(date)
    if omega(j) < UserMinLoss
        TAa(j) = UserMinLoss-omega(j);
    else
        TAa(j) = UserMinLoss-omega(j)+360;
    end
    TAd(j) = TAa(j) + 180 - 2*UserMinLoss;
    if TAd(j) >= 360
        TAd(j) = TAd(j) - 360;
    end
    % Use True Anomaly value to determine Range to Ground Station (asc)
    Ra(j) = a(j)*((1-e(j)^2)/(1+e(j)*cosd(TAa(j))));
    % Determine altitude of CPA for various maximum elevation angles
    Rng90a(j) = Ra(j)-RE;
    Rng80a(j) = Rng90a(j)/sind(80);
    Rng70a(j) = Rng90a(j)/sind(70);
    Rng60a(j) = Rng90a(j)/sind(60);
    Rng50a(j) = Rng90a(j)/sind(50);
    Rng40a(j) = Rng90a(j)/sind(40);
    % Use True Anomaly value to determine Range to Ground Station (des)
    Rd(j) = a(j)*((1-e(j)^2)/(1+e(j)*cosd(TAd(j))));
    % Determine altitude of CPA for various maximum elevation angles
    Rng90d(j) = Rd(j) - RE;
    Rng80d(j) = Rng90d(j)/sind(80);
    Rng70d(j) = Rng90d(j)/sind(70);
```

```
    Rng60d(j) = Rng90d(j)/sind(60);
    Rng50d(j) = Rng90d(j)/sind(50);
    Rng40d(j) = Rng90d(j)/sind(40);
    % Using referece range of 500 km, calculate dB losses
    RefRng=500;
    dBa(j) = 10*log10((RefRng^2)/((Rng90a(j)^2)));
    dBd(j) = 10*log10((RefRng^2)/((Rng90d(j)^2)));
    % dB loss values for all maximum elevations will be the same
end
% Plot 1: Ascending Pass Min Range
subplot(2,2,1)
plot(date,Rng90a,date,Rng80a,date,Rng70a,date,Rng60a,date,...
    Rng50a,date,Rng40a)
title(sprintf('%s - %s - Ascending Pass Min Range', UserGS,...
    UserSat),'FontSize',13);
dateFormat = 10;
datetick('x',dateFormat)
xlim([date(1) date(end)]);
ylim([400 1600]);
xlabel('Historical Pass Date','FontSize',13);
ylabel('Pass Min Range (km)','FontSize',13);
legend('Max El 90\circ','Max El 80\circ','Max El 70\circ',...
    'Max El 60\circ','Max El 50\circ','Max El 40\circ')
grid on
% Plot 2: Ascending Pass Loss (as compared to 500 km range)
subplot(2,2,2)
plot(date,dBa,'k')
title(sprintf('%s - %s - Ascending Pass Losses', UserGS, ...
    UserSat),'FontSize',13);
dateFormat = 10;
datetick('x',dateFormat)
xlim([date(1) date(end)]);
ylim([-5 2]);
xlabel('Historical Pass Date','FontSize',13);
ylabel('dB Loss (Ref Alt = 500 km)','FontSize',13);
legend('Max El 40\circ - 90\circ','Location','northwest')
grid on
% Plot 3: Descending Pass Min Range
subplot(2,2,3)
plot(date,Rng90d,date,Rng80d,date,Rng70d,date,Rng60d,date,...
    Rng50d,date,Rng40d)
title(sprintf('%s - %s - Descending Pass Min Range', UserGS, ...
    UserSat),'FontSize',13);
dateFormat = 10;
datetick('x',dateFormat)
xlim([date(1) date(end)]);
ylim([400 1600]);
xlabel('Historical Pass Date','FontSize',13);
ylabel('Pass Min Range (km)','FontSize',13);
legend('Max El 90\circ','Max El 80\circ','Max El 70\circ',...
    'Max El 60\circ',...
    'Max El 50\circ','Max El 40\circ','Location','north')
grid on
% Plot 4: Descending Pass Loss (as compared to 500 km range)
subplot(2,2,4)
```

```
plot(date,dBd,'k')
title(sprintf('%s - %s - Descending Pass Losses', UserGS,...
    UserSat),'FontSize',13);
dateFormat = 10;
datetick('x',dateFormat)
xlim([date(1) date(end)]);
ylim([-5 2]);
xlabel('Historical Pass Date','FontSize',13);
ylabel('dB Loss (Ref Alt = 500 km)','FontSize',13);
legend('Max El 40\circ - 90\circ','Location','northwest')
grid on
set(gcf,'color','w')
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E. EXTRANEOUS LOG AND TRACK FILE DELETION MATLAB SCRIPT

```
% ===== APPENDIX E. EXTRANEOUS LOG AND TRACK FILE DELETION =====

% Portions of script generated from MATLAB Data Import Script Generator

% ===== 1. DELETE UNNEEDED TRACK AND LOG FILES =====
% Prereqs: Two folders with MC3 Network track and log files

%%
% This script deletes all Raw and Track files that are not required
% for data analysis:
% Raw: All files that have no data (0 bytes)
% Track: Ground Stations NZL, Tyvak, NPS, HSFL2, UKSquareDance
% Track: Satellites DIDO, Celtee, Amybeth, Buccaneer
% Track: All files that do not start/end at 10 degrees elevation
% Track: All Fauna files with the incorrect satellite catalog number
clear; close all; clc

% Delete Raw files that have no data (0 bytes)
% Create listing of all Raw files in directory
dirRaw = dir('FilesLog\*raw*.txt');
for i=1:size(dirRaw,1)
    RawFilename = dirRaw(i).name;
    if dirRaw(i).bytes == 0;
       delete(['FilesLog\' RawFilename])
    end
end
% No file created for 0 byte KISS, so no need to perform on dirKiss

% Delete Track files for Satelites and GS not of interest
% Create listing of all Track files in directory
dirTrack = dir('FilesTrack\*.dat');
for i=1:size(dirTrack,1);
    disp(['Current Track File (Delete Extra Sats/GS): ' num2str(i)...
        ' of ' num2str(size(dirTrack,1))])
    TrackFilename = dirTrack(i).name;
    GS = string(extractBefore(extractAfter(extractAfter...
        (TrackFilename,'_'),'_'),'_'));
    Sat =
string(extractBefore(extractAfter(extractAfter(extractAfter...
        (TrackFilename,'_'),'_'),'_'),'_'));
    if GS == 'NZL' || ...
            GS == 'Tyvak' || ...
            GS == 'NPS' || ...
            GS == 'HSFL2' || ...
            GS == 'UKSquareDance' || ...
            Sat == 'FAUNAi' ||...
            Sat == 'DIDO' || ...
            Sat == 'CELTEE-1' || ...
            Sat == 'AMYBETHB' || ...
```

```
            Sat == 'AMYBETHC' || ...
            Sat == 'AMYBETHD' || ...
            Sat == 'AMYBETH' || ...
            Sat == 'BUCCANEER' ||...
            Sat == 'BUCCANEERI'
        delete(['FilesTrack\' TrackFilename])
    end
    % Delete Fauna tracks with incorrect satellite catalog number
    if Sat == 'FAUNA'
        CatNo =
string(extractBefore(extractAfter(extractAfter(extractAfter...
            (extractAfter(TrackFilename,'_'),'_'),'_'),'_'),'.'));
        if CatNo ~= '43052'
            delete(['FilesTrack\' TrackFilename])
        end
    end
end

% Delete all Track files that do not start or end at 10 deg elevation
% Create listing of all Track files in directory
dirTrack = dir('FilesTrack\*.dat');
for i=1:size(dirTrack,1);
    disp(['Current Track File (Elevation Angle Fix): ' num2str(i)...
        ' of ' num2str(size(dirTrack,1))])
    TrackFilename = ['FilesTrack\' dirTrack(i).name];
    % Initialize variables, format each line, and open text file
    delimiter = ' ';
    formatSpec = '%11s%19s%7f%6f%8f%s%[^\n\r]';
    fileID = fopen(TrackFilename,'r');
    % Read columns of data
    dataArray = textscan(fileID, formatSpec, 'Delimiter', '',...
        'WhiteSpace', '', 'TextType', 'string', 'ReturnOnError',...
        false);
    fclose(fileID);
    % Column 4 = Elevation
    FirstEl = dataArray{4}(1);
    LastEl = dataArray{4}(end);
    MaxEl = max(dataArray{4});
    if FirstEl ~= 10 || LastEl ~= 10 || MaxEl == 10;
        delete(TrackFilename)
    end
end
```

APPENDIX F. SEGMENTED PASS ANALYSIS MATLAB SCRIPT

```
% ===== APPENDIX F. SEGMENTED PASS ANALYSIS =====

% Portions of script generated from MATLAB Data Import Script Generator

% ===== 1. SINGLE PASS TRACK DATA IMPORT AND PLOT =====
% Prereqs: Pass Track Data File Name in 'filename =' line

%%
% This script imports data for a single pass from pass's .dat file
close all; clear all; clc
% Confirm filename and directory of .dat file
% Initialize variables, format each line, and open text file
filename = 'FilesTrack\2017-05-11_02-18-01_PTSUR_13FLORA_90736.dat';
delimiter = ' ';
formatSpec = '%s%s%s%{HH:mm:ss.SSSSSSSSS}D%f%f%f%s%[^\n\r]';
fileID = fopen(filename,'r');
% Read columns of data
dataArray = textscan(fileID, formatSpec, 'Delimiter', delimiter,...
    'MultipleDelimsAsOne', true, 'TextType', 'string',...
    'ReturnOnError', false);
% Close text file, create output variable, clear temp variables
fclose(fileID);
PassData = table(dataArray{1:end-1}, 'VariableNames',...
    {'Day','Month','Year','Time','AZ','EL','Range','GS'});
clearvars -except PassData
% Create polar plot shell for data input with elevation tick
% marks from GS to horizon (0 - 90 degrees)
El = 0:10:90;
ElTick = El/90;
theta = linspace(0,2*pi);
polarplot(theta,ElTick(end)+zeros(size(theta)),'k')
thetaticks(0:20:350)
pax = gca;
pax.FontSize = 10;
pax.GridLineStyle = '-';
pax.ThetaDir = 'clockwise';
pax.ThetaZeroLocation = 'top';
set(gcf,'color','w');
rlim([ElTick(1) ElTick(end)])
rticks([ElTick(1) ElTick(3) ElTick(5) ElTick(7) ElTick(9)])
rticklabels({'\fontsize{6}90\circ','\fontsize{6}70\circ',...
    '\fontsize{6}50\circ','\fontsize{6}30\circ','\fontsize{6}10\circ'})
rticklabels({'','','','',''})
thetaticklabels({'\fontsize{8}000\circ','\fontsize{8}020\circ',...
    '\fontsize{8}040\circ','\fontsize{8}060\circ',...
    '\fontsize{8}080\circ','\fontsize{8}100\circ',...
    '\fontsize{8}120\circ','\fontsize{8}140\circ',...
    '\fontsize{8}160\circ','\fontsize{8}180\circ',...
    '\fontsize{8}200\circ','\fontsize{8}220\circ',...
    '\fontsize{8}240\circ','\fontsize{8}260\circ',...
    '\fontsize{8}280\circ','\fontsize{8}300\circ',...
```

```
    '\fontsize{8}320\circ','\fontsize{8}340\circ'})
title('MC3 Ground Station Pass')
hold on
% Plot pass data onto polar plot shell
SatEl = 1-PassData.EL/90;
SatAz = (PassData.AZ)*pi/180;
polarplot(SatAz,SatEl,'b')
clearvars -except PassData

%%
% ===== 2. EXTRACT RAW, KISS, TRACK INFO FROM .TXT AND .DATS =====
% Prereqs: Two folders with MC3 Network Track and Log files

%%
% This script extracts Time/Az/El and Time/Pckt information from .dat
% files for multiple passes.
clear; close all; clc
% Create listing of all files in "Raw Kiss Track Analysis" directory
dirRaw = dir(['FilesLog\*raw*.txt']);
dirKiss = dir(['FilesLog\*KISS*.txt']);
dirTrack = dir('FilesTrack\*.dat');
% Extract data from Raw files into table format (GS, Sat, Date, Time)
RawGS = strings;
RawSat = strings;
RawDate = strings;
RawTime = strings;
for i=1:size(dirRaw,1);
    RawGSi = strings;
    RawSati = strings;
    RawDatei = strings;
    RawTimei = strings;
    disp(['Current Raw File: ' num2str(i) ' of '...
        num2str(size(dirRaw,1))])
    % Initialize variables, format each line, and open text file
    filename = ['FilesLog\' dirRaw(i).name];
    delimiter = ' ';
    formatSpec = '%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%[^\n\r]';
    fileID = fopen(filename,'r');
    % Read Sat name from filename
    Sat = extractBefore(extractAfter(extractAfter...
        (filename,'raw_'),'_'),'_');
    % Read columns of data
    dataArray = textscan(fileID, formatSpec, 'Delimiter',...
        delimiter, 'MultipleDelimsAsOne', true, 'TextType',...
        'string',  'ReturnOnError', false);
    % Close text file, create output variables
    fclose(fileID);
    for col=1:length(dataArray)-1
        raw(1:length(dataArray{col}),col) = mat2cell(...
            dataArray{col},ones(length(dataArray{col}), 1));
    end
    rawStringColumns = string(raw(:,[1,2,3,4,5,6,7,9,11,12,13,14,16]));
    for j=1:size(rawStringColumns,1)
        % Only extract data from cells that have GS/Date/Time Format
        if rawStringColumns(j,1) == 'AFIT' ||...
```

```
                rawStringColumns(j,1) == 'HSFL' ||...
                rawStringColumns(j,1) == 'PTSUR' ||...
                rawStringColumns(j,1) == 'SDL' ||...
                rawStringColumns(j,1) == 'UNM'
            RawGSi = rawStringColumns(j,1);
            RawSati = Sat;
            if isempty(rawStringColumns{j,2}) == 0
                if rawStringColumns{j,2}(3) == '/'
                    RawDatei = rawStringColumns(j,2);
                end
                if rawStringColumns{j,3}(3) == ':'
                % Delete decimal from Raw Time to format hh:mm:ss
                    RawTimei = extractBefore(...
                        rawStringColumns(j,3),'.');
                    for k=1:size(RawGSi,1)
                        RawGS(end+1,1) = RawGSi(k);
                        RawSat(end+1,1) = RawSati;
                        RawDate(end+1,1) = RawDatei(k);
                        RawTime(end+1,1) = RawTimei(k);
                    end
                end
            end
        end
    end
    % Clear variables is required at this point so new
    % variables are created instead of overwriting the
    % old ones.  If old variable has more values than
    % new one, not all will be overwritten, inflating the number.
    clearvars -except dirKiss dirRaw dirTrack RawGS ...
        RawDate RawTime RawSat
end
% Extract data from all Kiss files into table format (GS, Date, Time)
KissGS = strings;
KissSat = strings;
KissDate = strings;
KissTime = strings;
for i=1:size(dirKiss,1);
    KissGSi = strings;
    KissSati = strings;
    KissDatei = strings;
    KissTimei = strings;
    disp(['Current Kiss File: ' num2str(i) ' of ' ...
        num2str(size(dirKiss,1))])
    % Initialize variables, format each line, and open text file
    filename = ['FilesLog\' dirKiss(i).name];
    delimiter = ' ';
    formatSpec = '%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%[^\n\r]';
    fileID = fopen(filename,'r');
    % Read Sat name from filename
    Sat = extractBefore(extractAfter(extractAfter...
        (filename,'KISS_'),'_'),'_');
    % Read columns of data
    dataArray = textscan(fileID, formatSpec, 'Delimiter',...
        delimiter, 'MultipleDelimsAsOne', true, 'TextType',...
        'string', 'ReturnOnError', false);
```

```
    % Close text file, create output variables
    fclose(fileID);
    for col=1:length(dataArray)-1
        raw(1:length(dataArray{col}),col) = mat2cell...
            (dataArray{col},...
            ones(length(dataArray{col}), 1));
    end
    rawStringColumns = string(raw(:,...
        [1,2,3,4,5,6,7,9,11,12,13,14,16]));
    for j=1:size(rawStringColumns,1)
        % Only extract data from cells that have GS/Date/Time Format
        if rawStringColumns(j,1) == 'AFIT' ||...
                rawStringColumns(j,1) == 'HSFL' ||...
                rawStringColumns(j,1) == 'PTSUR' ||...
                rawStringColumns(j,1) == 'SDL' ||...
                rawStringColumns(j,1) == 'UNM'
            KissGSi = rawStringColumns(j,1);
            KissSati = Sat;
            if isempty(rawStringColumns{j,2}) == 0
                if rawStringColumns{j,2}(3) == '/'
                    KissDatei = rawStringColumns(j,2);
                end
                if rawStringColumns{j,3}(3) == ':'
                % Delete decimal from Kiss Time to format hh:mm:ss
                    KissTimei = extractBefore...
                        (rawStringColumns(j,3),'.');
                    for k=1:size(KissGSi,1)
                        KissGS(end+1,1) = KissGSi(k);
                        KissSat(end+1,1) = KissSati;
                        KissDate(end+1,1) = KissDatei(k);
                        KissTime(end+1,1) = KissTimei(k);
                    end
                end
            end
        end
    end
    % Clear variables is required at this point so new
    % variables are created instead of overwriting the
    % old ones.  If old variable has more values than
    % new one, not all will be overwritten, inflating the number.
    clearvars -except dirKiss dirRaw dirTrack RawGS...
        RawSat RawDate RawTime KissGS KissSat KissDate KissTime
end
% Extract data from Track files into table format (GS, Sat, Date,
% Time, Azimuth, Elevation, Range)
TrackGS = strings;
TrackSat = strings;
TrackDate = strings;
TrackTime = strings;
TrackAZ = strings;
TrackEL = strings;
TrackRange = strings;
TrackPassStart = strings;
TrackPassEnd = strings;
TrackPassAZo = strings;
```

```
TrackPassMaxEl = strings;
for i=1:size(dirTrack,1);
    TrackGSi = strings;
    TrackSati = strings;
    TrackDatei = strings;
    TrackTimei = strings;
    TrackAZi = strings;
    TrackELi = strings;
    TrackRangei = strings;
    TrackPassStarti = strings;
    TrackPassEndi = strings;
    TrackPassAZoi = strings;
    TrackPassMaxEli = strings;
    disp(['Current Track File: ' num2str(i) ' of '...
        num2str(size(dirTrack,1))])
    % Initialize variables, format each line, and open text file
    filename = ['FilesTrack\' dirTrack(i).name];
    delimiter = ' ';
    formatSpec = '%11s%19s%7f%6f%8f%s%[^\n\r]';
    fileID = fopen(filename,'r');
    % Read Sat name from filename
    Sat = extractBefore(extractAfter(extractAfter(extractAfter...
        (filename,'_'),'_'),'_'),'_');
    % Read columns of data
    dataArray = textscan(fileID, formatSpec, 'Delimiter',...
        '', 'WhiteSpace', '', 'TextType', 'string',...
        'ReturnOnError', false);
    % Remove white space, close text file, create output variables
    dataArray{1} = strtrim(dataArray{1});
    dataArray{2} = strtrim(dataArray{2});
    dataArray{6} = strtrim(dataArray{6});
    fclose(fileID);
    for col=1:length(dataArray)-1
        raw(1:length(dataArray{col}),col) = mat2cell...
            (dataArray{col}, ones(length(dataArray{col}), 1));
    end
    TrackGSi = string(raw(:,6));
    for j=1:size(raw,1);
        TrackSati(j,1) = Sat;
        TrackPassStarti(j,1) = string(raw(1,2));
        TrackPassEndi(j,1) = string(raw(end,2));
        TrackPassAZoi(j,1) = string(raw(1,3));
        TrackPassMaxEli(j,1) = string(raw(ceil(end/2),4));
    end
    TrackDatei = string(raw(:,1));
    TrackTimei = string(raw(:,2));
    TrackAZi = string(raw(:,3));
    TrackELi = string(raw(:,4));
    TrackRangei = string(raw(:,5));
    for j=1:size(TrackGSi,1)
        TrackGS(end+1,1) = TrackGSi(j);
        TrackSat(end+1,1) = TrackSati(j);
        % Match date format of Raw and Kiss Dates (mm/dd/yy)
        TrackDate(end+1,1) =  datestr(datenum...
            (char(TrackDatei(j))),'mm/dd/yy');
```

```
        % Delete decimal from Track Time to format hh:mm:ss
        TrackTime(end+1,1) = extractBefore(TrackTimei(j),'.');
        TrackAZ(end+1,1) = TrackAZi(j);
        TrackEL(end+1,1) = TrackELi(j);
        TrackRange(end+1,1) = TrackRangei(j);
        % Delete decimal from Start Time format hh:mm:ss
        TrackPassStart(end+1,1) = extractBefore...
            (TrackPassStarti(j),'.');
        % Delete decimal from End Time format as hh:mm:ss
        TrackPassEnd(end+1,1) = extractBefore...
            (TrackPassEndi(j),'.');
        TrackPassAZo(end+1,1) = TrackPassAZoi(j);
        TrackPassMaxEl(end+1,1) = TrackPassMaxEli(j);
    end
    clearvars -except dirKiss dirRaw dirTrack RawGS RawSat...
        RawDate RawTime KissGS KissSat KissDate KissTime...
        TrackGS TrackSat TrackDate TrackTime TrackAZ TrackEL...
        TrackRange TrackPassStart TrackPassEnd TrackPassAZo...
        TrackPassMaxEl
end
% Save the Files
SaveNameRaw = 'AllRaw';
SaveNameKiss = 'AllKiss';
SaveNameTrack1 = 'AllTrack1';
SaveNameTrack2 = 'AllTrack2';
SaveNameTrack3 = 'AllTrack3';
% Cannot save all files together due to size
% Save each of the files into the following groups
save(['FilesLog/' SaveNameRaw],'RawGS','RawSat','RawDate',...
    'RawTime')
save(['FilesLog/' SaveNameKiss],'KissGS','KissSat','KissDate',...
    'KissTime')
% Track must be segmented into multiple files
save(['FilesTrack/' SaveNameTrack1],'TrackGS','TrackSat',...
    'TrackDate','TrackTime')
save(['FilesTrack/' SaveNameTrack2],'TrackAZ','TrackEL',...
    'TrackRange')
save(['FilesTrack/' SaveNameTrack3],'TrackPassStart',...
    'TrackPassEnd','TrackPassAZo','TrackPassMaxEl')
clearvars SaveNameKiss SaveNameRaw SaveNameTrack1 SaveNameTrack2...
    SaveNameTrack3
%Create Tables of Raw, Kiss, and Track Data
% Begins at cell 2 due to empty cell 1 during import process
RawTable = table;
Raw.GS = RawGS(2:end);
Raw.Sat = RawSat(2:end);
Raw.Date = RawDate(2:end);
Raw.Time = RawTime(2:end);
KissTable = table;
Kiss.GS = KissGS(2:end);
Kiss.Sat = KissSat(2:end);
Kiss.Date = KissDate(2:end);
Kiss.Time = KissTime(2:end);
TrackTable = table;
Track.GS = TrackGS(2:end);
```

```
Track.Sat = TrackSat(2:end);
Track.Date = TrackDate(2:end);
Track.Time = TrackTime(2:end);
Track.AZ = TrackAZ(2:end);
Track.EL = TrackEL(2:end);
Track.Range = TrackRange(2:end);
Track.PassStart = TrackPassStart(2:end);
Track.PassEnd = TrackPassEnd(2:end);
Track.PassAZo = TrackPassAZo(2:end);
Track.PassMaxEl = TrackPassMaxEl(2:end);
clearvars -except dirKiss dirRaw dirTrack Raw Kiss Track

%%
% ===== 3. CORRELATE TRACK AZ/EL TO RAW/KISS TIMES/LOCATIONS =====
% Prereqs: Raw, Kiss, and Track Variables from Script 2

%%
% This script takes the imported Raw, Kiss, and Track information
% and creates Raw and Kiss structures showing times and associated
% Az/El of satellite during preamble and decodes.
clearvars -except dirKiss dirRaw dirTrack Raw Kiss Track
clc; close all
% Determine the Az/El at the Raw timestamps (Preambles)
for i = 1:length(Raw.Time);
    disp(['Current Raw File: ' num2str(i) ' of ' num2str...
        (length(Raw.Time))])
    RawGSIndex = find(Raw.GS(i)==Track.GS);
    RawSatIndex = find(Raw.Sat(i)==Track.Sat);
    RawDateIndex = find(Raw.Date(i)==Track.Date);
    RawTimeIndex = find(Raw.Time(i)==Track.Time);
    Index = intersect(intersect(intersect(RawGSIndex,RawSatIndex),...
        RawDateIndex),RawTimeIndex);
    % Accounts for missing Track Data for given Raw data
    if isempty(Index)
    else
        Raw.AZ(i,1) = Track.AZ(Index(1));
        Raw.EL(i,1) = Track.EL(Index(1));
        StartTime = datetime(Track.PassStart(Index(1)),...
            'InputFormat','HH:mm:ss');
        EndTime = datetime(Track.PassEnd(Index(1)),...
            'InputFormat','HH:mm:ss');
        Duration = seconds(EndTime-StartTime);
        % Account for passing midnight
        if Duration < 0
            Duration = Duration + 86400; % Adding 24 hours in seconds
        end
        Raw.PassDur(i,1) = string(Duration);
        Raw.PassStart(i,1) = Track.PassStart(Index(1));
        Raw.PassEnd(i,1) = Track.PassEnd(Index(1));
        Raw.PassAZo(i,1) = Track.PassAZo(Index(1));
        Raw.PassMaxEl(i,1) = Track.PassMaxEl(Index(1));
    end
end
clearvars -except dirKiss dirRaw dirTrack Raw Kiss Track
% Determine the Az/El at the Kiss timestamps (Decodes)
```

```
for i = 1:length(Kiss.Time);
    disp(['Current Kiss File: ' num2str(i) ' of ' ...
        num2str(length(Kiss.Time))])
    KissGSIndex = find(Kiss.GS(i)==Track.GS);
    KissSatIndex = find(Kiss.Sat(i)==Track.Sat);
    KissDateIndex = find(Kiss.Date(i)==Track.Date);
    KissTimeIndex = find(Kiss.Time(i)==Track.Time);
    Index = intersect(intersect(intersect(KissGSIndex,KissSatIndex),...
        KissDateIndex),KissTimeIndex);
    % Accounts for missing Track Data for given Kiss data
    if isempty(Index)
    else
        Kiss.AZ(i,1) = Track.AZ(Index(1));
        Kiss.EL(i,1) = Track.EL(Index(1));
        StartTime = datetime(Track.PassStart(Index(1))...
            ,'InputFormat','HH:mm:ss');
        EndTime = datetime(Track.PassEnd(Index(1)),...
            'InputFormat','HH:mm:ss');
        Duration = seconds(EndTime-StartTime);
        % Account for passing midnight
        if Duration < 0
            Duration = Duration + 86400; % Adding 24 hours (seconds)
        end
        Kiss.PassDur(i,1) = string(Duration);
        Kiss.PassStart(i,1) = Track.PassStart(Index(1));
        Kiss.PassEnd(i,1) = Track.PassEnd(Index(1));
        Kiss.PassAZo(i,1) = Track.PassAZo(Index(1));
        Kiss.PassMaxEl(i,1) = Track.PassMaxEl(Index(1));
    end
end
% Delete the rows where Az/El could not be found
for i=length(Raw.AZ):-1:1
    if ismissing(Raw.AZ(i))==1 || ismissing(Raw.AZ(i))==1
        Raw.GS(i)=[];
        Raw.Sat(i)=[];
        Raw.Date(i)=[];
        Raw.Time(i)=[];
        Raw.AZ(i)=[];
        Raw.EL(i)=[];
        Raw.PassDur(i)=[];
        Raw.PassStart(i)=[];
        Raw.PassEnd(i)=[];
        Raw.PassAZo(i)=[];
        Raw.PassMaxEl(i)=[];
    end
end
for i=length(Kiss.AZ):-1:1
    if ismissing(Kiss.AZ(i))==1 || ismissing(Kiss.AZ(i))==1
        Kiss.GS(i)=[];
        Kiss.Sat(i)=[];
        Kiss.Date(i)=[];
        Kiss.Time(i)=[];
        Kiss.AZ(i)=[];
        Kiss.EL(i)=[];
        Kiss.PassDur(i)=[];
```

```
        Kiss.PassStart(i)=[];
        Kiss.PassEnd(i)=[];
        Kiss.PassAZo(i)=[];
        Kiss.PassMaxEl(i)=[];
    end
end
clearvars -except dirKiss dirRaw dirTrack Raw Kiss Track
save('FilesLog/KissTrack','Kiss')
save('FilesLog/RawTrack','Raw')

%%
% ===== 4. MONO-COLOR RAW/KISS PLOT =====
% Prereqs: KissTrack and RawTrack Files

%%
clear; clc; close all
% Load Raw and Kiss Data (GS, Sat, Date, Time, Az, El)
load('FilesLog/KissTrack')
load('FilesLog/RawTrack')
% User selects Ground Station name
UserGS = input...
    ('\nAFIT / HSFL / PTSUR / SDL / UNM\nSelect Ground Station: ',...
    's');
% User selects satellite name
UserSat = input...
    ('\n13FLORA / 10MERRYW / FAUNA\nSelect Satellite: ','s');
% Prepare pass data for plotting
Count=1;
for i=1:length(Raw.AZ);
    if Raw.GS(i)==UserGS;
        if Raw.Sat(i)==UserSat;
            % ismissing accounts for missing track entries in data
            if ismissing(Raw.AZ(i)) == 0;
                if ismissing(Raw.EL(i)) == 0;
                    PlotRawAZ(Count) = str2num(char(Raw.AZ(i)));
                    PlotRawEL(Count) = str2num(char(Raw.EL(i)));
                    Count=Count+1;
                end
            end
        end
    end
end
Count=1;
for i=1:length(Kiss.AZ);
    if Kiss.GS(i)==UserGS;
        if Kiss.Sat(i)==UserSat;
            % ismissing accounts for missing track entries in data
            if ismissing(Kiss.AZ(i)) == 0;
                if ismissing(Kiss.EL(i)) == 0;
                    PlotKissAZ(Count) = str2num(char(Kiss.AZ(i)));
                    PlotKissEL(Count) = str2num(char(Kiss.EL(i)));
                    Count=Count+1;
                end
            end
        end
```

```
    end
end
% Plot the pass with successful decodes and preambles
% Create polar plot shell for data input with elevation tick
% marks from GS to horizon (0 - 90 degrees)
El = 0:10:90;
ElTick = El/90;
theta = linspace(0,2*pi);
polarplot(theta,ElTick(end)+zeros(size(theta)),'k')
thetaticks(0:20:340)
pax = gca;
pax.FontSize = 10;
pax.GridLineStyle = '-';
pax.ThetaDir = 'clockwise';
pax.ThetaZeroLocation = 'top';
set(gcf,'color','w');
rlim([ElTick(1) ElTick(end)])
rticks([ElTick(1) ElTick(3) ElTick(5) ElTick(7) ElTick(9)])
rticklabels({'\fontsize{6}90\circ','\fontsize{6}70\circ',...
    '\fontsize{6}50\circ','\fontsize{6}30\circ',...
    '\fontsize{6}10\circ'})
thetaticklabels({'\fontsize{8}000\circ','\fontsize{8}020\circ',...
    '\fontsize{8}040\circ','\fontsize{8}060\circ',...
    '\fontsize{8}080\circ','\fontsize{8}100\circ',...
    '\fontsize{8}120\circ','\fontsize{8}140\circ',...
    '\fontsize{8}160\circ','\fontsize{8}180\circ',...
    '\fontsize{8}200\circ','\fontsize{8}220\circ',...
    '\fontsize{8}240\circ','\fontsize{8}260\circ',...
    '\fontsize{8}280\circ','\fontsize{8}300\circ',...
    '\fontsize{8}320\circ','\fontsize{8}340\circ'})
title(sprintf('%s - %s\nPreambles/Decodes', UserGS,...
    UserSat),'FontSize',14);
hold on
SatRawEl = 1-PlotRawEL/90;
SatRawAz = PlotRawAZ*pi/180;
polarplot(SatRawAz,SatRawEl,'og','MarkerSize',2,...
    'MarkerFaceColor','g')
SatKissEl = 1-PlotKissEL/90;
SatKissAz = PlotKissAZ*pi/180;
polarplot(SatKissAz,SatKissEl,'or','MarkerSize',2,...
    'MarkerFaceColor','r')

%%
% ===== 5. BINS UNIQUE SAT/GS PAIR FOR HEAT MAP PLOTTING =====
% Prereqs: KissTrack File

%%
clear; clc; close all
% Load Kiss Data (GS, Sat, Date, Time, Az, El...)
load('FilesLog/KissTrack')
% Compile the individual fields in the structures to a single array
KissComp = [Kiss.GS,Kiss.Sat,Kiss.Date,Kiss.Time,Kiss.AZ,Kiss.EL,...
    Kiss.PassDur,Kiss.PassStart,Kiss.PassEnd,Kiss.PassAZo];
% User selects Ground Station name
UserGS = input...
```

```
    ('\nAFIT / HSFL / PTSUR / SDL / UNM\nSelect Ground Station: '...
    ,'s');
% User selects satellite name
UserSat = input...
    ('\n13FLORA / 10MERRYW / FAUNA\nSelect Satellite: ','s');
% Apply User-selected GS and Satellite and extract specific data
Count=1;
for i=1:size(KissComp,1);
    if KissComp(i,1)==UserGS;
        if KissComp(i,2)==UserSat;
            if ismissing(KissComp(i,5))==0
                KissUser(Count,1) = KissComp(i,1);
                KissUser(Count,2) = KissComp(i,2);
                KissUser(Count,3) = KissComp(i,3);
                KissUser(Count,4) = KissComp(i,4);
                KissUser(Count,5) = KissComp(i,5);
                KissUser(Count,6) = KissComp(i,6);
                KissUser(Count,7) = KissComp(i,7);
                KissUser(Count,8) = KissComp(i,8);
                KissUser(Count,9) = KissComp(i,9);
                KissUser(Count,10) = KissComp(i,10);
                Count=Count+1;
            end
        end
    end
end
% Column 1 = Ground Station
% Column 2 = Satellite
% Column 3 = Preamble/Decode Date
% Column 4 = Preamble/Decode Time
% Column 5 = Preamble/Decode Azimuth
% Column 6 = Preamble/Decode Elevation
% Column 7 = Total Pass Duration
% Column 8 = Pass Start
% Column 9 = Pass End
% Column 10 = Pass AZo
% Sort rows by Date/Time Column (Should be already sorted...)
KissUser = sortrows(KissUser,[3 4]);
% Create 11th column that labels distinct passes in order
% of occurence.
% If decodes/preambles are more than 30 min (1800 sec)
% apart = Next pass
Count=1;
KissUser(1,11) = 1;
for i=1:(size(KissUser,1)-1)
    datetime1 = datetime(strcat(KissUser(i,3),KissUser(i,4)),...
        'InputFormat','MM/dd/yyHH:mm:ss');
    datetime2 = datetime(strcat(KissUser(i+1,3),KissUser(i+1,4)),...
        'InputFormat','MM/dd/yyHH:mm:ss');
    if abs(seconds(datetime2-datetime1))<1800
        KissUser(i+1,11) = Count;
    else
        Count = Count+1;
        KissUser(i+1,11) = Count;
    end
```

```
end
% Create 12th column to characterize Ascending and Descending passes
% Create 13th column for elapsed time of decode/preamble
% since pass start
for i=1:max(str2num(char(KissUser(:,11))))
    PassIndex = find(KissUser(:,11)==num2str(i));
    StartIndex = min(PassIndex);
    PassAZo = str2num(char(KissUser(StartIndex,10)));
    if PassAZo < 90;
        PassAZo = PassAZo + 360;
    end
    if PassAZo < 270;
        KissUser(PassIndex,12) = 'Asc';
    else
        KissUser(PassIndex,12) = 'Des';
    end
    for j=1:size(KissUser,1);
        if KissUser(j,11) == num2str(i);
        EventTime = datetime(KissUser(j,4),'InputFormat'...
            ,'HH:mm:ss');
        PassStartTime = datetime(KissUser(j,8),'InputFormat'...
            ,'HH:mm:ss');
        ElapsedTime = seconds(EventTime - PassStartTime);
        % Account for passing midnight
        if ElapsedTime < 0
            ElapsedTime = ElapsedTime + 86400;
            % Adding 24 hours in seconds
        end
        KissUser(j,13) = ElapsedTime;
        end
    end
end
% Bin times for each pass into 10 equal intervals
% (0-10%, 10-20%, etc...)
% Create Column 14 for Bin Number on Specific Pass
% Create Column 15 for Total Preamb/Decodes in Bin (Within Pass)
% Create Column 16 for Preamb/Decode per sec in Bin (Within Pass)
% Create Column 17 for Time interval of specific pass bin
TimeBins = 10;
for i=1:max(str2num(char(KissUser(:,11))))
    PassIndex = find(KissUser(:,11)==num2str(i));
    PassDur = str2num(char(KissUser(PassIndex(1),7)));
    edges = linspace(0,PassDur,TimeBins+1);
    [N,edges,bin] = histcounts(str2num(char(KissUser...
        (PassIndex,13))),edges);
    KissUser(PassIndex,14) = bin;
    for j=1:TimeBins;
        BinIndex = PassIndex(find(KissUser(PassIndex,14)...
            ==num2str(j)));
        if isempty(BinIndex)==0;  % Skips empty bins
            KissUser(BinIndex,15) = N(j);
            KissUser(BinIndex,16) = N(j)/edges(2);
            KissUser(BinIndex,17) = edges(2);
        end
    end
```

```
end
% Delete all remaining rows that have missing data
for i=size(KissUser,1):-1:1
    if ismissing(KissUser(i,16))==1
        KissUser(i,:)=[];
    end
end
% Bin AZo for each pass into 10 degree increments
% Create Column 18 for AZo bins
SpanAZo = 10;
AZoBins = 360/SpanAZo;
edges = linspace(0,360,AZoBins+1);
[N,edges,bin] = histcounts(str2num(char(KissUser(:,10))),edges);
KissUser(:,18) = bin;
% clearvars -except Kiss Raw KissUser RawUser UserGS UserSat a
for i=1:TimeBins
    for j=1:AZoBins
        Index = intersect(find(str2num(char(KissUser(:,14)))==i),...
            find(str2num(char(KissUser(:,18)))==j));
        TotalDec = length(Index);
        % Create 19th Column Total Decodes for Time/AZo Bin Intersect
        KissUser(Index,19) = TotalDec;
    end
end
% Create a Column 20 to represent colors based on Total Decode Values
% Yellow to Red (autumn)
a = flipud((autumn(10)));
%Normalize colors to range of values (To next highest 50)
MaxValue = ceil(max(str2num(char(KissUser(:,19))))/50)*50;
ColorRange = linspace(0,MaxValue,10);
% Assign colors to Values
for i=1:size(KissUser(:,19),1)
    KissValue = str2num(char(KissUser(i,19)));
    if KissValue==ColorRange(1)
        KissUser(i,20) = num2str(a(1,:));
    elseif KissValue>ColorRange(1) && KissValue<=ColorRange(2)
        KissUser(i,20) = num2str(a(2,:));
    elseif KissValue>ColorRange(2) && KissValue<=ColorRange(3)
        KissUser(i,20) = num2str(a(3,:));
    elseif KissValue>ColorRange(3) && KissValue<=ColorRange(4)
        KissUser(i,20) = num2str(a(4,:));
    elseif KissValue>ColorRange(4) && KissValue<=ColorRange(5)
        KissUser(i,20) = num2str(a(5,:));
    elseif KissValue>ColorRange(5) && KissValue<=ColorRange(6)
        KissUser(i,20) = num2str(a(6,:));
    elseif KissValue>ColorRange(6) && KissValue<=ColorRange(7)
        KissUser(i,20) = num2str(a(7,:));
    elseif KissValue>ColorRange(7) && KissValue<=ColorRange(8)
        KissUser(i,20) = num2str(a(8,:));
    elseif KissValue>ColorRange(8) && KissValue<=ColorRange(9)
        KissUser(i,20) = num2str(a(9,:));
    elseif KissValue>ColorRange(9)
        KissUser(i,20) = num2str(a(10,:));
    end
end
```

```
%%
% ===== 6. PLOTS ASC AND DES HEAT MAP OF SAT/GS PAIR KISS =====
% Prereqs: Unique GS/Sat data from Script 5

%%
% Prepare pass data for plotting
CountA=1;
CountD=1;
for i=1:size(KissUser,1);
    % ismissing accounts for missing track entries in data
    if ismissing(KissUser(i,5)) == 0;
        if ismissing(KissUser(i,6)) == 0;
            if KissUser(i,12)=='Asc'
                PlotKissAscAZ(CountA) = str2num(char(KissUser(i,5)));
                PlotKissAscEL(CountA) = str2num(char(KissUser(i,6)));
                PlotKissAscCol(CountA) = KissUser(i,20);
                CountA=CountA+1;
            end
            if KissUser(i,12)=='Des'
                PlotKissDesAZ(CountD) = str2num(char(KissUser(i,5)));
                PlotKissDesEL(CountD) = str2num(char(KissUser(i,6)));
                PlotKissDesCol(CountD) = KissUser(i,20);
                CountD=CountD+1;
            end
        end
    end
end
% Plot the passes using different colors for values
% Create polar plot shell for data input with
% elevation tick marks from GS to horizon (0 - 90 degrees)
figure
El = 0:10:90;
ElTick = El/90;
theta = linspace(0,2*pi);
polarplot(theta,ElTick(end)+zeros(size(theta)),'k')
thetaticks(0:20:340)
pax = gca;
pax.FontSize = 10;
pax.GridLineStyle = '-';
pax.ThetaDir = 'clockwise';
pax.ThetaZeroLocation = 'top';
set(gcf,'color','w');
rlim([ElTick(1) ElTick(end)])
rticks([ElTick(1) ElTick(3) ElTick(5) ElTick(7) ElTick(9)])
rticklabels({'\fontsize{6}90\circ','\fontsize{6}70\circ',...
    '\fontsize{6}50\circ','\fontsize{6}30\circ',...
    '\fontsize{6}10\circ'})
thetaticklabels({'\fontsize{8}000\circ','\fontsize{8}020\circ',...
    '\fontsize{8}040\circ','\fontsize{8}060\circ',...
    '\fontsize{8}080\circ','\fontsize{8}100\circ',...
    '\fontsize{8}120\circ','\fontsize{8}140\circ',...
    '\fontsize{8}160\circ','\fontsize{8}180\circ',...
    '\fontsize{8}200\circ','\fontsize{8}220\circ',...
    '\fontsize{8}240\circ','\fontsize{8}260\circ',...
```

```
    '\fontsize{8}280\circ','\fontsize{8}300\circ',...
    '\fontsize{8}320\circ','\fontsize{8}340\circ'})
title(sprintf('%s - %s\nAscending Pass Decodes',...
    UserGS, UserSat),'FontSize',14);
hold on
for i=1:length(PlotKissAscEL)
    SatKissAscElL = 1-PlotKissAscEL(i)/90;
    SatKissAscAz = PlotKissAscAZ(i)*pi/180;
    h = polarplot(SatKissAscAz,SatKissAscElL,'ok',...
        'MarkerSize',1.5);
    h.Color = str2num(char(PlotKissAscCol(i)));
    h.MarkerFaceColor = str2num(char(PlotKissAscCol(i)));
end
colormap(a);
c =colorbar('Location','southoutside');
c.Label.String = 'Total Decodes';
caxis([0 MaxValue])
hold off

figure
polarplot(theta,ElTick(end)+zeros(size(theta)),'k')
thetaticks(0:20:340)
pax = gca;
pax.FontSize = 10;
pax.GridLineStyle = '-';
pax.ThetaDir = 'clockwise';
pax.ThetaZeroLocation = 'top';
set(gcf,'color','w');
rlim([ElTick(1) ElTick(end)])
rticks([ElTick(1) ElTick(3) ElTick(5) ElTick(7) ElTick(9)])
rticklabels({'\fontsize{6}90\circ','\fontsize{6}70\circ',...
    '\fontsize{6}50\circ','\fontsize{6}30\circ',...
    '\fontsize{6}10\circ'})
thetaticklabels({'\fontsize{8}000\circ','\fontsize{8}020\circ',...
    '\fontsize{8}040\circ','\fontsize{8}060\circ',...
    '\fontsize{8}080\circ','\fontsize{8}100\circ',...
    '\fontsize{8}120\circ','\fontsize{8}140\circ',...
    '\fontsize{8}160\circ','\fontsize{8}180\circ',...
    '\fontsize{8}200\circ','\fontsize{8}220\circ',...
    '\fontsize{8}240\circ','\fontsize{8}260\circ',...
    '\fontsize{8}280\circ','\fontsize{8}300\circ',...
    '\fontsize{8}320\circ','\fontsize{8}340\circ'})
title(sprintf('%s - %s\nDescending Pass Decodes'...
    , UserGS, UserSat),'FontSize',14);
hold on
for i=1:length(PlotKissDesEL)
    SatKissDesElL = 1-PlotKissDesEL(i)/90;
    SatKissDesAz = PlotKissDesAZ(i)*pi/180;
    h = polarplot(SatKissDesAz,SatKissDesElL,'ok'...
        ,'MarkerSize',1.5);
    h.Color = str2num(char(PlotKissDesCol(i)));
    h.MarkerFaceColor = str2num(char(PlotKissDesCol(i)));
end
colormap(a);
c =colorbar('Location','southoutside');
```

```
c.Label.String = 'Total Decodes';
caxis([0 MaxValue])

%%
% ===== 7. KISS AZO/TIME SEGMENT HEAT MAPS (ACTUAL DATA) =====
% Prereqs: Unique GS/Sat data from Script 5

%%
clearvars -except AZoTimeArray AZoBins TimeBins KissUser...
    UserGS UserSat
% Create Array showing the decodes/preambles per second based
% on the initial azimuth range and ratio of time into pass
AZoTimeArray = NaN(AZoBins,TimeBins);
% Find indices where AZo bins intersect Time bins
for i=1:AZoBins
    AZoBinIndex = find(str2num(char(KissUser(:,18)))==i);
    % If no index, all columns in that row are 0
    if isempty(AZoBinIndex)==1;
        AZoTimeArray(i,1:end)=0;
    else
        for j=1:TimeBins
            TimeBinIndex = find(str2num(char(KissUser(:,14)))==j);
            % If no index, value for individual cell is 0
            if isempty(TimeBinIndex)==1;
                AZoTimeArray(i,j)=0;
            else
                % If no index (intersect), value for indiv cell = 0
                TimeAZoBinIndex = intersect(AZoBinIndex,TimeBinIndex);
                if isempty(TimeAZoBinIndex)==1;
                    AZoTimeArray(i,j)=0;
                else
                    % IndexDecPre gives index locations of total dec
                    % for pass segments within the AZo and Time Bins
                    IndexDecPre = KissUser([TimeAZoBinIndex(...
                        find(diff(str2num(char(KissUser...
                        (TimeAZoBinIndex,11))))>0));...
                        TimeAZoBinIndex(end)],15);
                    % IndexTime gives index location of time
                    % for pass segments within the AZo and Time Bins
                    IndexTime = KissUser([TimeAZoBinIndex...
                        (find(diff(str2num(char(KissUser...
                        (TimeAZoBinIndex,11))))>0));...
                         TimeAZoBinIndex(end)],17);
                    % AZoTimeArray is total decodes / measured time
                    % Gives Decodes/Second for each AZo/Time Segment
                    AZoTimeArray(i,j) = sum(str2num(char(...
                        IndexDecPre)))/sum(str2num(char(IndexTime)));
                end
            end
        end
    end
end
% Shifts Values so plot is centered on 270 deg Initial Azimuth
AZoTimeArrayPlot = [AZoTimeArray(10:36,:) ; AZoTimeArray(1:9,:)];
contourf(AZoTimeArrayPlot,'LineColor','none')
```

```
a=colormap(flipud(autumn(256)));
a(1,:)=1;
colormap(a)
colorbar
% Max Value in array and rounds up to next 0.02 for scale
MaxVal = max(AZoTimeArrayPlot(:));
MaxVal = ceil(MaxVal*50)/50;
ColorRange = [0:0.04:MaxVal];
c = colorbar('Ticks',ColorRange);
c.Label.String = 'Decodes per Second';
caxis([0 MaxVal])
xlim([1 10])
xticks([1:0.45:10])
xticklabels({'0%','','10%','','20%','','30%','','40%',...
    '','50%','','60%','','70%','','80%','','90%','','100%'});
YTickVal = 0.5:1:AZoBins+0.5;
yticks([YTickVal])
yticklabels({'090\circ','100\circ','110\circ','120\circ',...
    '130\circ','140\circ','150\circ','160\circ','170\circ',...
    '180\circ','190\circ','200\circ','210\circ','220\circ',...
    '230\circ','240\circ','250\circ','260\circ','270\circ',...
    '280\circ','290\circ','300\circ','310\circ','320\circ',...
    '330\circ','340\circ','350\circ','000\circ','010\circ',...
    '020\circ','030\circ','040\circ','050\circ','060\circ',...
    '070\circ','080\circ','090\circ'});
ylim([0 AZoBins])
set(gcf,'color','w');
xlabel('Percent of Total Pass Time','FontSize',13);
ylabel('Initial Azimuth','FontSize',13);
title(sprintf('Pass Decodes - %s - %s', UserGS, UserSat),...
    'FontSize',14);


%%
% ===== 8. KISS AZO/TIME SEGMENT HEAT MAPS (THEORETICAL) =====
% Prereqs: GSdata variable from Appendix C

%%
% This script extracts range data for various points along a track
% separated by the AZo and Time within pass (%)
close all; clearvars -except GSdata; clc
% User selects Ground Station name
UserGS = input...
    ('\nAFIT / HSFL / PTSUR / SDL / UNM\nSelect Ground Station: '...
    ,'s');
% User selects satellite name
UserSat = input...
    ('\n13FLORA / 10MERRYW / FAUNA\nSelect Satellite: ','s');
% Create listing of all Track Files
dirTrack = dir('FilesTrack\*.dat');
 % For Loop to extract data for specific Satellite and GS
Count = 1
for i=1:size(dirTrack,1);
    disp(['Current Track File: ' num2str(i) ' of '...
        num2str(size(dirTrack,1))])
    filename = ['FilesTrack\' dirTrack(i).name];
```

```
    Sat = extractBefore(extractAfter(extractAfter(extractAfter...
        (filename,'_'),'_'),'_'),'_');
    GS = extractBefore(extractAfter(extractAfter(filename,...
        '_'),'_'),'_');
    if strcmp(UserSat,Sat)
        if strcmp(UserGS,GS)
            % Initialize variables, format each line, open text file
            delimiter = ' ';
            formatSpec = '%11s%19s%7f%6f%8f%s%[^\n\r]';
            fileID = fopen(filename,'r');
            % Read columns of data
            dataArray = textscan(fileID, formatSpec,...
                'Delimiter', '','WhiteSpace', '', 'TextType',...
                'string', 'ReturnOnError', false);
            % Remove white space, close file, create output variables
            dataArray{2} = strtrim(dataArray{2});
            fclose(fileID);
            % Column 1 = Date
            % Column 2 = Time in Pass
            % Column 3 = Azimuth
            % Column 4 = Elevation
            % Column 5 = Range
            AZo = dataArray{3}(1);
            El = dataArray{4}(1);
            Time10Perc = length(dataArray{2})/10;
            % Sort information for Time Percent and AZo
            for j=1:10
                TimeInd = round([(j-1)*Time10Perc+1 (j)*Time10Perc]);
                Time = j;
                Range = mean(dataArray{5}(TimeInd(1):TimeInd(2)));
                AZoTimeRange(Count,:) = [AZo,Time,Range];
                Count = Count + 1;
            end
         end
    end
end
% Create Array with Time Percent (Columns) vs Initial Az (Rows)
for i=1:36
    AZoIndex = intersect(find(AZoTimeRange(:,1)>((i-1)*10)),...
        find(AZoTimeRange(:,1)<(i*10)));
    for j=1:10
        if isempty(AZoIndex)==1;
            AZoTimeArray(i,j) = inf;
        else
            TimeIndex = find(AZoTimeRange(:,2)==j);
            AZoTimeIndex = intersect(AZoIndex,TimeIndex);
            AZoTimeArray(i,j) = mean(AZoTimeRange(AZoTimeIndex,3));
        end
    end
end

%
% USE THIS SECTION FOR 40 DEGREE ACCEPTANCE REGION
% Extract data for User Sat/GS selection
Count=1;
```

```
for i=1:length(GSdata.AZo);
    if GSdata.gs_name(i)==UserGS
        if GSdata.sat_name(i)==UserSat
            AZoGS(Count)=GSdata.AZo(i);
            ELGS(Count)=GSdata.max_EL(i);
            Count=Count+1;
        end
    end
end
% Determine Acceptance Region for Asc/Des Passes with Max El of at
% least 40 degrees (Rounded to 5 or 0)
EL40 = intersect(find(ELGS>40),find(ELGS<45));
AscHi = 10*ceil(max(AZoGS(EL40(find((AZoGS(EL40))<270))))/10);
AscLo = 10*floor(min(AZoGS(EL40))/10);
AcsNad = (AscHi+AscLo)/2;
DesHi = 10*ceil(max(AZoGS(EL40))/10);
DesLo = 10*floor(min(AZoGS(EL40(find((AZoGS(EL40))>270))))/10);
DesNad = (DesHi+DesLo)/2;
% Establish AZoTimeArray for later countorf
AZoTimeArray40 = AZoTimeArray;
for i=1:length(AZoTimeArray40)
    if i<AscLo/10
        AZoTimeArray40(i,:)=inf;
    elseif (i>AscHi/10) && (i<DesLo/10)
        AZoTimeArray40(i,:)=inf;
    elseif (i>DesHi/10)
        AZoTimeArray40(i,:)=inf;
    end
end
% Eliminates any "Track Data Gaps" appearing as NaNs
AZoTimeArray40 = fillmissing(AZoTimeArray40,'linear');
AZoTimeArray40 = -10*log10((AZoTimeArray40.^2)/(500^2));
% Changes Negative inf values and replaces with min value
AZoTimeArray40(find(isinf(AZoTimeArray40))) = min...
    (AZoTimeArray40(find(isfinite(AZoTimeArray40))));
% Shifts Values so plot is centered on 270 deg Initial Azimuth
AZoTimeArray40Plot = [AZoTimeArray40(10:36,:) ; ...
    AZoTimeArray40(1:9,:)];
contourf(AZoTimeArray40Plot,'LineStyle',':','LineColor','none')
a=colormap(flipud(autumn(12)));
%a(1:2,:)=1;
a(1,:)=1;
colormap(a)
c = colorbar('Ticks',[-12:1:0]);
c.Label.String = 'Loss vs 500 km (dB)';
caxis([-12 0])
set(gcf,'color','w');
xlim([1 10])
xticks([1:0.45:10])
xticklabels({'0%','','10%','','20%','','30%','','40%','',...
    '50%','','60%','','70%','','80%','','90%','','100%'});
YTickVal = 0.5:1:36+0.5;
yticks([YTickVal])
yticklabels({'090\circ','100\circ','110\circ','120\circ',...
    '130\circ','140\circ','150\circ','160\circ','170\circ',...
```

```
    '180\circ','190\circ','200\circ','210\circ','220\circ',...
    '230\circ','240\circ','250\circ','260\circ','270\circ',...
    '280\circ','290\circ','300\circ','310\circ','320\circ',...
    '330\circ','340\circ','350\circ','000\circ','010\circ',...
    '020\circ','030\circ','040\circ','050\circ','060\circ',...
    '070\circ','080\circ','090\circ'});
ylim([0 36])
xlabel('Percent of Total Pass Time','FontSize',13);
ylabel('Initial Azimuth','FontSize',13);
title(sprintf('Expected Free Space Loss - %s - %s', ...
    UserGS, UserSat),'FontSize',14);
% Eliminates any "Track Data Gaps" appearing as NaNs
AZoTimeArray = fillmissing(AZoTimeArray,'linear');
AZoTimeArray = -10*log10((AZoTimeArray.^2)/(500^2));
% Changes Negative inf values and replaces with min value
AZoTimeArray(find(isinf(AZoTimeArray))) = min(AZoTimeArray...
    (find(isfinite(AZoTimeArray))));
% Shifts Values so plot is centered on 270 deg Initial Azimuth
AZoTimeArrayPlot = [AZoTimeArray(10:36,:) ; AZoTimeArray(1:9,:)];
figure
contourf(AZoTimeArrayPlot,'LineStyle',':','LineColor','none')
a=colormap(flipud(autumn(12)));
a(1,:)=1;
colormap(a)
c = colorbar('Ticks',[-12:1:0]);
c.Label.String = 'Loss vs 500 km (dB)';
caxis([-12 0])
set(gcf,'color','w');
xlim([1 10])
xticks([1:0.45:10])
xticklabels({'0%','','10%','','20%','','30%','','40%','',...
    '50%','','60%','','70%','','80%','','90%','','100%'});
YTickVal = 0.5:1:36+0.5;
yticks([YTickVal])
yticklabels({'090\circ','100\circ','110\circ','120\circ',...
    '130\circ','140\circ','150\circ','160\circ','170\circ',...
    '180\circ','190\circ','200\circ','210\circ','220\circ',...
    '230\circ','240\circ','250\circ','260\circ','270\circ',...
    '280\circ','290\circ','300\circ','310\circ','320\circ',...
    '330\circ','340\circ','350\circ','000\circ','010\circ',...
    '020\circ','030\circ','040\circ','050\circ','060\circ',...
    '070\circ','080\circ','090\circ'});
ylim([0 36])
xlabel('Percent of Total Pass Time','FontSize',13);
ylabel('Initial Azimuth','FontSize',13);
title(sprintf('Expected Free Space Loss - %s - %s', ...
    UserGS, UserSat),'FontSize',14);
```

LIST OF REFERENCES

[1] G. Minelli, M. Karpenko, I.M. Ross, and J. Newman, “Autonomous Operations of Large-Scale Satellite Constellations and Ground Station Networks,” presented at AAS/AIAA Astrodynamics Specialist Conf., Stevenson, WA, USA, 2017.

[2] S. Spangelo, “Modeling and Optimizing Space Networks for Improved Communications Capacity,” Ph.D. dissertation, Dept. of Aerospace Eng., University of Michigan, Ann Arbor, MI, USA, 2013.

[3] O. Popescu, “Power Budgets for CubeSat Radios to Support Ground Communications and Inter-Satellite Links,” *IEEE Access*, vol. 5, pp. 12618–12625, Jun. 2017. [Online]. doi: 10.1109/ACCESS.2017.2721948.

[4] L. Mehnen and B. Preindl, “A Distributed Orbital Measurement Instrument for Link Quality Assurance,” in *IFAC Proc.*, vol. 45, no. 7, pp. 232–237, 2012. [Online]. Available: doi:10.3182/20120523-3-CZ-3015.00045.

[5] S. Spangelo, D. Boone, and J. Cutler, “Assessing the Capacity of a Federated Ground Station,” in *2010 IEEE Aerospace Conf. Proc.*, March 2010. [Online]. Available: doi:10.1109/AERO.2010.5446950.

[6] J. Straub, “Reducing Link Budget Requirements with Model-Based Transmission Reduction Techniques,” presented at AIAA/USU Conf. on Small Satellites, Logan, UT, USA, 2014.

[7] M. Sorgenfrei, M. Nehrenz, and K. Shish, “Operational Considerations for a Swarm of CubeSat-Class Spacecraft,” presented at AIAA Space Operations Conf., Pasadena, CA, USA, 2014.

[8] A. J. Writt, “Optimization of CubeSat Ground Stations for Increased Satellite Numbers,” M.S. thesis, Space Syst. Academic Group, NPS, Monterey, CA, USA, 2018.

[9] I. Vertat, R. Linhart, M. Pokorny, and T. Kavalir, “Signal Quality Evaluation for Picosatellite Communication Systems,” *2012 Int. Conf. on Appl. Electron*., 2012, pp. 331–334.

[10] G. Minelli, “Resource-Constrained Autonomous Operations of Satellite Constellations and Ground Station Networks,” Ph.D. dissertation, Space Syst. Academic Group, NPS, Monterey, CA, USA, 2018.

[11] H. D. Curtis, *Orbital Mechanics for Engineering Students*. Waltham, MA, USA: Elsevier Ltd., 2014.

[12] J. M. Roehrig, "Development of a Versatile Groundstation Utilizing Software Defined Radio," M.S. thesis, Space Syst. Academic Group, NPS, Monterey, CA, USA, 2016.

[13] Federal Communications Commission, "Comprehensive Review of Licensing and Operating Rules for Satellite Services," Washington, DC, USA, 2016. [Online]. Available: https://www.gpo.gov/fdsys/pkg/FR-2016-08-18/pdf/2016-14800.pdf

[14] C. R. Le Gaux, "STARE CubeSat Communications Testing, Simulation, and Analysis," M.S. thesis, Space Syst. Academic Group, NPS, Monterey, CA, USA, 2012.

[15] B. Sklar, *Digital Communications Fundamentals and Applications*. Upper Saddle River, NJ, USA. Prentice Hall, 2017.

[16] Google. "Naval Postgraduate School (NPS/PTSUR)." [Online] Available: https://www.google.com/maps/@36.5952546,-121.8760605,119m/data=!3m1!1e3. Accessed Apr. 20, 2018.

[17] Google. "University of New Mexico (UNM)." [Online] Available: https://www.google.com/maps/@35.0697015,-106.598869,21082m/data=!3m1!1e3. Accessed Apr. 20, 2018.

[18] R. C. Griffith, "Mobile CubeSat Command and Control (MC3)," M.S. thesis, Space Syst. Academic Group, NPS, Monterey, CA, USA, 2011.

[19] G. C. Morrison, "Mobile CubeSat Command and Control Assembly and Lessons Learned," M.S. thesis, Space Syst. Academic Group, NPS, Monterey, CA, USA, 2011.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California