NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

DESIGN OF A PARALLEL EDUCATIONAL SATELLITE GROUND STATION

by

Bradley R. Stephens

June 2022

Thesis Advisor:	Giovanni Minelli
Co-Advisors:	Noah Weitz
	Jennifer Hudson

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE				*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC, 20503.				
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** June 2022	**3. REPORT TYPE AND DATES COVERED** Master's thesis		
4. TITLE AND SUBTITLE DESIGN OF A PARALLEL EDUCATIONAL SATELLITE GROUND STATION				**5. FUNDING NUMBERS**
6. AUTHOR(S) Bradley R. Stephens				
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 Mobile CubeSat Command and Control (MC3) ground station network provides communications infrastructure for small satellites developed by U.S. government organizations, contractors, and educational institutions. As the network has matured, so too have the cybersecurity requirements that govern its operations. By implementing tight configuration constraints on software, hardware, and networking, ground stations are unable to utilize the equipment outside standard operations. This poses a particular problem for educational institutions that routinely involve such equipment in their curricula for hands-on instruction and research. This research effort focuses on design, implementation, and testing of a parallel ground station to MC3 that allows educational institutions the freedom to innovate and perform research. The parallel station will share the MC3 antenna, the most valuable component of the ground station, but provide a separate rack of equipment that functions as a separate ground station. This research applies directly to those institutions that are currently members of the MC3 system. Currently, the Naval Postgraduate School, the United States Coast Guard Academy, and the United States Naval Academy have expressed interest in a parallel ground station. The resulting approach benefits these educational institutions by removing barriers of development and increasing instructional impact in the ever-competitive worldwide aerospace industry.

14. SUBJECT TERMS educational ground station, satellite ground station, Mobile CubeSat Command and Control, MC3, satellite communications, smallsat, CubeSat, network, flexible, educational, research and development			**15. NUMBER OF PAGES** 101
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

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

THIS PAGE INTENTIONALLY LEFT BLANK

DESIGN OF A PARALLEL EDUCATIONAL SATELLITE GROUND STATION

Bradley R. Stephens
Ensign, United States Navy
BS, United States Naval Academy, 2021

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ASTRONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 2022

Approved by: Giovanni Minelli
Advisor

Noah Weitz
Co-Advisor

Jennifer Hudson
Co-Advisor

Garth V. Hobson
Chair, Department of Mechanical and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The Mobile CubeSat Command and Control (MC3) ground station network provides communications infrastructure for small satellites developed by U.S. government organizations, contractors, and educational institutions. As the network has matured, so too have the cybersecurity requirements that govern its operations. By implementing tight configuration constraints on software, hardware, and networking, ground stations are unable to utilize the equipment outside standard operations. This poses a particular problem for educational institutions that routinely involve such equipment in their curricula for hands-on instruction and research. This research effort focuses on design, implementation, and testing of a parallel ground station to MC3 that allows educational institutions the freedom to innovate and perform research. The parallel station will share the MC3 antenna, the most valuable component of the ground station, but provide a separate rack of equipment that functions as a separate ground station. This research applies directly to those institutions that are currently members of the MC3 system. Currently, the Naval Postgraduate School, the United States Coast Guard Academy, and the United States Naval Academy have expressed interest in a parallel ground station. The resulting approach benefits these educational institutions by removing barriers of development and increasing instructional impact in the ever-competitive worldwide aerospace industry.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGEMENTS

I would first like to thank Dr. Giovanni Minelli and Noah Weitz for their incredible dedication and assistance in this research effort. Whether it be doing satellite tracks at 0700 or making writing edits at 2300 to make a deadline, they were always available to help out. I am very grateful for their direction. Thank you to Dr. Jennifer Hudson for stepping up as a co-advisor when I was in need. Thank you to the Coast Guard Academy, their MC3 faculty, and cadets (specifically Cadet 1/C Rebecca Emerson) for providing their valued input and interest in this research effort. Lastly, I would like to thank my family and friends for their support throughout this process!

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ACU	Antenna Control Unit
AFRL	Air Force Research Laboratory
ASL	Authentication Servers List
AUS	Authentication Server
AWS	Amazon Web Services
Az/El	Azimuth and Elevation
BWG	Beam Waveguide
COTS	Commercial-off-the-shelf
DOD	Department of Defense
DSN	Deep Space Network
ESA	European Space Agency
GENSO	Global Education Network for Satellite Operations
GEO	Geosynchronous Earth Orbit
GSaaS	Ground Station as a Service
GSS	Ground Station Server
LEO	Low-Earth Orbit
MAC	Media Access Control
MC3	Mobile CubeSat Command and Control
MCC	Mission Control Client
MOC	Mission Operations Center
MOS	Mission Operations System
MPSA	Multiple Spacecraft Per Antenna
MSU	Morehead State University
NPS	Naval Postgraduate School
NRL	Naval Research Laboratory
R&D	Research and Development
RF	Radio Frequency
SATRN	Satellite Agile Transmit and Receive Network
SDR	Software-defined radio
TLE	Two-Line Element

TT&C	Telemetry, Tracking, and Control
USCGA	United States Coast Guard Academy
USNA	United States Naval Academy
VM	Virtual Machine
VPN	Virtual Private Network

I. INTRODUCTION

The recent boom in commercial space industries demonstrates first-hand an increasing interest in profitable ventures for this traditionally governmental enterprise. This is due to advancements in space applications that have taken place over the past decade, one of which is the prevalence of CubeSats [1]. CubeSats are miniaturized satellites that often utilize commercial-off-the-shelf (COTS) products, which greatly drive down the cost of development. Advances in technology, such as the miniaturization of electronics with lower power draw requirements and autonomous capabilities, lighter and stronger materials, and advanced manufacturing processes, have further increased the accessibility of space. This, coupled with the ease with which CubeSats can be deployed as secondary payloads of launch vehicles, has led to a low-cost and attractive option for a variety of entities to get involved in the valuable industry of space.

For academic institutions, this is not fresh news. Stanford University, in collaboration with California Polytechnic State University, is credited with starting the CubeSat program in 1999 with the goal of creating low-mass, low-cost satellites [1]. Since then, over a thousand CubeSat missions have been launched [1]. CubeSats provide an accessible platform for a variety of space-related applications, which make them the ideal choice for researchers and scientists. Not only do they enable advancement in aerospace and satellite technology, CubeSats provide students valuable, hands-on experience with satellite design that fosters a compounding interest, furthering the expansion of space-related possibilities [2].

With the rapid growth and interest in CubeSat missions, there must be an accommodatingly capable ground station infrastructure. Because a majority of CubeSat missions occupy low-Earth orbit (LEO), communication windows with the spacecraft are relatively short. Therefore, it is difficult to communicate with the satellite with only one local ground station [3]. To maximize contact times with spacecraft, LEO satellite operators utilize a resilient ground station network that connects multiple remote ground stations that are strategically dispersed. Through various methods, the remote ground stations share downlinked data and provide multiple pathways for command uplink. A long list of ground station networks exists today, operated by a variety of government, military, commercial, and educational entities. Prominent ground station networks will be discussed in depth in Chapter II. One such example is the Mobile CubeSat Command and Control (MC3) ground station network, which is the focus of this research effort.

A. MC3 OVERVIEW

MC3 is a U.S. government project, led by the Department of Defense (DOD), that supports small satellite operations via a community-based common ground infrastructure [4]. It began in 2012, primarily as a joint program of the Naval Research Laboratory (NRL) and Naval Postgraduate School (NPS) to allow the U.S. government to focus on the development of its small satellite capabilities by providing a streamlined, common-use infrastructure for small satellite missions [5], [6].

Originally, MC3 ground stations were designed to have the flexibility to support operational missions as well as operate locally to support testing and development priorities of the local host institution [4]. However, since its inception, MC3 has grown to become a fully operational ground station network that supports a wide variety of CubeSat missions from a diverse range of government organizations, commercial contractors, and educational universities [4]. Because so many operators depend on the reliable operation of MC3, institutions have lost the freedom to experiment, research, and test new innovations locally on the ground segment infrastructure they host. In order to remain competitive and advance United States' small satellite capabilities, this research addresses the rigidity of the research and development (R&D) process in MC3 and proposes a flexible solution.

B. PURPOSE

The main effort of this research is to design, implement, and test a ground station parallel to MC3 that provides institutions the flexibility to innovate and perform research on the ground segment without unnecessarily interrupting the operation of MC3 missions. The parallel station will use the existing radio frequency (RF) assets of MC3, namely the radios and antennas. MC3 has invested in high-value hardware to provide reliable communications with spacecraft. Sharing resources with the existing MC3 assets greatly reduces costs of a fully separate educational ground station and distributes the responsibility of maintenance. A major objective is to determine where to integrate the parallel station into the MC3 infrastructure and how to optimally switch between the operational MC3 network and the parallel station without disturbing the configurations and security of the operational network. Current and potential MC3 operators—namely NPS, U.S. Coast Guard Academy (USCGA), and U.S. Naval Academy (USNA)—have expressed interest in adding a separate educational ground station to their MC3 infrastructure. The parallel station will benefit these institutions by serving as an educational tool for students to foster experience with running a ground station and small satellite operations. Furthermore, by allowing the freedom to innovate and develop the ground infrastructure, MC3 as a whole will benefit—successful testing of new concepts on the parallel station will lead to eventual implementation on the operational MC3 stations that will increase its capabilities and mitigate pain points.

This thesis is broken down into five chapters. Chapter II provides background information on existing ground networks to lay a fundamental understanding of current practices and demonstrate a niche that this research effort fills. Chapter III provides an extensive overview of the MC3 ground network as well as the design of the educational ground station and potential use cases. Chapter IV presents various testing and results of the educational station, verifying and validating the design and use cases presented in Chapter III. Lastly, Chapter V concludes the thesis and offers suggestions for future work.

THIS PAGE INTENTIONALLY LEFT BLANK

II. SATELLITE GROUND STATION OVERVIEW

A wide variety of ground station networks have come to fruition since the beginning of the space age, spanning government, commercial, and educational undertakings. This chapter serves as a detailed overview of various stations and networks from each of the aforementioned institutions. Various aspects of each, such as mission, hardware and software, limitations, and resources, will be compared. Particular attention will be paid to technological advancement and integration capabilities of the networks to demonstrate how this thesis effort will fill a particular niche in this fluid new space age.

A. GOVERNMENT SATELLITE GROUND NETWORKS

1. NASA Deep Space Network

The NASA Deep Space Network (DSN) is a satellite ground network designed specifically for space missions that extend beyond the Geosynchronous Earth Orbit (GEO) (typically interplanetary spacecraft), as well as scientific observations in deep space radio and radar astronomy [7]. Since 1958, DSN has been designated as the standard ground segment for interplanetary missions. DSN consists of stations at three locations: Goldstone, California; Madrid, Spain; and Canberra, Australia. These locations are strategically placed approximately 120 degrees apart longitudinally in order to maintain constant communication with spacecraft as the planet rotates [7]. Each station consists of multiple large parabolic dish antennas, ranging in size from 70 meters, 34 meters, and 26 meters. DSN provides a wide variety of data services, including commanding, telemetry, tracking, and radio/radar science and astronomy [7]. Figure 1 shows the standard functional interoperability of DSN with a mission operations system (MOS). DSN stations have the ability to communicate and track spacecraft in the S-band, X-band, K-band and Ka-band, as shown in Figure 2.

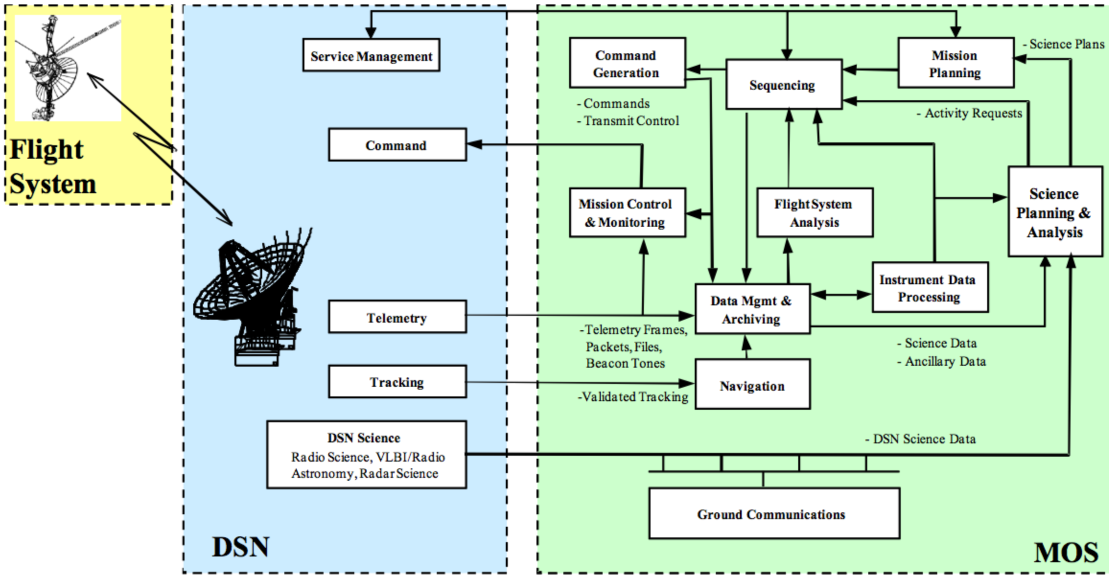
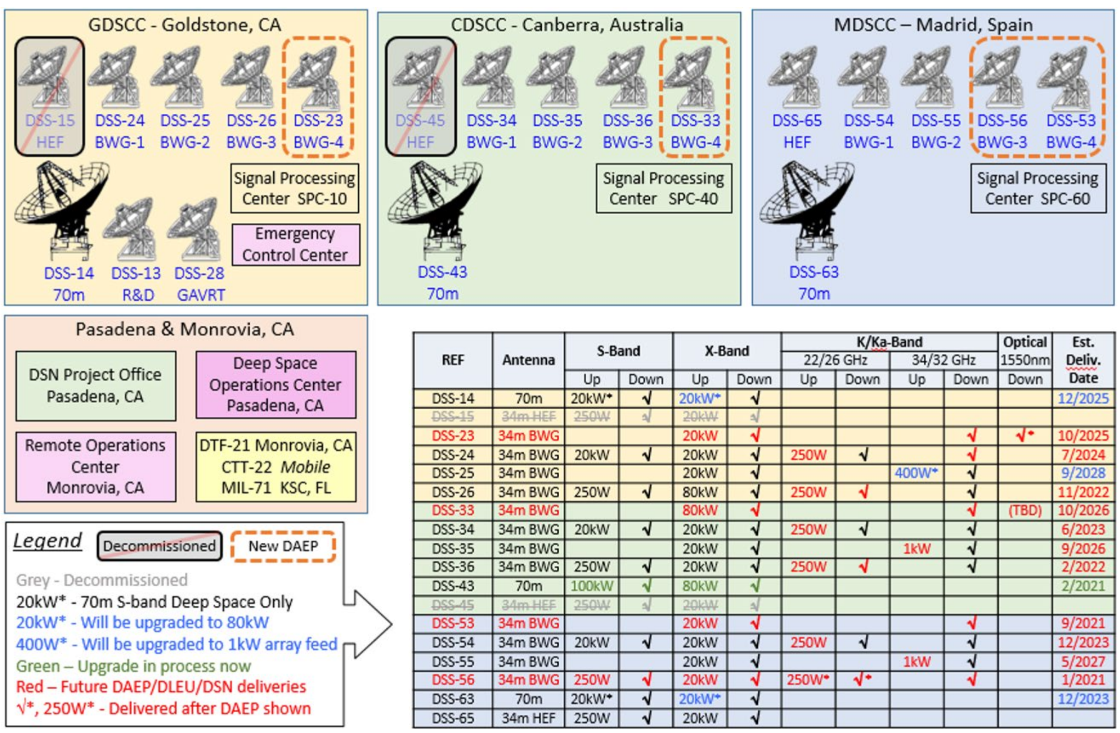


Figure 1. DSN functional diagram. Source: [8].



REF	Antenna	S-Band		X-Band		K/Ka-Band 22/26 GHz		K/Ka-Band 34/32 GHz		Optical 1550nm	Est. Deliv. Date
		Up	Down	Up	Down	Up	Down	Up	Down	Down	
DSS-14	70m	20kW*	√	20kW*	√						12/2025
~~DSS-15~~	~~34m HEF~~	~~250W~~	√	~~20kW~~	√						
DSS-23	34m BWG			20kW	√				√	√*	10/2025
DSS-24	34m BWG	20kW	√	20kW	√	250W	√		√		7/2024
DSS-25	34m BWG			20kW	√			400W*	√		9/2028
DSS-26	34m BWG	250W	√	80kW	√	250W	√		√		11/2022
DSS-33	34m BWG			80kW	√				√	(TBD)	10/2026
DSS-34	34m BWG	20kW	√	20kW	√	250W	√		√		6/2023
DSS-35	34m BWG			20kW	√			1kW	√		9/2026
DSS-36	34m BWG	250W	√	20kW	√	250W	√		√		2/2022
DSS-43	70m	100kW	√	80kW	√						2/2021
~~DSS-45~~	~~34m HEF~~	~~250W~~	√	~~20kW~~	√						
DSS-53	34m BWG			20kW	√				√		9/2021
DSS-54	34m BWG	20kW	√	20kW	√	250W	√		√		12/2023
DSS-55	34m BWG			20kW	√			1kW	√		5/2027
DSS-56	34m BWG	250W	√	20kW	√	250W*	√*		√		1/2021
DSS-63	70m	20kW*	√	20kW*	√						12/2023
DSS-65	34m HEF	250W	√	20kW	√						

Figure 2. DSN antenna frequencies. Source: [8].

DSN employs some alternative configurations to improve the performance and efficiency of a station. Typically, parabolic dish antennas like those used by DSN are only able to communicate with one spacecraft at a time. However, DSN has the capability to communicate with multiple spacecraft, known as Multiple Spacecraft Per Antenna (MPSA). In this configuration, multiple receivers are connected to a single antenna to allow downlink reception from up to four spacecraft, provided they are transmitting at different frequencies [8]. This configuration allows DSN stations to operate more efficiently and increases their volume for missions. Another alternative configuration provided by DSN is the ability to array multiple antennas at a given station. This capability increases the performance by effectively creating a larger antenna by combining the available parabolic dishes [8].

As such a large and important government project, DSN has a higher budget and considerably more manpower than most ground networks. However, as demands for deep space missions grows with the new space age, including the Artemis missions to the moon, DSN is under pressure to modernize in order to accommodate the increasing volume of missions [9]. This challenge is compounded by the fact that some legacy missions, such as Voyager, continue to operate long after their expected lifetime. This requires DSN to keep systems that are compatible with these aging missions in order to maintain communications. With antennas and hardware that are decades old, consistent maintenance is also required for DSN, sometimes leaving antennas out of commission for months.

In order to develop new technology and innovations to be integrated into DSN, there is a dedicated antenna for experimental purposes. DSS-13 in the Goldstone complex is a 34-meter beam waveguide (BWG) parabolic dish antenna that is not a part of the operational network. Its sole purpose is to test experimental concepts in space communications, such as experimental receivers and optical communication systems. In fact, the new 34-meter BWG antennas will include a hybrid optical/RF communication system that was tested at DSS-13. In addition to DSS-13, DSN uses a host of partner installations, such as Morehead State University, to be discussed later in this chapter, to conduct research and development when these stations are not being used operationally.

Despite the means to conduct research and development, it is difficult to implement new technology onto DSN. As mentioned, DSN is a very important asset to many missions, and thus cannot be taken offline very long. Furthermore, it is difficult to make new additions compatible to the legacy hardware, and it costs millions of dollars to stand up a new, state-of-the-art antenna that includes advancements. Regardless of the limitations to modernize DSN, it will remain a critical asset as the number of space missions in Earth orbit and beyond continue to grow in number.

2. NASA Near Space Network

The Near Space Network (NSN) is NASA's counterpart of DSN. Together, the networks make up NASA's Space Communications and Navigation (SCaN) program. As the name implies, NSN's primary mission is to communicate with near-Earth spacecraft within the range of two million kilometers. NSN was created in October of 2020 when NASA combined its Near Earth Network and Space Network. The Near Earth Network is made up of global NASA and commercial ground stations, responsible for the communications link to Earth-orbiting satellites. The Space Network's primary responsibility is to provide relay communications via Tracking and Data Relay Satellites (TDRS) for launch vehicles and LEO spacecraft. Together, the networks are responsible for 98% of NASA's daily space data [10]. The networks were combined to accommodate SCaN's goal of relying primarily on commercial communications services by 2030. By using a large conglomerate of decentralized ground stations, NASA can "focus on perfecting transformational space communications and navigation technologies," by shifting manpower and resources from nominal operations to technology development, according to SCaN Deputy Associate Administrator Badri Younes [10].

NSN, which is based out of NASA's Goddard Space Flight Center in Greenbelt, Maryland, operates as a single, end-to-end network that leverages government and commercially operated ground stations. As an end-to-end network, NSN provides services for the full mission life cycle and seamlessly negotiates with its wide variety of link providers on behalf of the mission. Its customers include government institutions such as the DOD and the National Oceanic and Atmospheric Administration (NOAA), commercial entities such as Boeing and Lockheed Martin, and international organizations like the Japan Aerospace Exploration Agency and the European Space Agency [11]. NSN currently supports over 50 missions and operates in S, X, Ku, and Ka frequency bands. Figure 3 shows the locations of NSN ground stations and Figure 4 summarizes NSN architecture.

Figure 3. Map of NSN ground station infrastructure. Source: [11].

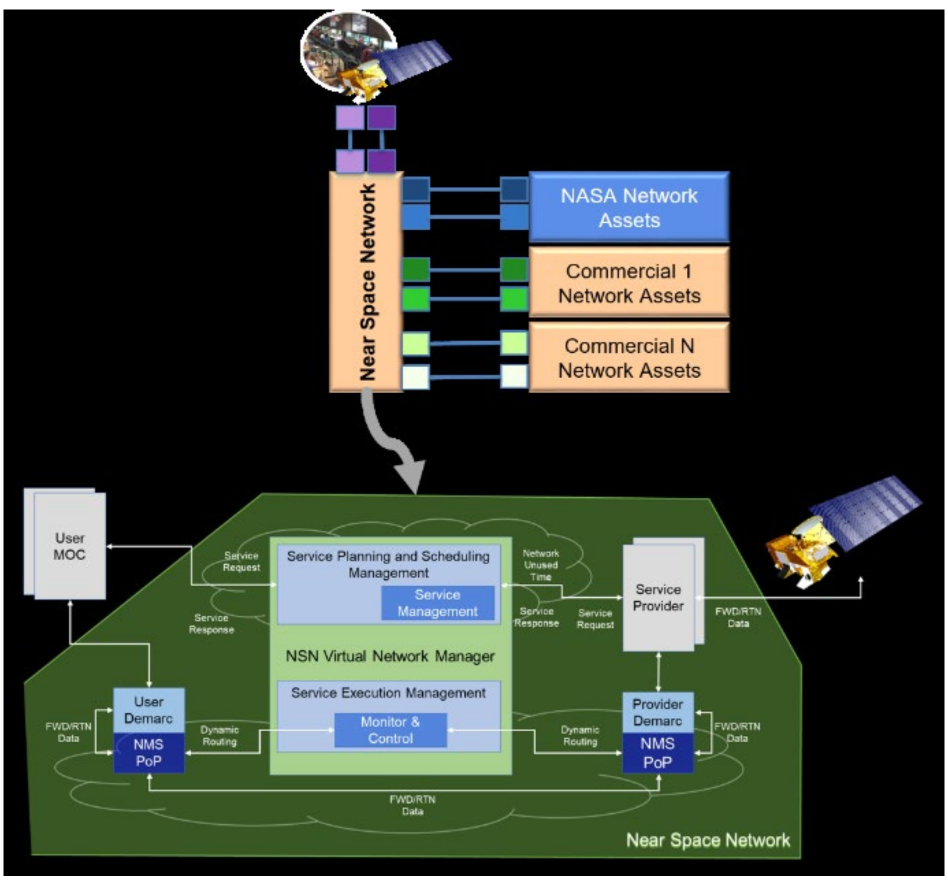


Figure 4. NSN architecture overview. Source: [12].

By leveraging a broad spectrum of commercial link providers, NSN has more flexibility to implement advances in communications technology compared to DSN. Additionally, NSN provides avid support for small satellite research through dedicated commercial small satellite services providers, such as KSATlite which will be discussed later. While large, expensive satellite missions usually have the budget to perform extensive, in-house communication compatibility and integration testing with ground stations, small satellites do not usually have this luxury. Therefore, NSN developed a standardized test lab dedicated to small satellites that simulates end-to-end communications between spacecraft and ground station, known as NSN Integration and Testing for Reliable Operations, or NITRO [12]. Although NSN provides relatively low costs by essentially monopolizing link service providers, the broad services provided by NSN can be unaffordable to smaller organizations such as educational institutions.

B. COMMERCIAL SATELLITE GROUND NETWORKS

1. KSATlite

Kongsberg Satellite Service, or KSAT, is a branch of Kongsberg Gruppen, a Norwegian-based international technology company that specializes in a broad reach of disciplines, including maritime, defense and aerospace, and digital industry [13]. KSAT in particular provides ground network services for a wide variety of space missions, extending out to Lunar missions, as well as Earth remote sensing services. KSATlite is KSAT's dedicated small satellite ground communication network. Whether it be a single small satellite or a large constellation, KSATlite provides flexible and scalable Ground Station as a Service (GSaaS) through its global ground network, shown in Figure 5. With its global coverage, KSATlite boasts 45000 passes per month for its client satellites.

Figure 5. KSAT global ground network. Source: [14].

By using software-defined radios (SDR), KSATlite is capable of conforming to any communication standard. KSATlite provides up and downlink communications in the S-band and UHF, as well as downlink in X-band and Ka-band. With a standardized infrastructure, KSATlite provides customers with streamlined and fast integration, as well as spacecraft testing services. KSATlite offers a range of data-handling solutions for different levels of security, including public and private cloud solutions and computing and data processing at remote sites. Furthermore, KSATlite provides a web-based automated scheduling portal for customers to schedule and access data transmissions to their spacecraft [15].

As mentioned previously, KSATlite has been integrated with NASA's NSN to provide lower direct-to-Earth communication costs through the use of small aperture parabolic dish antennas, at a size of 3.7 meters compared to the nominal 11-meter dish used by NSN. Furthermore, KSATlite increases equatorial and mid-latitude coverage for NSN [12]. Kongsberg, as a dedicated technology company, is one of the aforementioned commercial providers that delivers NSN modern communication technology advancement and implementation through KSATlite and its global ground network.

2. SatNOGS

SatNOGS is a project by the Libre Space Foundation, a Greek establishment that seeks to provide open source and free accessibility to space technologies. SatNOGS is the dedicated satellite ground network that provides users free and open source LEO satellite communications by leveraging amateur radio operators and idle small satellite ground stations. They provide the civilian infrastructure to allow interested individuals and organizations to build their own ground stations—estimated to cost $300-$500 to build from scratch—or to leverage their network to communicate with small satellites in LEO. By using a modular approach to the ground station segment and flexible software and hardware options, like SDRs, existing ground stations, amateur or otherwise, can easily be integrated into the SatNOGS network [16]. SatNOGS currently has 60 operational stations with more in development, as shown in Figure 6.

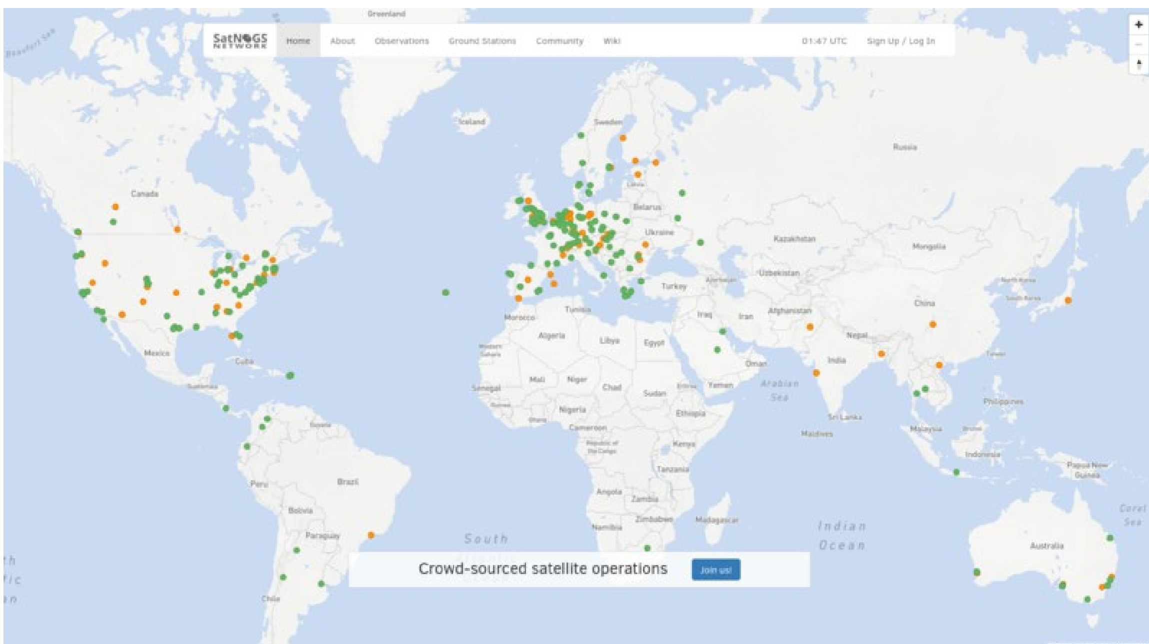


Figure 6. SatNOGS ground network. Source: [17].

There are four major software components that make up SatNOGS: Network, Database, Client, and Ground Station. The SatNOGS Network is a web-based interface that manages scheduling for satellite passes. The Database is a crowd-sourced, centralized repository of satellite information, such as relevant frequencies and modulations, for over 250 satellites with more than 433 transmitters. It also stores collected telemetry data, with an estimated 16+ million telemetry frames in storage [16]. The Client contains the computer software that controls, at a high level, the ground station hardware and connects the ground station to the Network. The SatNOGS Ground Station incorporates the hardware of a specific ground station, including the rotator, antennas, and RF path [16]. SatNOGS provides flexible, modular options for the four major software components to comply with most industry standards to allow for ease of integration. The relationship between the components is shown visually in Figure 7.

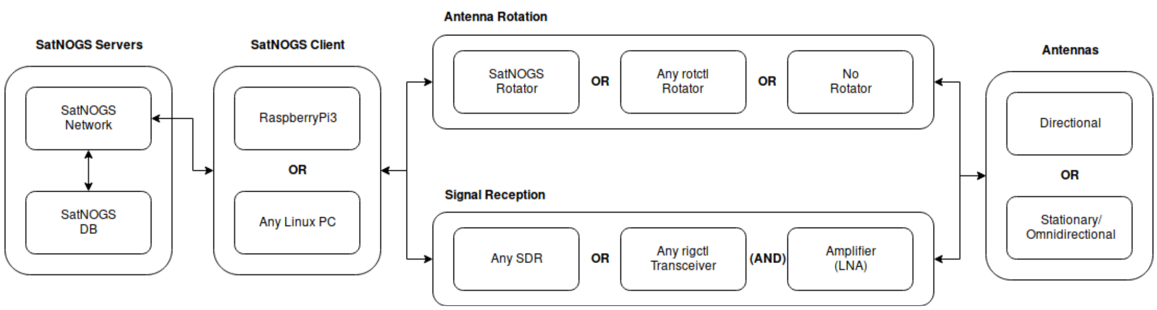


Figure 7. Relationship between major SatNOGS components and modular options. Source: [16].

In order to remain inexpensive, most SatNOGS stations utilize any combination of commercial or home-made UHF and VHF antennas. These antennas can either be omnidirectional or rotator-controlled. These frequency bands are usually not ideal for satellite communications due to higher atmospheric losses and thus lower data rates, but they do meet the SatNOGS goal of cheap and attainable hardware. S-band dish antennas can also be, and are, integrated into the SatNOGS network if the operator desires to spend the extra money.

Labelling SatNOGS as a commercial venture would be a stretch, as it is a free and open-source operation. However, it provides the infrastructure for commercial organizations to easily get involved in the lucrative business of space or for educational institutions to build the foundations of a space program. As a web-based, open-source network with freely available received data, security is obviously not a priority for SatNOGS users. Furthermore, given the low cost of the network, it is unlikely that individuals and organizations have the budget to perform meaningful research and development. However, since they essentially run their own ground station, users do have the flexibility to perform research and development if they have the means and desire. Overall, SatNOGS is a unique flavor of ground network that is pursing the unique goal of making space more flexible and attainable.

C. EDUCATIONAL SATELLITE GROUND STATIONS

1. GENSO

The Global Educational Network for Satellite Operations (GENSO) was a project sponsored by the European Space Agency (ESA) with the goal of uniting amateur and university ground stations around the world to increase the accessibility of space communications to educational institutions. GENSO was ideally very similar to SatNOGS, but placed a greater emphasis on educational and university participation. Essentially, GENSO was a software standard that allowed station operators to share LEO satellite data with the spacecraft mission controllers through the internet. Like SatNOGS, GENSO allowed for flexible integration with most COTS hardware but required specific software to be integrated onto the network.

The Authentication Servers (AUS) serves as the central hub of the network and is only accessible by GENSO administrators. Several AUS exist, collectively known as the Authentication Servers List (ASL), to provide redundancy and reduce probability of single point failure. The main role of the AUS is to validate a ground station operator on the network and maintain an active list of ground stations and spacecraft and their statuses. The Ground Station Server (GSS) is the software that is local to each GENSO ground station that performs the tracking and up/downlink communication for GENSO authorized spacecraft. The Mission Control Client (MCC) is where the satellite operator accesses downlinked data or provides commands to be sent to their spacecraft through a ground station of their choosing [3]. Figure 8 demonstrates the relationship between the different pieces of GENSO.

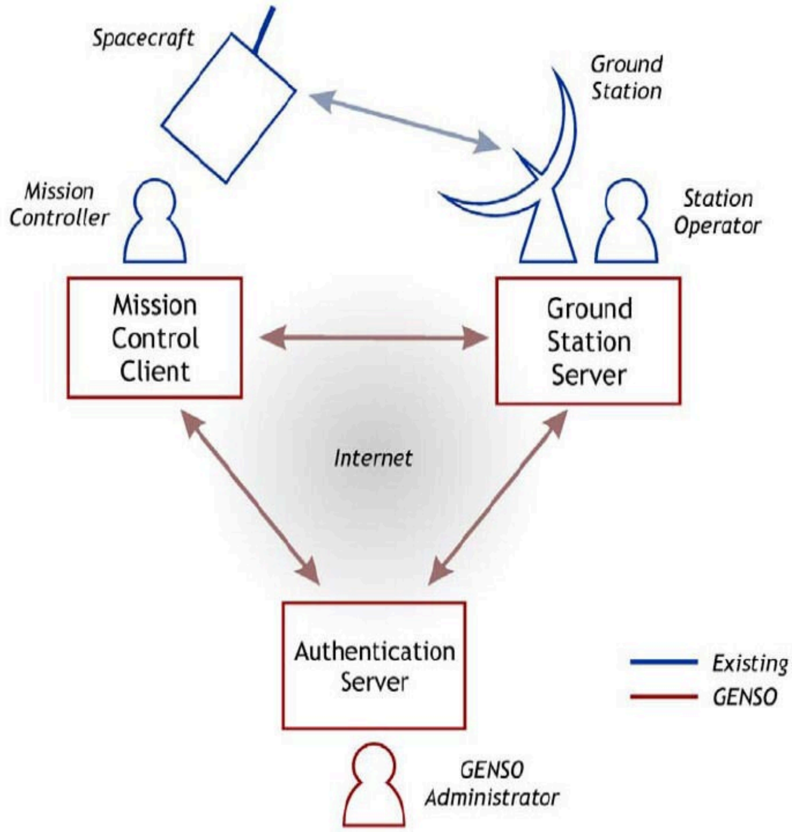


Figure 8. GENSO network architecture. Source: [3].

Through the use of the AUS, GENSO would theoretically provide more security than SatNOGS, as users must be authorized to access/send data as well as operate a ground station on the network. However, as an internet-based network, vulnerabilities in security still exist. Furthermore, there is a level of trust involved between mission controllers and ground station operators, as ground station operators have the ability to alter command and control messages provided by mission controllers. With the stringent requirements on software, GENSO ground stations do not have as great of flexibility in research and development as SatNOGS. Similar to SatNOGS, though, flexibility in hardware requirements allows station operators the freedom to experiment with different hardware as they please.

Despite its promise, GENSO was cancelled before it was fully operational. This was likely because the project came before the commercial golden era of space and failed to foster interest from American institutions, from where CubeSat development has been most prominent.

2. Morehead State University Ground Station

Morehead State University (MSU) in Morehead, Kentucky, has been operating a 21-meter parabolic dish antenna since 2006. The dish serves many functions beyond space communications. The dish is critical for radio astronomy research, and also serves as a testbed for advanced RF systems [18]. Because small satellites typically have low power transmission capabilities, the large 21-meter dish is a valuable asset to the CubeSat community, as it provides communications capabilities from LEO out to lunar orbits, and potentially to Earth-Sun Lagrange points. MSU provides flexible options in feed systems, offering L-band, Ku-band, and S-band, with X-band and C-band feeds being developed for integration. Furthermore, as a university-based ground station, MSU provides its undergraduate students hands-on experience of operating a ground station, as well as a plethora of research opportunities for both faculty and students [19].

As mentioned previously, MSU’s 21-meter dish is an affiliated node on the DSN, denoted as DSS-17. In order to be integrated into DSN, a lengthy, complicated process was required to effectively separate the educational/commercial network from the DSN in order to ensure proper operation and security. The resulting product is essentially two separate networks that share some hardware components. In order to manage the complex switching between the university network and DSN, MSU created a computer-controlled interface software that manages configurations and directs RF and digital traffic to the correct port.

It is crucial that there is no crossover between the networks. Therefore, the station cannot service both networks at the same time. As a government entity, DSN takes priority over the university schedule. DSN manages their own schedule, and blocks out time for which the MSU node will be used. MSU manages the schedule for commercial satellites and local research and works around DSN’s scheduled passes. Similarly, students are not involved with DSN passes due to their level of importance. Students train on the ground station during university research and non-DSN satellite passes including commercial satellites if the customer approves.

MSU's 21-meter antenna is a uniquely flexible asset in space communications by serving government, educational, and commercial interests. The relationship between DSN and the university network provides the closest semblance to the objective of this thesis: creating a parallel, but separate, ground station that can be used for educational purposes and research alongside the secure operational network.

THIS PAGE INTENTIONALLY LEFT BLANK

III. DESIGN AND INTEGRATION OF PARALLEL GROUND STATION WITH MC3

This chapter first provides a detailed overview of MC3, and then describes the design and integration of the educational station with the operational MC3 network. Furthermore, a Concept of Operations (CONOP) is provided on how to properly inject the parallel station into the network and return to the operational configuration.

A. MC3 OPERATIONAL NETWORK

1. Network Overview

MC3 currently consists of nine installations fielded at partner institutions across the United States, as shown in Figure 9. The stations in yellow—Johnson Space Center (JSC) and the U.S. Naval Academy (USNA)—are potential MC3 nodes that have not yet been integrated into the network. Table 1 provides further information on the installations.

Figure 9. MC3 station locations

Table 1. MC3 station locations and capabilities

Site (Designator)	Location	Capability
Naval Information Warfare Center (PCH)	Pearl City, HI	UHF/S-band/X-band
Naval Postgraduate School (NPS)	Monterey, CA	UHF/S-band/X-band
Space Dynamics Laboratory (SDL)	Logan, UT	UHF/S-band
University of New Mexico/Cosmiac (UNM)	Albuquerque, NM	UHF/S-band
Air Force Institute of Technology (AFIT)	Dayton, OH	UHF/S-band
US Coast Guard Academy (USCGA)	New London, CT	S-band
Malabar Transmitter Annex (MLB)	Palm Bay, FL	UHF/S-band
University of Alaska, Fairbanks (UAF)	Fairbanks, AK	S-band
Space and Missile Defense Command (SMDC)	Huntsville, AL	S-band/X-band

The ground stations are connected through Amazon Web Services (AWS), a DOD accredited, COTS virtual private network (VPN) architecture that provides secure, cost-effective communications between MC3 nodes. Housed within AWS is the Satellite Agile Transmit and Receive Network (SATRN) software, which allows satellites operators the ability to remotely schedule passes and transmit commands to their satellite, as well as access received data from their satellite. Thus, satellite operators can leverage MC3 stations as a means of bent-pipe communications from any location with internet, which effectively decreases the ground segment cost of small satellite missions [4].

Developed by the Space Dynamics Laboratory, SATRN consists of four applications: Server, Client, GroundSite, and Hardware Manager. Figure 10 demonstrates the relationship between the four components. The satellite operator will VPN into SATRN Client via AWS from their mission operations center (MOC) to request a contact. The SATRN Server, which manages a database of hardware, missions, and ground stations, is charged with scheduling the pass using a simple deconfliction algorithm. Once a pass is initiated, a direct network path is opened between the MOC and the ground station via SATRN, through which the MOC can send uplink data through the ground station to the antenna, and vice versa for downlink data. SATRN GroundSite is responsible for passing data between MOC and antenna, as well as steering the antenna to track the satellite. SATRN Hardware Manager lives at the ground station with GroundSite. Hardware Manager contains a database of all the radios and rotators available at a specific ground station and is tasked with activating the correct radio (or SDR solution) and rotator associated with the specific satellite in the contact [4]. The specifics of the ground station hardware will be discussed in next section.

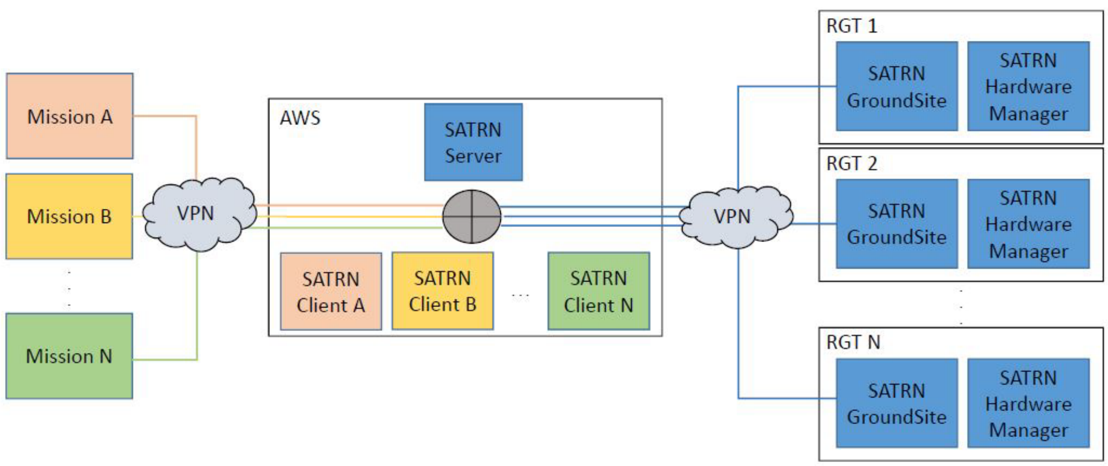


Figure 10. SATRN architecture. Source: [20].

2. Ground Station Overview

As observed from Table 1, the different ground stations on the MC3 network have different capabilities. However, their overarching architectures are very similar. Ground stations typically utilize a 3-meter parabolic dish antenna and Yagi antennas to service common small satellite frequency bands: S-band and UHF, respectively [4]. Figure 11 presents a typical antenna setup at an MC3 ground station. The antennas are controlled by a rotator which receives position commands from SATRN to move the antenna and track a satellite throughout its pass.

Figure 11. MLB ground station, UHF Yagi and S-band dish/radome. Source: [4].

Like some of the small satellite ground stations mentioned in the previous chapter, MC3 mainly utilizes SDRs at its ground stations. SDRs provide a cost-effective technique to support a wide variety of communication protocols, which gives the station flexibility in the types of protocols it can support. Satellite operators can either write software or use open-source software that is compatible with their specific spacecraft radio. When they schedule a pass, SATRN can retrieve the satellite-specific software and load it to the SDR to ensure proper transition between digital and RF signals and thus effective communication with the spacecraft. Some satellite operators prefer to use commercial radios for their spacecraft, which often require compatible radios at the ground stations. For example, in addition to its SDR, the NPS stations also has a Kratos radio. While commercial radios like Kratos are more expensive and less flexible, they are simpler to implement as the protocol software is already written into the radio.

In addition to the antennas and radios, MC3 ground stations utilize typical RF hardware such as amplifiers, filters, and switches. Figure 12 shows the hardware configuration of the NPS ground station. Figure 12 also includes access points into the hardware configuration for the educational station. These entry points are investigated further in the next section.

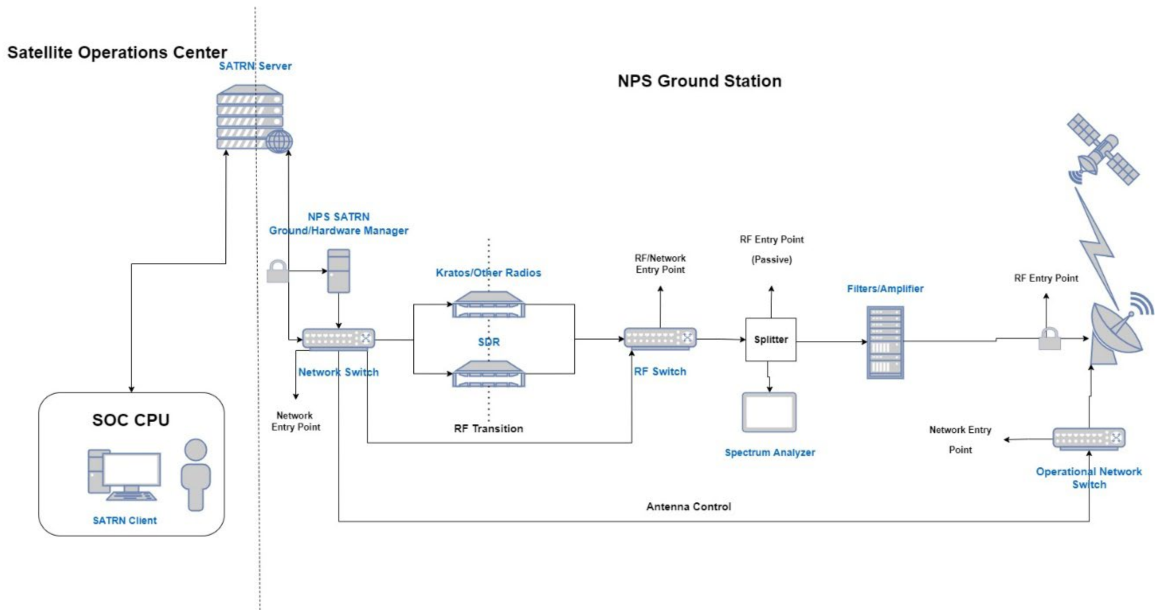


Figure 12. NPS ground station hardware configuration

B. MC3 EDUCATIONAL NETWORK AND CONOP

The minimal requirements for a parallel ground station are a computing device and a switching device to connect other hardware components. Examining Figure 12, there are essentially two paths that route from the operational network to the antenna that diverge at a network switch. There is one potential entry point for the educational station at the network switch where the pathways diverge. This injection point would be the ideal point of access for the educational stations since it maximizes the amount of shared hardware between the operational and educational stations, including radios, RF hardware, and antenna. However, this network switch is securely configured, meaning that the station will lock up if an unrecognized MAC address is connected to the switch. This is because that network switch has direct access to the SATRN server, for which security is of utmost importance. Therefore, this entry point is not feasible to connect the educational station because reconfiguring the switch to recognize the MAC address of the computer of the educational station would require a lengthy approval process, thus defeating the purpose of the educational station.

The upper path is an RF pathway that consists of RF hardware. The injection points in this pathway are mostly infeasible to connect the parallel station CPU because the RF nature of the pathway is incompatible with the digital nature of the CPU. However, the RF entry points at the splitter and the RF switch are useful to injecting RF hardware, such as transmitters and receivers, that can be connected to the educational CPU through another injection point. The process of injecting RF hardware to be used with the parallel station is explained further in Chapter IV, Section C.

The lower path is a digital pathway that runs from the operational network, specifically the SATRN software, to the antenna control unit to allow the operational network to control the dish. Within this pathway, there is one potential injection point for the educational station to access the antenna at the ground station. This network switch does not employ the same secure configuration that the other network switch uses, which would allow the educational ground station computer to be connected to the antenna seamlessly. Therefore, this was considered the optimal and simplest injection point for the educational station for the purposes of this thesis. Because this injection point does not directly connect to the upper pathway, the educational station would have to connect its own radios and RF hardware to the switch in order to radiate and receive signals from the antenna. Figure 13 is a modification of the hardware configuration presented in Figure 12 to include the educational station at this injection point. It can be observed from Figure 13 that the antenna control pathway of the operational network is disconnected to ensure that the educational station is completely separated from the operational network. The radios and RF equipment are realistically routed to the dish separately, but are included in the same pathway for ease of comprehension of the architecture.

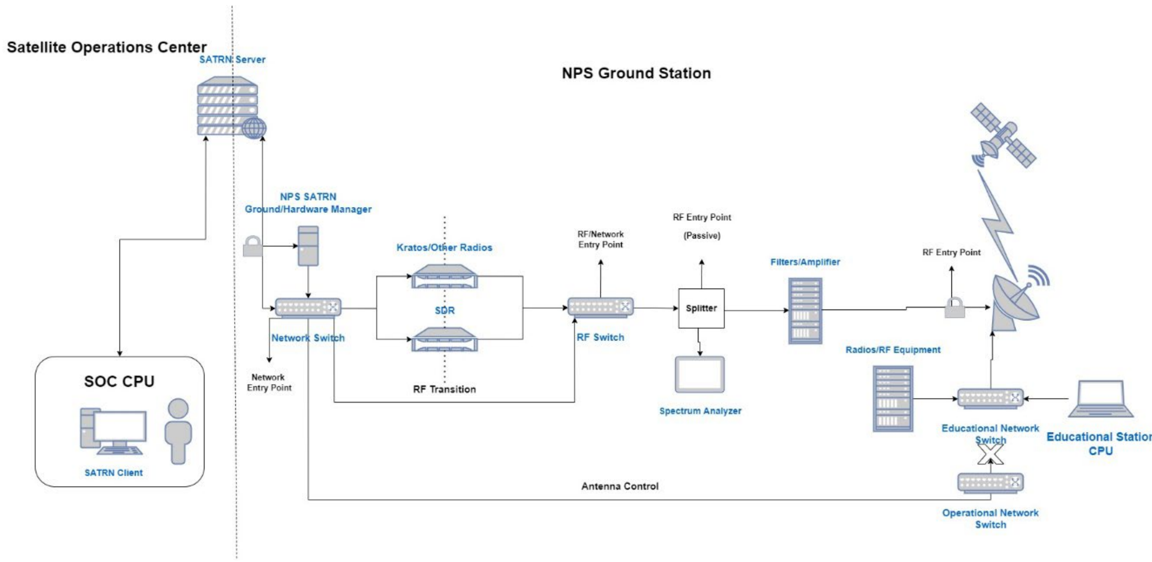


Figure 13. NPS ground station configuration with educational station connected

For the purposes of this research effort, a brute force injection is considered for simplicity. Brute force injection entails physically disconnecting and reconnecting interfaces of the operational network to the educational station. An alternative to brute force injection would be using switches to route network traffic digitally rather than physically. This could be accomplished either manually or autonomously. These techniques will be considered for future work. Because of the nature of brute force injection, the operational and educational stations cannot be operated simultaneously. Therefore, the operational station must be taken offline to utilize the educational station, which requires scheduling educational station usage outside of the designated operational station schedule. The operational station schedule always takes precedence over the educational station usage.

Table 2 outlines the general CONOP for configuring the educational station via brute force injection method. The development of this procedure, as well as examples on how it is executed, is described in Chapter IV.

Table 2. Educational station brute force injection procedure

Step	Procedure description
1	Perform test readiness review: describe experimental procedure for the educational station, identify hardware required and configuration changes to operational station
2	Notify users on MC3 network of potential station downtime (via NPS personnel), adjust station downtime to deconflict operational contacts.
3	Schedule downtime via SATRN Hardware Manager
4	Document operational configuration at the point of injection (i.e., port assignments and corresponding hardware)
5	Inject educational station and relevant experimental hardware per the test readiness review in step 1
6	Perform experiment on educational station, collect data
7	Disconnect educational station and experimental hardware, reconfigure operational configuration per documentation from step 4
8	End downtime in SATRN Hardware Manager
9	Run test pass via SATRN Client to ensure proper reconfiguration, notify users that the node is back online

IV. EDUCATIONAL GROUND STATION TESTS

This chapter describes the various tests performed on the educational ground station, including proper injection and return to operational configuration, implementation of velocity control on the antenna, and receiving and transmitting capabilities. Through these tests, the operational and educational benefits of the station are demonstrated.

A. TEST #1: BRUTE FORCE INJECTION

The purpose of this test was to validate and verify the CONOP provided in Chapter III Section B. The general procedure of this experiment was to follow the CONOP to inject a laptop that served as the computing device for the educational network. Once injected, the laptop was used to send commands to the S-band parabolic dish antenna through the antenna control unit (ACU) and move the dish to demonstrate proper connection of the educational station. Once completed, the operational station was reconfigured and tested to ensure proper reconnection.

A test readiness review was completed, outlining the information provided above (CONOP step 1). The point of injection was identified as the network switch as shown in Figure 14. The laptop was configured with an unused IP address on the NPS network in order to mitigate any unforeseen authentication errors in injection. At the time of testing, there was only one active satellite on the operational MC3 network. The test was scheduled to ensure there was no interference with the operational pass schedule of this satellite (CONOP step 2). Downtime was scheduled in SATRN for the NPS ground station (CONOP step 3).

The ACU was connected to the operational network switch at port 3 (CONOP step 4). Once the operational configuration was documented, the ACU was disconnected from the operational network switch and connected to the educational network switch. The laptop was also connected to the educational network switch (CONOP step 5). Figure 14 illustrates the configured educational station for this test. The ACU was disconnected from the operational network switch (top black box within equipment rack) and connected to the educational network switch (gray/blue box above equipment rack) along with the laptop (not pictured).

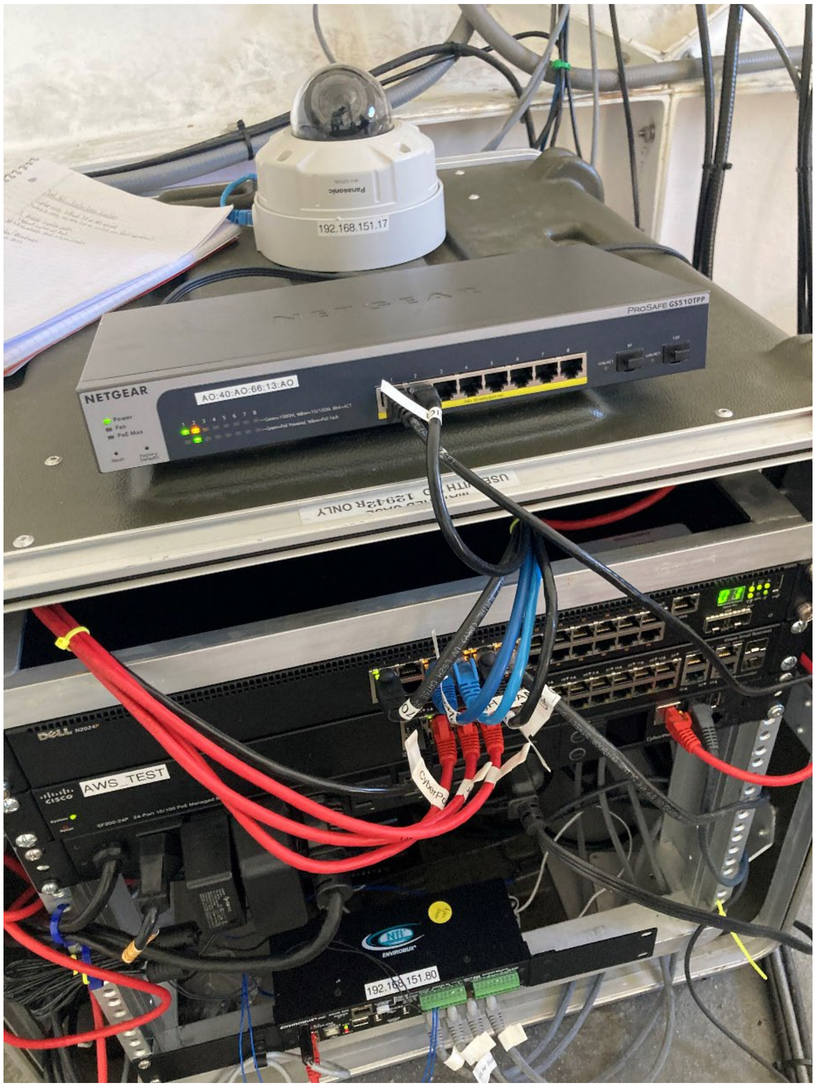


Figure 14. Configured educational station

The configuration of the educational station was tested by pinging the ACU through the laptop. The ping successfully returned packets, demonstrating a communications pathway to the dish through the ACU and thus a successful configuration of the educational station. The application MobaXterm was used on the laptop to communicate with the ACU. Various commands were sent to the ACU to observe the movement of the dish as well as the behavior of different commands. Position commands moved the dish to a specified azimuth or elevation angle, at which point the dish would stop. Velocity commands moved the dish at a constant angular velocity with respect to the azimuth or elevation axis until a new command was sent. The current dish position was requested using the status query command ("SQ"). The ACU successfully returned the dish's position to be displayed on the laptop's terminal (CONOP step 6).

Once the antenna control test was completed satisfactorily, the process of reconfiguring the operational station began. The communications terminal between the laptop and ACU was ended, and the ACU was disconnected from the educational network switch. The ACU was reconnected to port 3 on the operational network switch (CONOP step 7). A computer on the operational network was used to ping the ACU. The ping returned packets from the ACU, indicating successful reconfiguration of the ACU to the operational station. Station downtime was ended via SATRN Hardware Manager (CONOP step 8). A test pass at the NPS ground station using a simulated satellite was attempted through SATRN Client (CONOP step 9). However, there was an error with the SATRN session and the test pass could not be completed successfully. Since the ACU was successfully pinged through the operational network, the error was assumed to be connected to the SATRN software. It was later found through troubleshooting that disconnecting the ACU when there is scheduled downtime caused rotator and radio combinations to become unpaired. The combinations had to be manually reconnected in SATRN Hardware Manager. Because this only occurred when there was scheduled downtime in SATRN, this indicated a bug in the SATRN software was responsible for the error. Therefore, future tests were completed without scheduling official downtime via SATRN until the software bug was resolved. Because of this, special attention was paid to the operational schedule to ensure no conflicts with testing.

While the brute force injection test seemed relatively simple, it provided crucial proof of concept of the ability to inject the educational station and reconfigure the operational station. The test validated the injection CONOP as well as exposed a bug in the SATRN software. The intricacies of commanding the dish through the ACU were demonstrated, which proved beneficial in accomplishing future tests, namely the implementation of a new satellite tracking algorithm to be presented in the next section.

B. TEST #2: IMPLEMENTATION OF CLOSED-LOOP SATELLITE TRACKING CONTROL

The purpose of this test was to demonstrate the R&D capabilities of the educational station by investigating a relevant improvement to the tracking algorithm used by the MC3 operational network. Furthermore, this test was used to verify the receive capabilities of the educational station from an active satellite in orbit.

MC3's current tracking algorithm is straightforward and relatively simple. When a pass is scheduled by a client, SATRN pulls the satellite's two-line element (TLE)—a common satellite data set format that describes the satellite's location in orbit—from an online database managed by the Space Force known as Space-Track. From the TLE, SATRN propagates the satellite's orbit and provides time-stamped azimuth and elevation (Az/El) position pairs of the satellite relative to the respective ground station. Before the satellite begins its pass, the antenna is prepositioned to prepare to track the satellite. The Az/El position pairs are then sent to the ACU at the corresponding time to move the antenna throughout the satellite's contact window, tracking the satellite. For each commanded position, the angular velocity of the dish will ramp up and ramp down to achieve the commanded angular position in a rest-to-rest maneuver. Although this sort of open-loop controller is simple and easy to implement, the technique typically has lower pointing accuracy. The rest-to-rest maneuver causes the motion of the antenna to be jittery, and because the commands are sent instantaneously with the associated time-stamp, the antenna will lag behind the satellite's overhead trajectory.

As was demonstrated in Chapter IV, Section A, the ACU is capable of commanding antenna velocities as well as positions. The current tracking algorithm only uses position commands. A goal of this test is to investigate the benefits of using the velocity command feature to close the control loop and implement error correction into MC3's tracking algorithm. By using velocity commands, the antenna movement is smoother because the rest-to-rest maneuver is eliminated. Furthermore, it is expected that the implementation of error correction will increase pointing accuracy of the antenna and eliminate the lagging effect of the current tracking algorithm. A MATLAB code was created to implement the feedback control tracking algorithm. The algorithm was developed iteratively through multiple trials, starting with tracking a simulated satellite and concluding with tracking an active satellite on orbit.

1. Algorithm Development and Testing on Simulated Satellite

Systems Tool Kit (STK) was used to generate the trajectory of the simulated satellite. The International Space Station was chosen as the satellite to be simulated because its low-Earth orbit is similar to the orbits of the small satellites that utilize the MC3 network. For an arbitrary ground station, the Az/El trajectories were generated in STK for an arbitrary contact. A timestep of one second was used between each Az/El trajectory pair. Figure 15 illustrates the satellite's overhead trajectory.

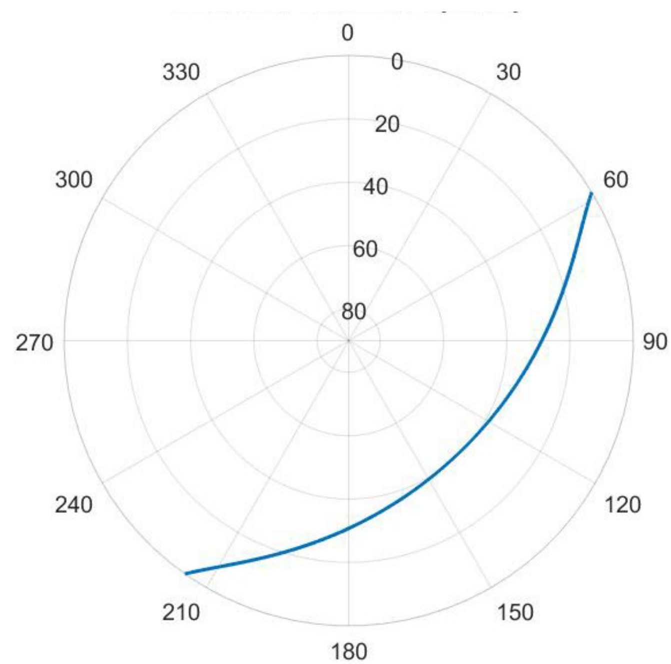


Figure 15. Simulated satellite overhead trajectory. Azimuth is measured on the angular axis (north = 0 deg), and elevation is measured on the radial axis (zenith = 90 deg)

The Az/El position pairs were loaded into a MATLAB script, and the angular velocities were calculated for the respective axes according to the angular velocity formula,

$$\omega = \frac{\Delta\theta}{\Delta t} = \frac{\theta(t+\Delta t) - \theta(t)}{\Delta t} \tag{1}$$

The dish was prepositioned using the initial azimuth and elevation positions of the satellite relative to the dish antenna. MATLAB's "tcpclient" function was used to create a communication link between MATLAB and the ACU. A for loop was constructed in MATLAB that iterated every timestep throughout the duration of the simulated pass. For each iteration, the azimuth and elevation velocity commands of that timestep were sent to the ACU. In addition to sending the commands, the dishes current status was queried using the "SQ" command, and the current position was pulled from the ACU's status message. The azimuth and elevation error were calculated by subtracting the dishes current Az/El position from the Az/El position pair retrieved from STK.

In the first iteration of the algorithm, the error correction occurred once each timestep. The velocity command of the next timestep was adjusted by subtracting the position of the next timestep from the dish's current position and dividing the timestep,

$$\omega(t + \Delta t) = \frac{\theta_{sat}(t+\Delta t) - \theta_{dish}(t)}{\Delta t} \tag{2}$$

where θ_{sat} is the angular position of the satellite relative to the dish, and θ_{dish} is the angular position of the dish. This command would theoretically provide the dish velocity needed to return the dish to the correct position in the next timestep that corresponds to the satellite's relative position.

To test the new tracking algorithm on the physical system, the same procedure used in the brute force injection test was used to configure the educational station and connect the ACU. First, the algorithm was tested without the error correction feedback, moving the dish solely with the velocity commands generated from Equation (1). This was done to observe the effects of the open-loop velocity control algorithm. Figure 16 compares the dish's trajectory with the satellite's trajectory, while Figure 17 provides a closer look at the time histories of the azimuth and elevation of both the dish and the satellite.

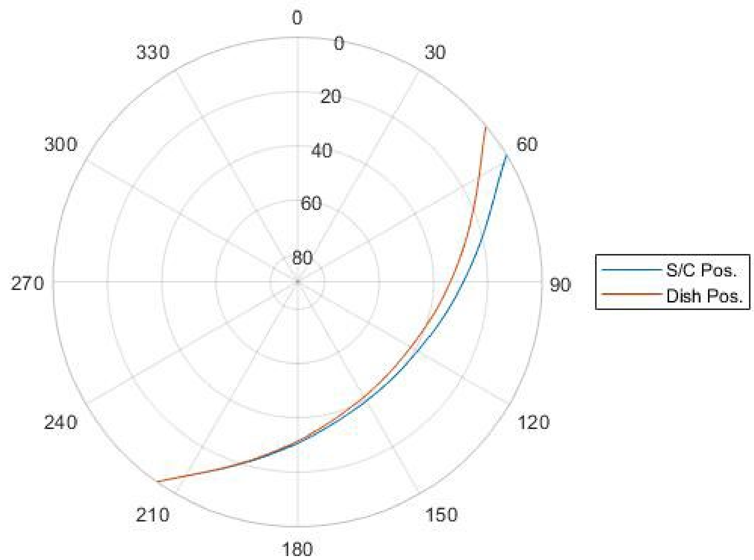


Figure 16. Comparison of the dish and spacecraft trajectories for the new tracking algorithm without error correction.

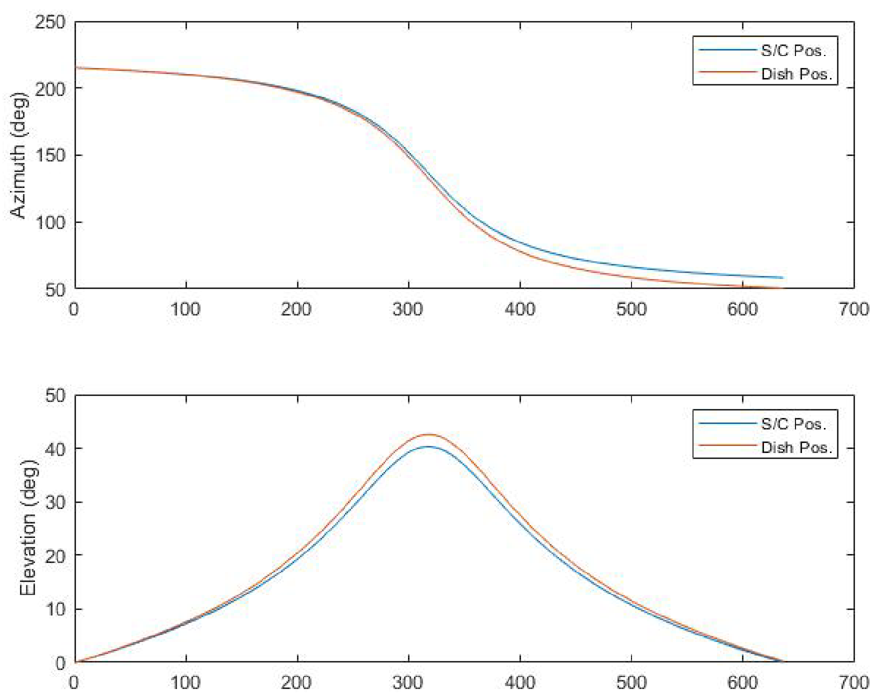


Figure 17. Azimuth (top) and elevation (bottom) time history (seconds) comparison of the dish and spacecraft

From Figures 16 and 17, it is evident that there is a large error associated with the tracking algorithm. This was expected since the complexities of the physical antenna system was neglected, and there was no feedback to correct the pointing error. In the next trial, the closed-loop error correction was included in the algorithm by updating the velocity commands through the process shown in Equation (2). Figures 18 and 19 provide the results for this iteration of the tracking algorithm.

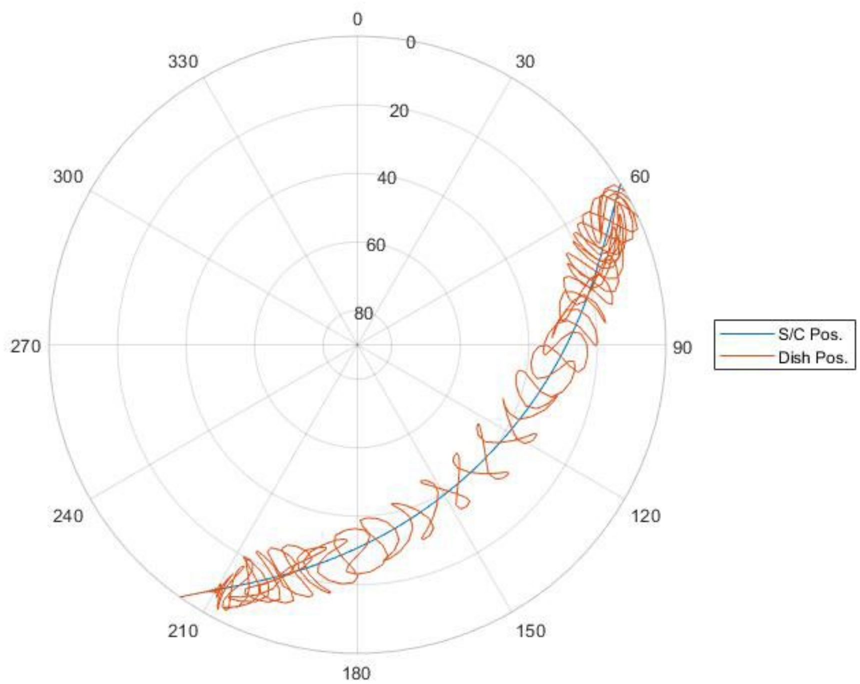


Figure 18. Comparison of the dish and spacecraft trajectories for the tracking algorithm with error correction every timestep (1 second)

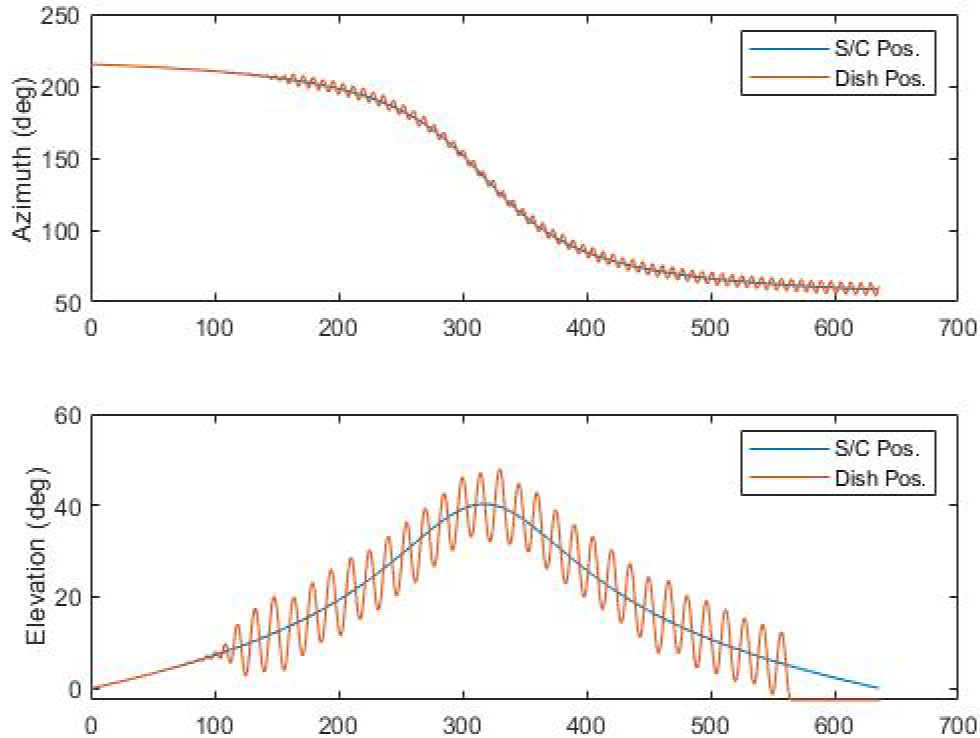


Figure 19. Azimuth (top) and elevation (bottom) time history (seconds) comparison of the dish and spacecraft

The tracking algorithm performed well until about 100 seconds into the pass, at which point the dish started largely oscillating about the spacecraft's trajectory. It was found that the error correction algorithm was commanding velocities that were larger than the dish's velocity limit of 3 deg/s. The dish was left to catch up to the large commanded velocities, and would thus overshoot the satellite's trajectory. This behavior was also likely due to the fact that error correction was being performed too often, i.e., every second. Essentially, the dish could not keep up with the corrected velocity commands.

To correct the algorithm, a velocity limit was placed on the generated commands. If the calculated command was greater than 3 deg/s, it would be corrected to be equal to 3 deg/s. Furthermore, the algorithm was adjusted so that the time between error corrections could be controlled. By using MATLAB's "mod" function, the error correction would only be applied after a certain number of timesteps. With this adjustment, the velocity correction equation presented by Equation (2) became,

$$\omega(t + \Delta t) = \frac{\theta_{sat}(t+c*\Delta t) - \theta_{dish}(t)}{c*\Delta t} \quad (3)$$

where c is the number of timesteps between each error correction. In theory, this formula provides the velocity command that would return the dish back to the satellite trajectory after c timesteps.

The updated algorithm was tested in the same manner as previously described. The algorithm was tested multiple times, varying c to observe the effects of changing the error correction frequency. It was expected that more frequent error correction would provide more accurate pointing. First, the algorithm was run with $c = 30$, an error correction every 30 seconds. Figures 20 and 21 provide the results of the test for $c = 30$.

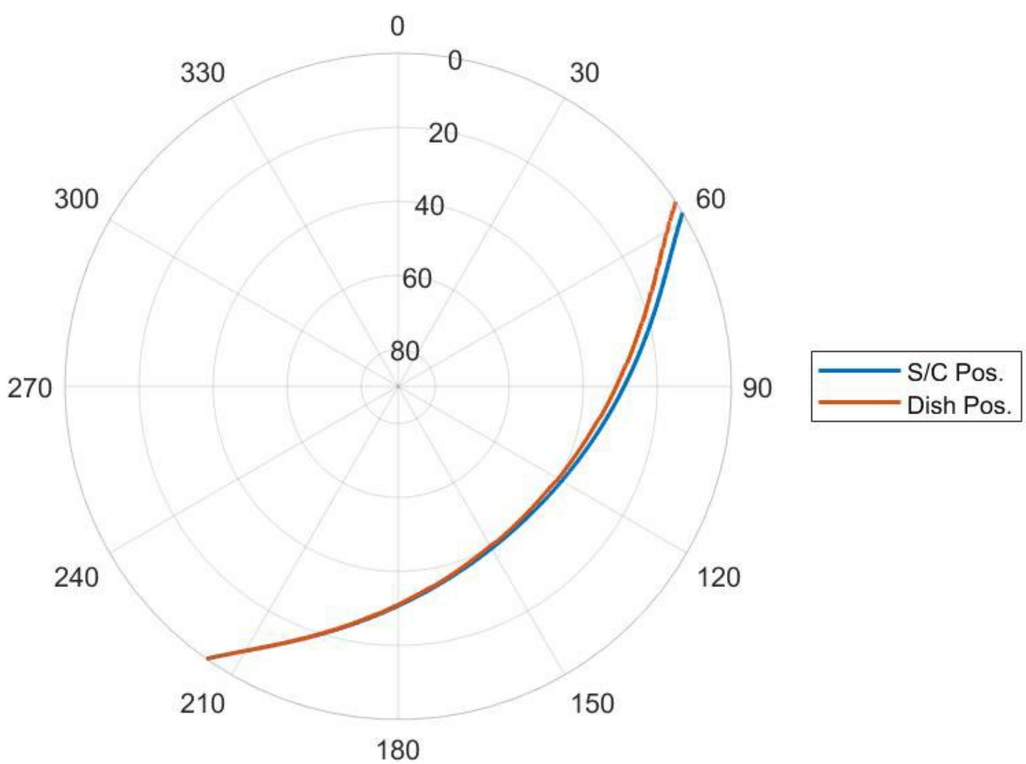


Figure 20. Comparison of satellite and dish trajectories, $c = 30$

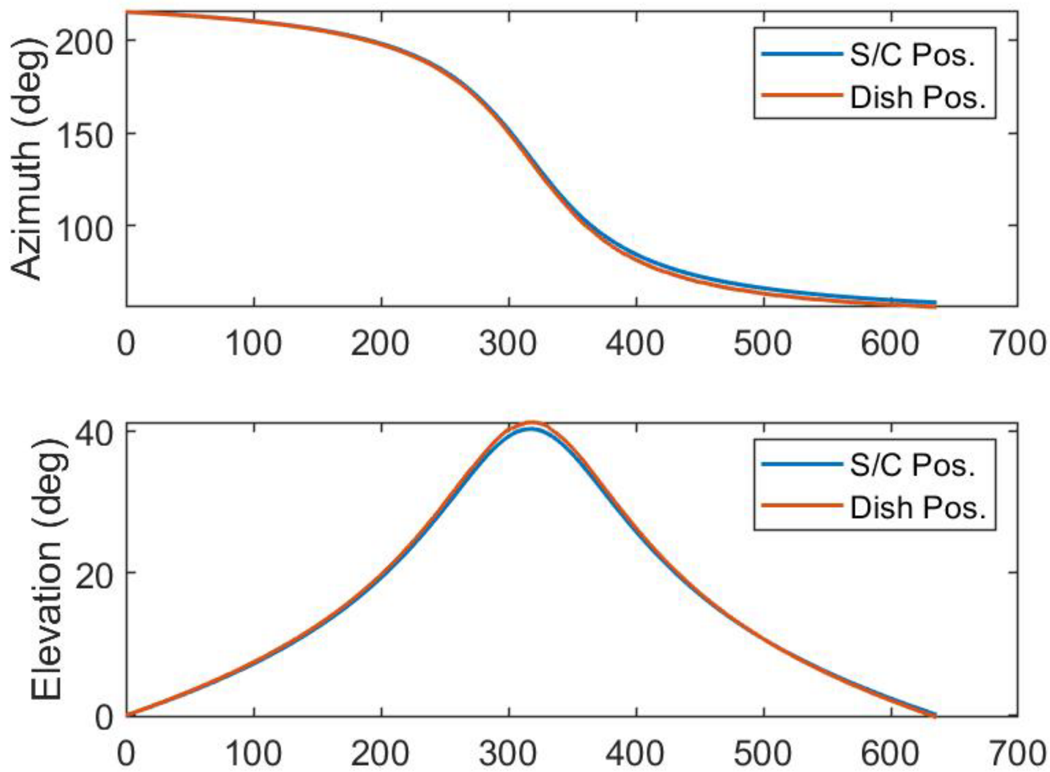


Figure 21. Azimuth and elevation trajectory comparison, $c = 30$

The algorithm bugs of the first iteration algorithm were corrected, and the updated algorithm provided successful error correction. Comparing the above results to the results from no error correction (Figures 16 and 17), it is evident that the feedback control provided more accurate antenna pointing. To further assess the performance of the algorithm, the pointing error was calculated for both azimuth and elevation axes by subtracting the dish's angular position from spacecraft's relative angular position throughout the simulated contact. The error data is pictured in Figure 22. The maximum azimuth error was 3.14 degrees, and the maximum elevation error was 1.03 degrees. Although the pointing error was still relatively large compared to acceptable pointing errors, there was improvement in the algorithm from the first iteration of testing.

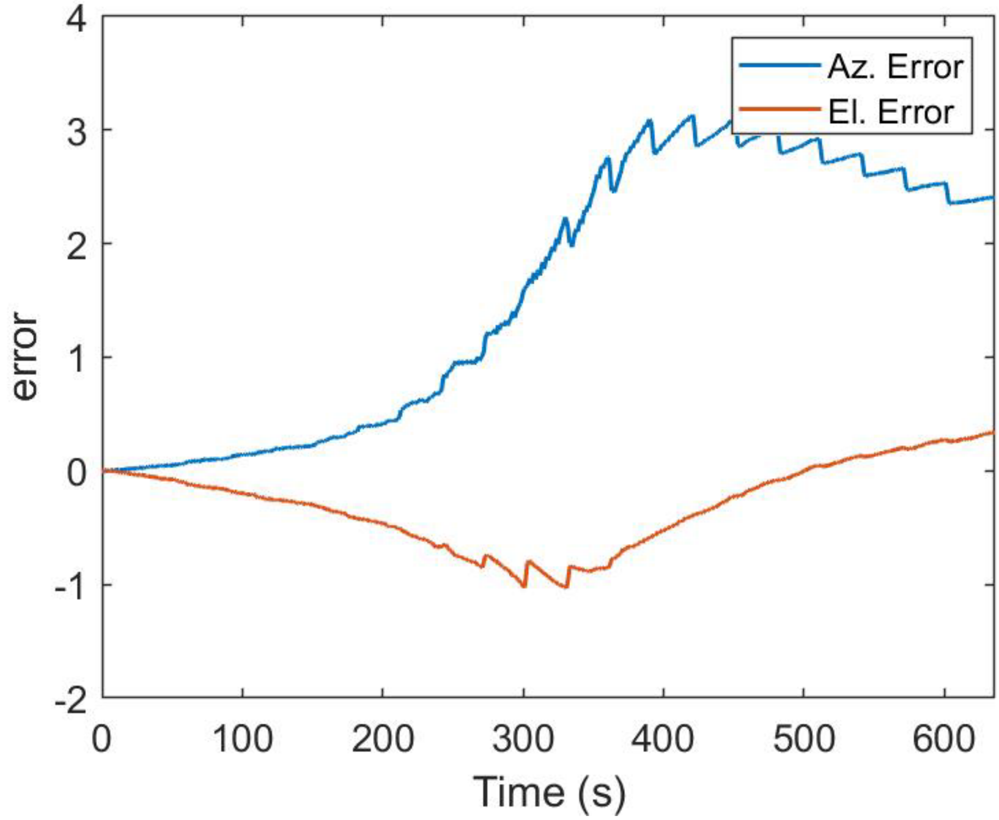


Figure 22. Azimuth and elevation pointing error (degrees), $c = 30$

The test was repeated with error correction every 10 seconds, 5 seconds and 2 seconds. Table 3 summarizes the performance of each test, and Figures 23 through 25 provide the results for error correction every 2 seconds. Comparing the results of error correction every 30 seconds presented in Figures 20 through 22 to the results presented in Figures 23 through 25, the higher sampling rate for error correction provided better tracking performance. The pointing error was reduced by approximately two orders of magnitude when the sampling rate was increased to once every 2 seconds from once every 30 seconds, and the trajectories of the dish and the spacecraft appear to be very closely aligned.

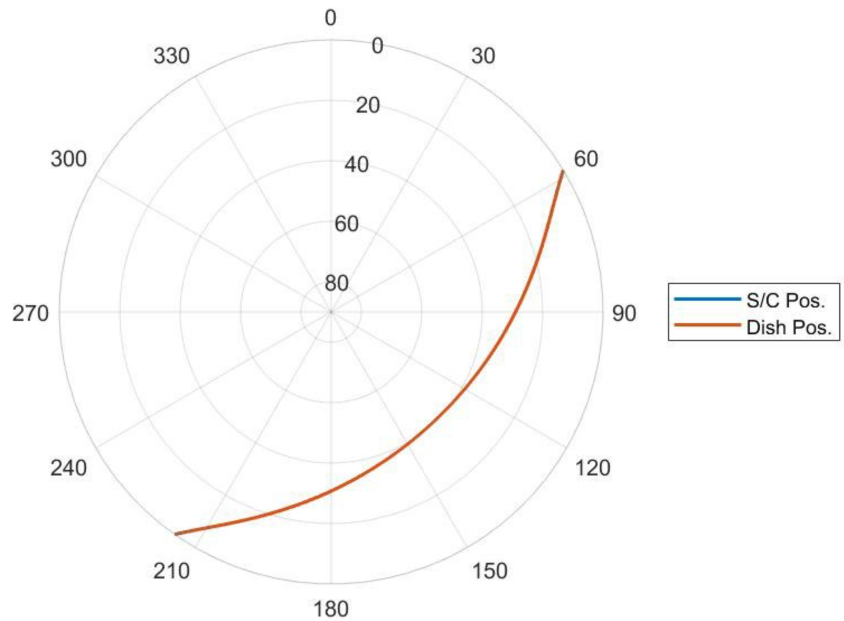


Figure 23. Comparison of satellite and dish trajectories, $c = 2$

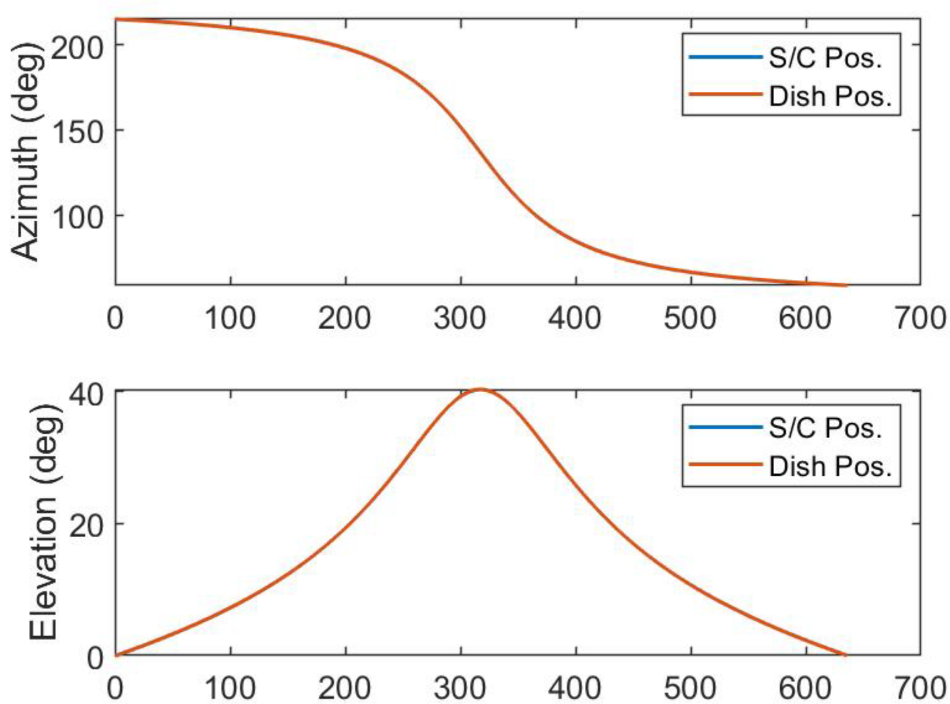


Figure 24. Azimuth and elevation trajectory comparison, $c = 2$

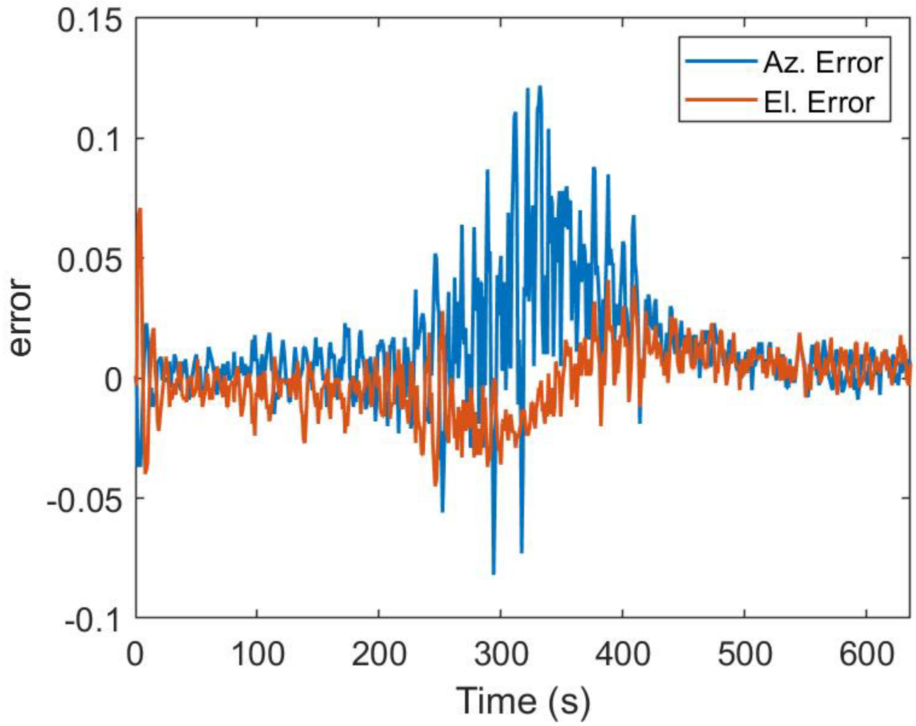


Figure 25. Azimuth and elevation pointing error, $c = 2$

Table 3. Error correction algorithm performance summary

Error Correction Freq. (*c*, sec)	Max. Azimuth Error (deg)	Max. Elevation Error (deg)
30	3.14	1.03
10	1.22	0.32
5	0.34	0.12
2	0.12	0.07

From the above results, increasing the frequency of error correction decreased the pointing error of the dish, as expected. However, when the error correction frequency was increased to once every second, the dish was unstable and moved erratically, similar to the results shown in Figures 18 and 19. It was theorized that the algorithm has some sensitivity to sampling rate, similar to a Nyquist frequency. Because the Az/El position pairs were pulled from STK at a timestep of 1 second, the error correction could not be performed at the same rate with stable results. For the next test of the feedback control algorithm, the

Az/El position pairs from STK were sampled at a higher rate than once per second to further examine the error correction frequency limit. The effects of sampling the position pairs at a higher rate are presented next in Subsection 2.

Overall, the feedback control tracking algorithm was able to successfully correct the antenna's pointing error to a magnitude of approximately one tenth of a degree for a simulated satellite pass. In the next section, the algorithm was tested against an active satellite to further prove its functionality.

2. Testing on Active Satellite

Testing the feedback control tracking algorithm on an active satellite required some further developments. Coriolis was chosen as the satellite to track because it has a constant S-band downlink telemetry, tracking, and control (TT&C) transmission, which is compatible with the S-band dish antenna being controlled at NPS. Coriolis is a joint mission between the NRL and Air Force Research Laboratory (AFRL) that was launched in 2003 with instruments to improve space-based observation of surface winds on Earth. It is in a circular sun-synchronous orbit—meaning it maintains the same orientation with respect to the sun with a revisit time of approximately 12 hours—at 840 kilometers altitude [21].

The NPS ground station was modeled in STK using its coordinates and elevation. Coriolis was added to the STK scenario using its standard object database. Its orbit was propagated into the future to determine possible access times to the NPS ground station. Figures 26 and 27 show the 3-D and 2-D ground track of the contact chosen to run the test. This contact was chosen because it passed nearly overhead of the NPS station had a long contact duration. The Az/El relative angular position of Coriolis with respect to NPS was pulled from STK with a timestep of 0.2 seconds, a higher sampling rate than what was used for the simulated satellite pass.

Figure 26. 3-D contact visualization of Coriolis's contact with NPS ground station

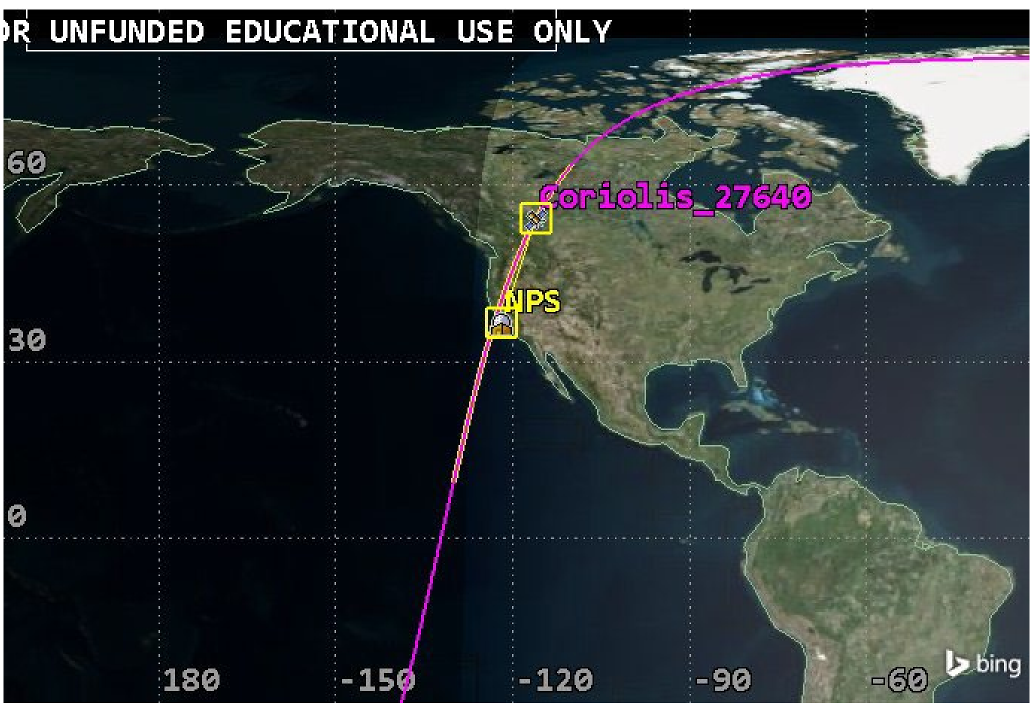


Figure 27. 2-D ground track of Coriolis-NPS contact

Figure 28 shows the trajectory of Coriolis with respect to the NPS ground station. It was observed that the azimuth position crosses over 0 degrees. The NPS dish is limited azimuth positions of -5 degrees to 385 degrees. Since velocity commands are being used in the tracking algorithm, Coriolis's trajectory will cause the dish to be commanded to an azimuth that is outside of the position bounds. This will cause the gearbox of the rotator to lock. To mitigate gearbox lockout, the trajectory was manipulated to not cross 0 degrees by reversing both the azimuth and elevation trajectories by 180 degrees. By reversing both axes, the trajectory of the dish is physically the same as the trajectory shown in Figure 28, but avoids gearbox lockout by not crossing 0 degrees azimuth, as shown in Figure 29. The original and manipulated trajectories appear to be mirror images, but the elevation axis ranges from 90 degrees to 180 degrees in the manipulated trajectory. This causes the dish commands to steer clear of the azimuth limits while still following Coriolis's overhead trajectory.

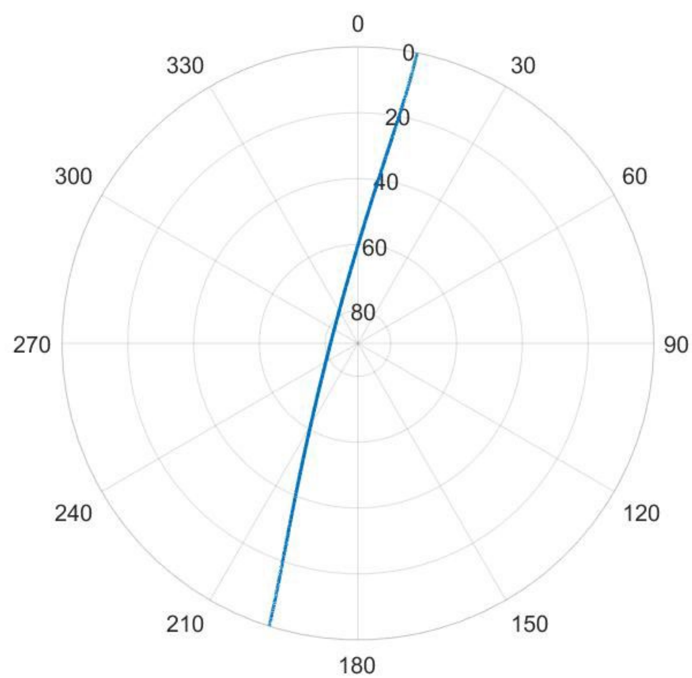


Figure 28. Original Coriolis trajectory with respect to the NPS ground station

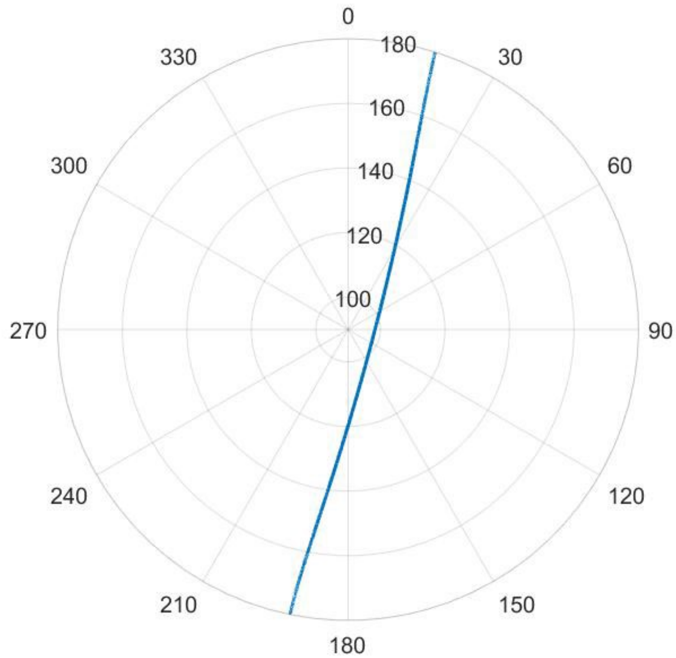


Figure 29. Manipulated Coriolis trajectory to avoid gearbox lockout

The main modification of the tracking algorithm was implementing the real-time aspect that is required for an active satellite pass. The final iteration of the tracking algorithm MATLAB script is included in the Appendix. The associated timestamp of each Az/El position pair was converted to local time. The iterative portion of the MATLAB code began when the local time was equal to the first timestamp of the STK-generated Coriolis access data and ran until the local time was equal to the final timestamp of the contact. The error correction was executed the same way as presented in the previous section. The satellite's relative angular position was pulled from the STK-generated access data by finding the Az/El position pair that was nearest to the local time using MATLAB's "find" function. The satellite's position was then compared to the position of the dish at that time to calculate the pointing error. The algorithm was initially tested prior to the pass with $c = 2$, an error correction once every two timesteps (0.4 seconds). Again, the dish moved erratically. It was found that an error correction once every second (5 timesteps, $c = 5$), provided smooth dish commanding. Although sampling the Az/El position pairs at a higher rate did provide a higher feasible error correction frequency than the simulated satellite test, the maximum error correction frequency relative to sampling frequency relationship was inconclusive. Further research and testing are required to determine this relationship.

The educational ground station was configured the same way as the simulated satellite test pass presented in the previous section. A spectrum analyzer was connected to the RF path of the antenna to visualize the received transmission and confirm successful tracking of Coriolis. The spectrum analyzer was set to a center frequency of 2.2125 GHz, as this is the S-band downlink frequency transmitted by Coriolis. Figures 30 through 32 present the tracking results of the Coriolis contact.

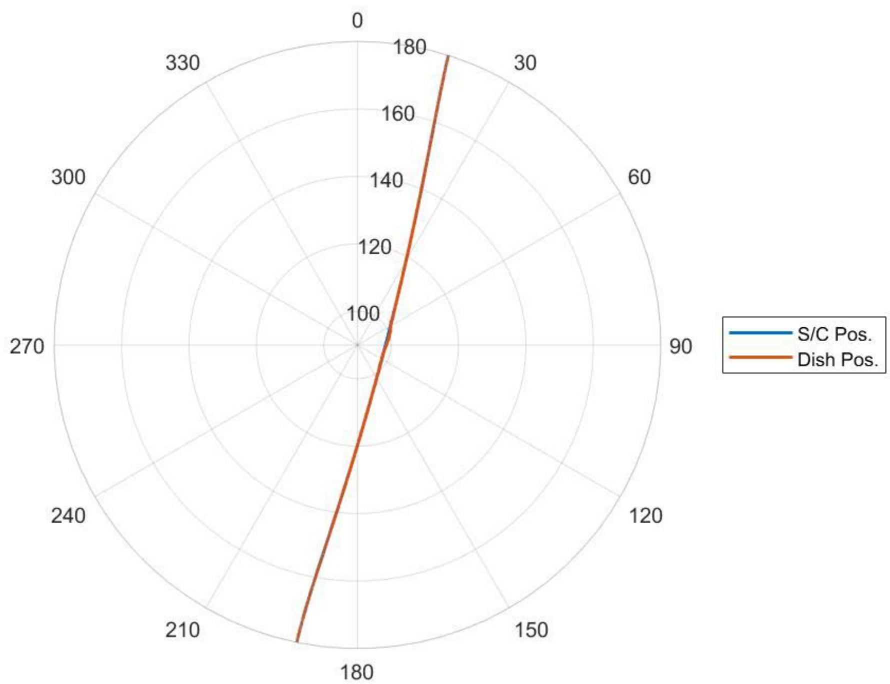


Figure 30. Comparison of Coriolis and dish trajectories

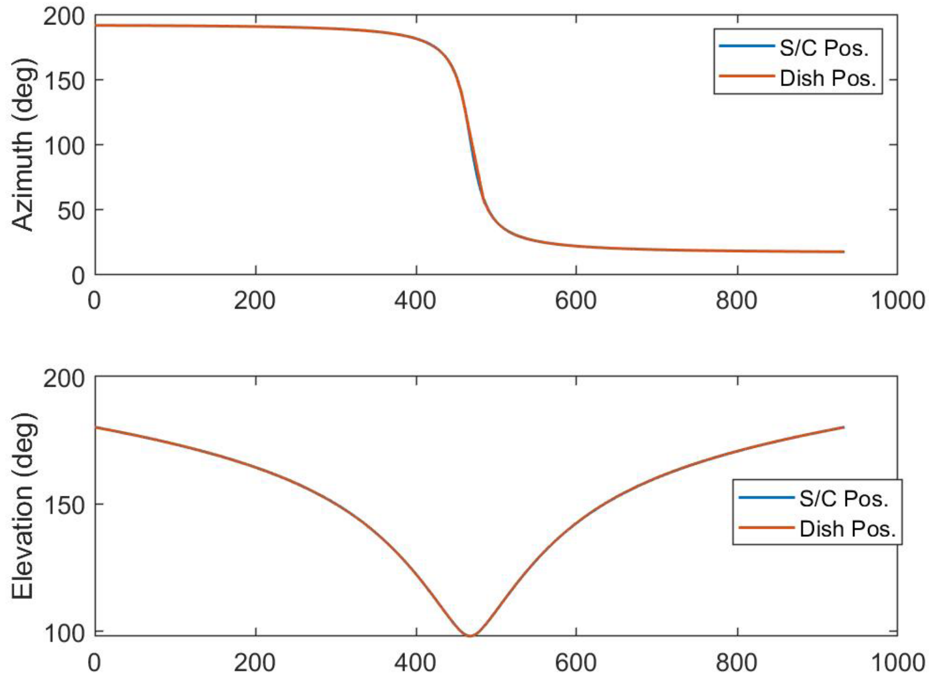


Figure 31. Azimuth and elevation trajectory comparison for Coriolis tracking

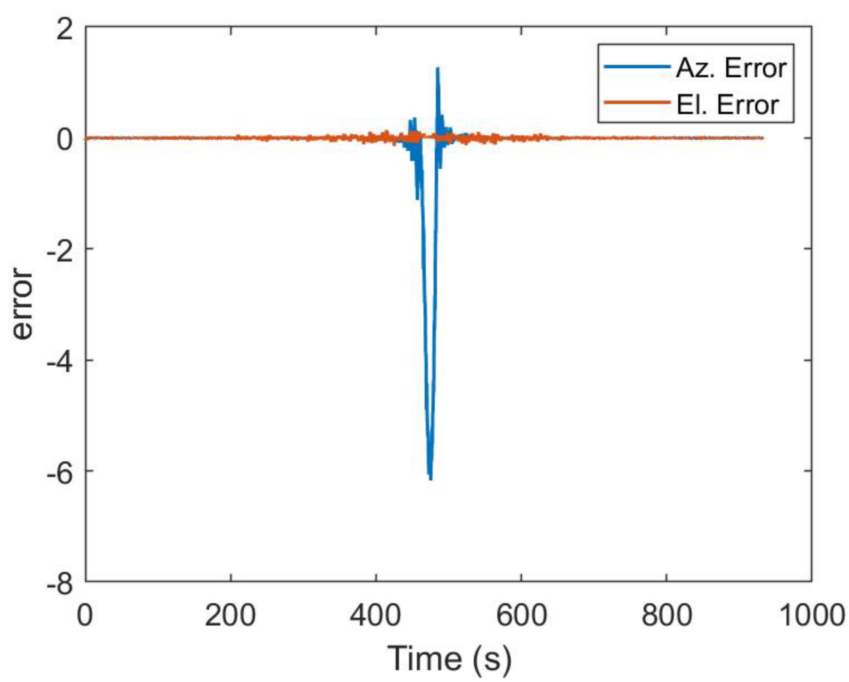


Figure 32. Azimuth and elevation pointing error for Coriolis tracking

From Figure 32, the tracking algorithm appears to be performing well until Coriolis passes overhead, where there is a large negative pointing error. This is due to a common phenomenon known as the keyhole effect. When a satellite passes overhead, or nearly overhead, it is moving its fastest in the sky with respect to the ground station. The azimuth is relatively constant until it passes overhead and then quickly flips to the opposite azimuth. This is evident when observing the azimuth trajectory from Figure 31. The large negative error is due to the fact that the dish lags behind the satellite as it passes overhead because the satellite is moving faster than the dish is capable of moving. Figure 33 highlights this observation. As Coriolis passes overhead, the dish hit its azimuth velocity limit, and remained there for 18.8 seconds until it was able to catch up to Coriolis's trajectory.

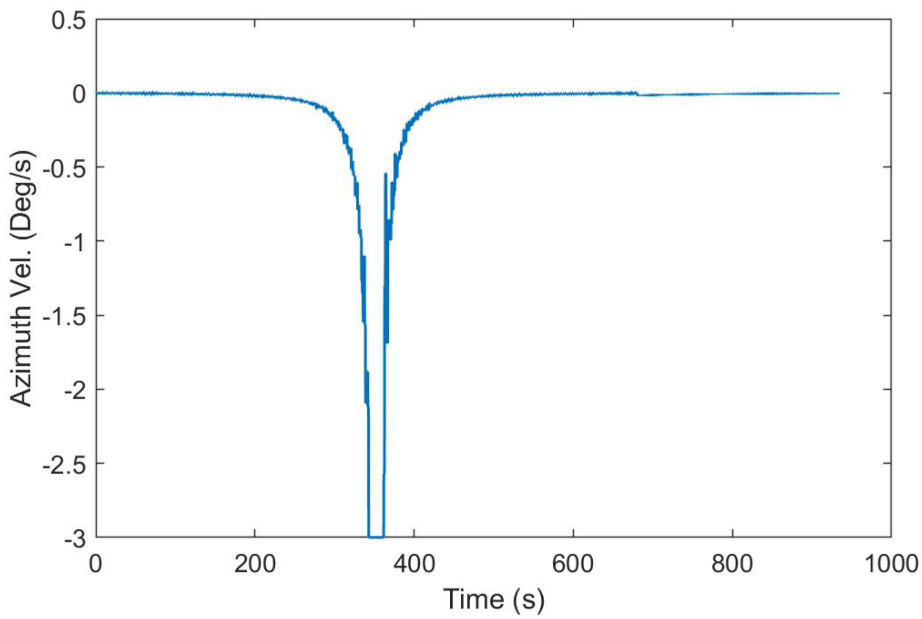


Figure 33. Dish azimuth velocity for Coriolis tracking

To correct for the keyhole effect, the dish's planned trajectory can be modified for overhead or near overhead satellite passes. Instead of having the dish elevation rise and fall as it does in Figure 31, the azimuth can be kept constant and the elevation continue to increase past 90 degrees, effectively tracking the satellite as it passes overhead. The dish would not lag behind the satellite because the change in elevation is more gradual than the change in azimuth when the satellite crosses overhead. Further work and testing are required to implement keyhole correction into the tracking algorithm.

Figure 34 provides a magnified view into the pointing error results from the Coriolis contact. The azimuth and elevation pointing errors stayed well within +/- 0.1 degrees for most of the contact, neglecting the keyhole effect. From this test, no conclusion can be drawn on the relationship between increasing the error correction frequency and pointing error because of the occurrence of the keyhole effect.

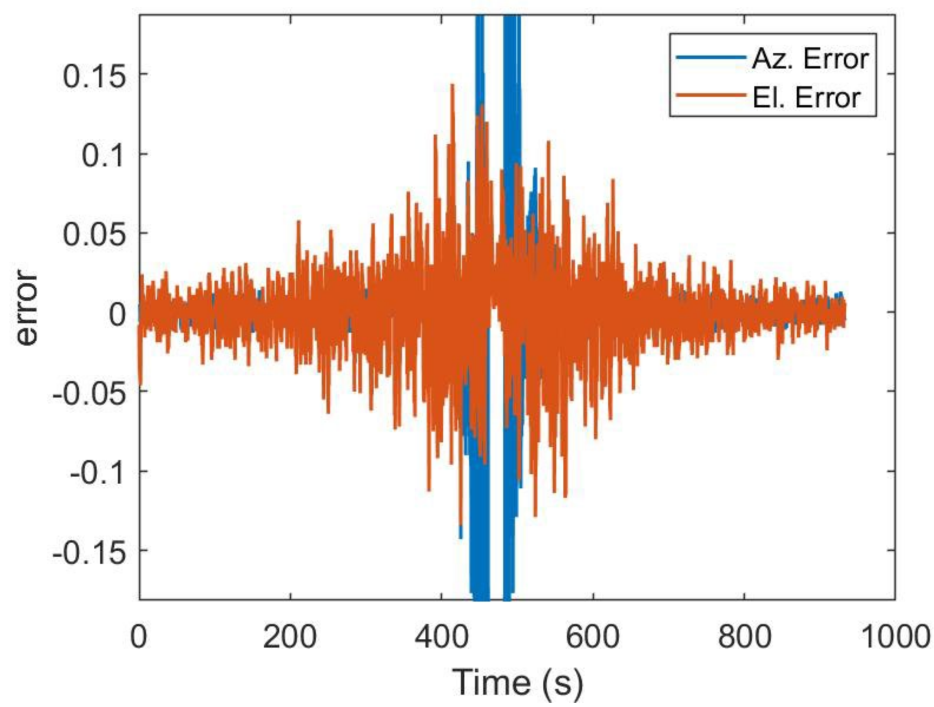


Figure 34. Magnified pointing error for Coriolis tracking

Figure 35 shows a screenshot of the spectrum analyzer during the Coriolis contact. The educational station successfully received RF signals from Coriolis throughout the duration of its pass as shown by the signal peak in the spectrum analyzer, further validating the feedback control tracking algorithm. The strength of the signal grew until the Coriolis passed overhead at its closest point of approach, where the signal strength peaked at -

35dBm, approximately 50 dBm above the noise floor. The signal strength then decreased as Coriolis moved further away to the horizon.

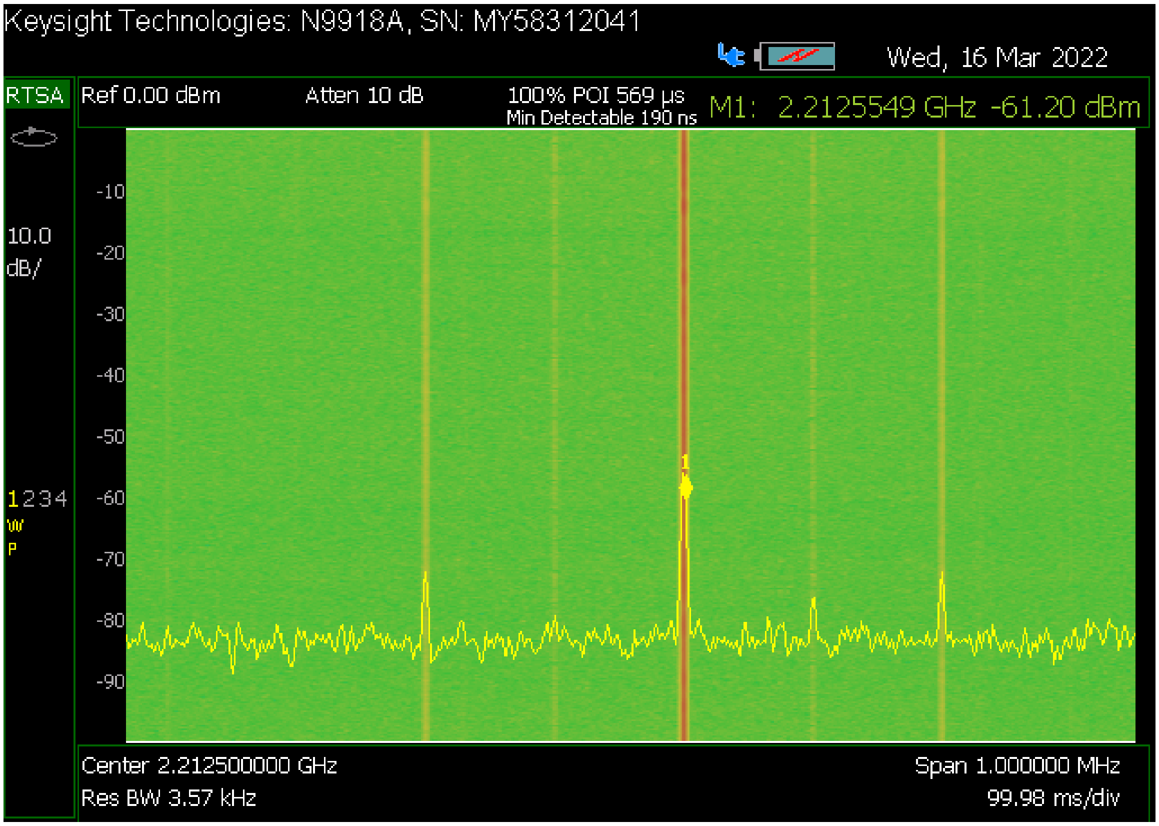


Figure 35. Spectrum analyzer showing receipt of Coriolis transmission

An interesting observation of the Doppler Effect was made during the contact as well. As Coriolis was ascending towards the ground station, the received frequency was consistently higher than the actual transmission frequency due to its forward velocity towards the antenna. After Coriolis passed overhead and was descending away from the antenna, the received frequency was consistently lower than Coriolis's transmission frequency.

The above test of the tracking algorithm was repeated with multiple active satellite passes to test the resiliency of the tracking algorithm as well as better observe the pointing accuracy of a pass without the keyhole effect. Coriolis was again tracked, but this pass was chosen such that it would not pass nearly overhead to avoid the keyhole effect. Figure 36 shows the 2-D ground track of the chosen contact from STK. From Figure 36, it can be observed that the trajectory does not cross from 0 to 360 degrees azimuth during the contact and thus does not need to be corrected to avoid gearbox lockout as was done for the first Coriolis track.

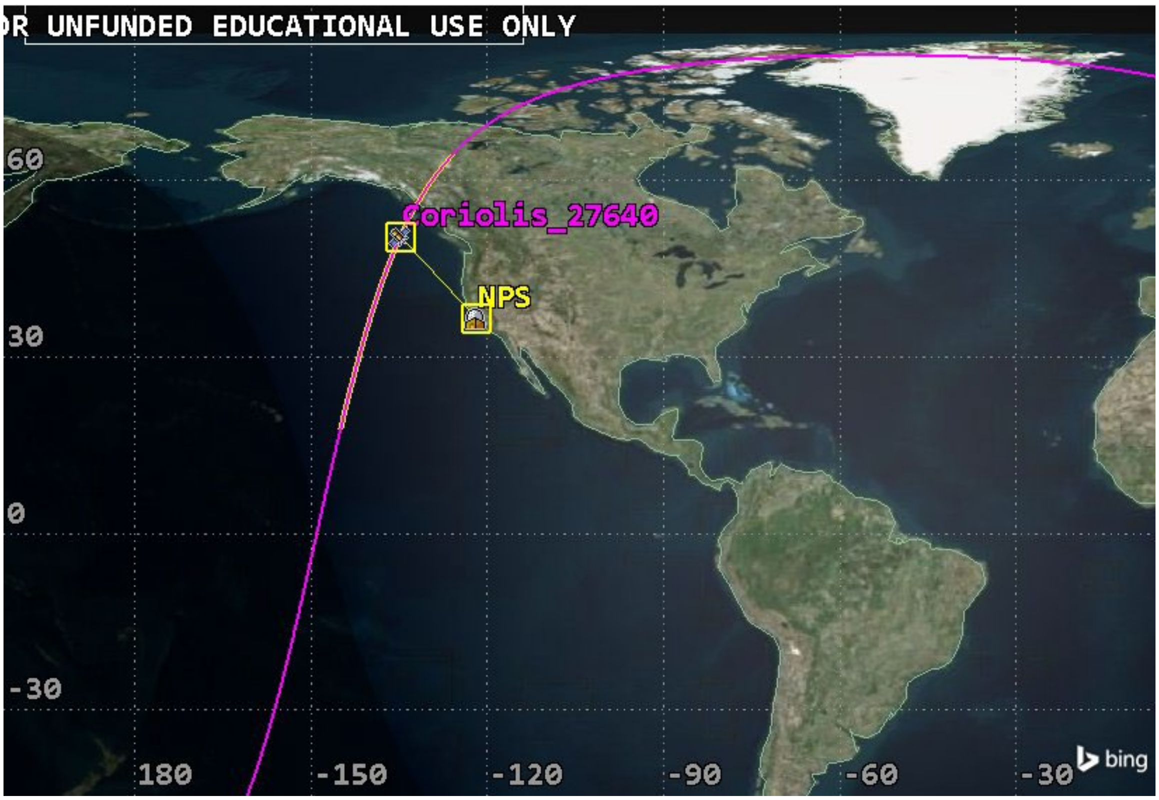


Figure 36. 2-D ground track of second Coriolis-NPS contact

As was done in the first Coriolis track, an error correction sample frequency of once per second (*c=5*) was used. Figures 37 through 39 show the results of the test.

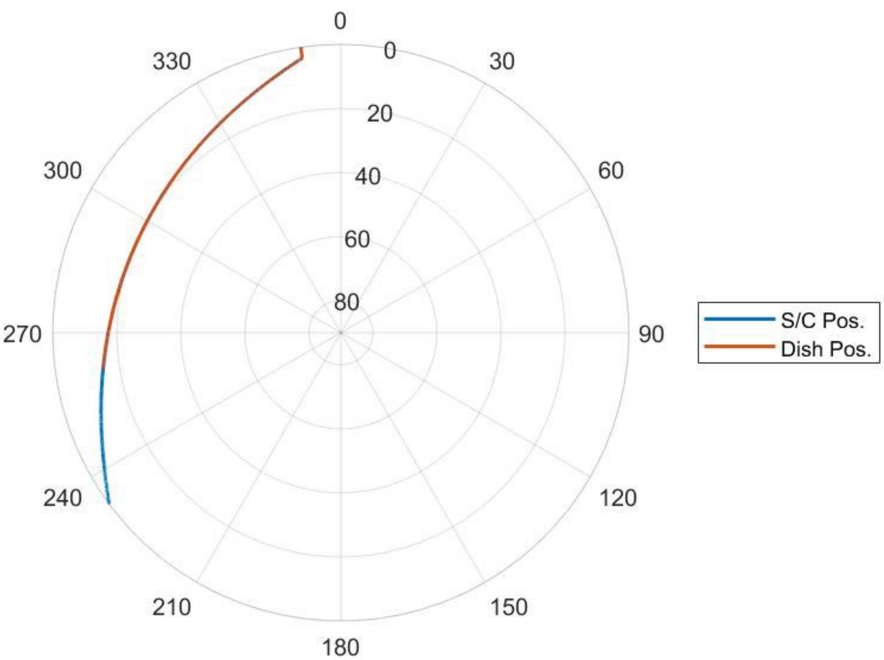


Figure 37.   Comparison of Coriolis and dish trajectory, trial #2

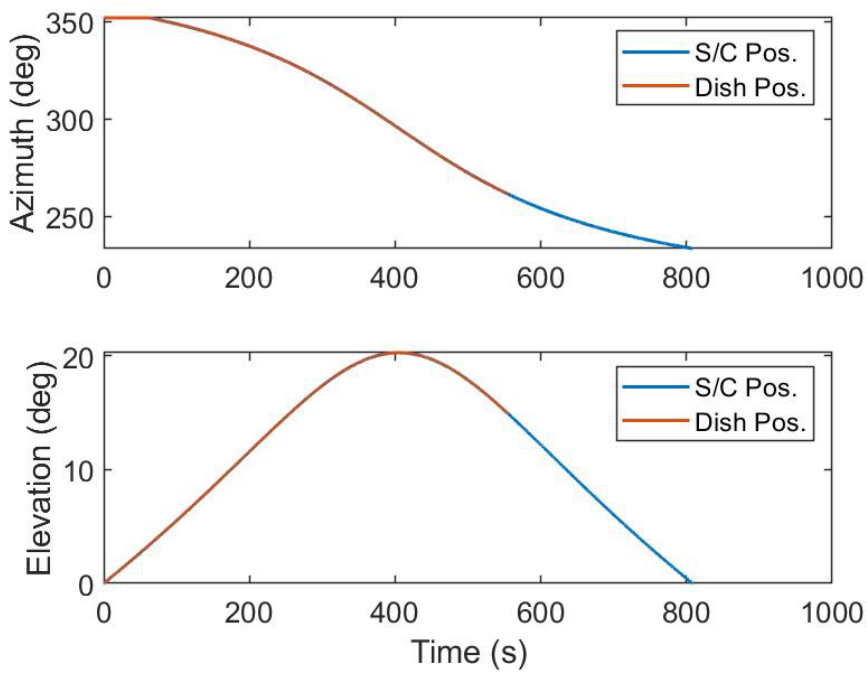


Figure 38.   Azimuth and elevation trajectory comparison for Coriolis contact #2

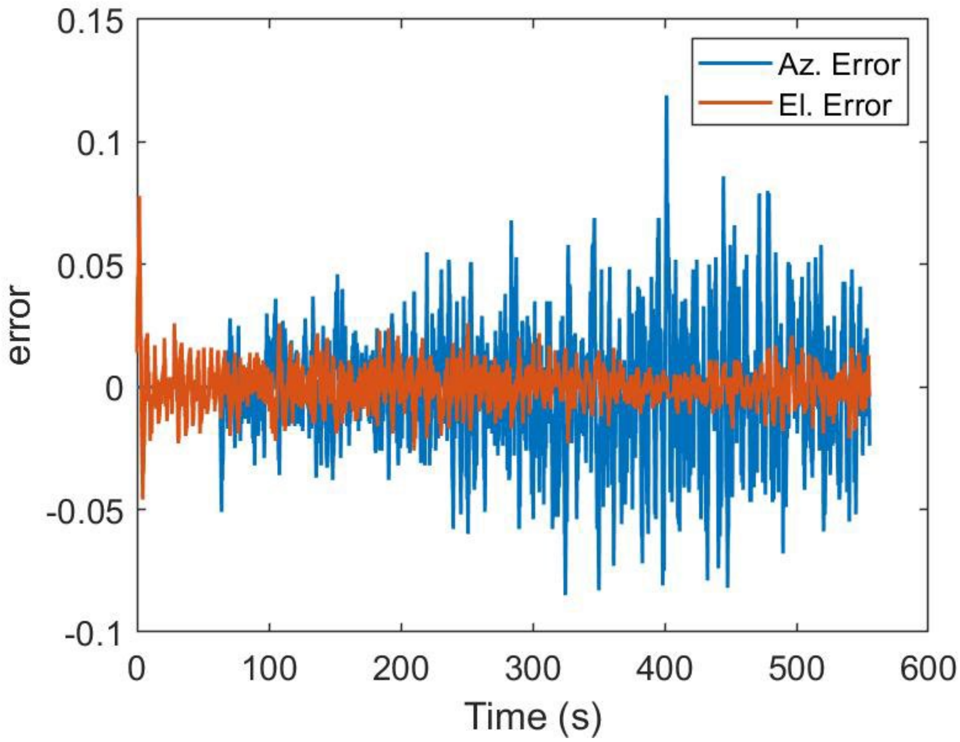


Figure 39. Azimuth and elevation pointing error for Coriolis contact #2

The test provided some interesting insights regarding the tracking algorithm. Before the pass began, the dish had difficulty moving to the initial azimuth of 356 degrees. When commanded to preposition at the initial azimuth, the dish moved to -4 degrees azimuth rather than 356 degrees, which would cause the dish to move clockwise once the pass began to intercept the satellite's relative azimuth or gearbox lockout. It was discovered that there had been upper limits imposed on the ACU of 354 degrees. To correct for this, the algorithm was modified to point at a maximum of 354 degrees if the satellite's relative azimuth was greater than this amount. This can be observed at the beginning of the pass from Figures 37 and 38 as there was constant dish azimuth for approximately 40 seconds. This test revealed a bug in the ACU's settings that can be easily corrected by updating the positional limits of the dish in the ACU settings.

Comparing Figure 39 to the pointing error results of the previous tests, the second Coriolis contact demonstrated the lowest absolute average pointing errors at 0.017 degrees for Azimuth and 0.0064 degrees for elevation. However, the maximum pointing errors in azimuth and elevation were consistent with the best results from the simulated satellite tracking at an error correction frequency of once every 2 seconds. It can be concluded, then, that increasing the error correction frequency provides lower average pointing errors but not necessarily lower maximum pointing errors. It appears that the minimum achievable pointing errors are limited by the physical performance of the ACU.

At approximately 575 seconds into the pass, an unexpected error occurred in the MATLAB script regarding its tcpclient function. The error stated that the expected length of data was not being received from the dish, causing the TCP connection to timeout. Therefore, the dish position status was unable to be queried, and the error correction could not be implemented for the remaining quarter of the pass. This is evident in Figures 37 and 38 as the dish trajectory does not continue at this point. This was a peculiar error because the tracking algorithm code had not been changed from the previous Coriolis track, besides updating the satellite's relative trajectory. Further contacts were tested to investigate the nature of this error.

As was done in the first Coriolis tracking test, a spectrum analyzer was connected to the dish receiver to visualize the incident RF downlink from the spacecraft. Figure 40 presents a screenshot from the spectrum analyzer during the pass. The received RF data shown by the spectrum analyzer further validates the receive capabilities of the educational station, as well as the performance of the tracking algorithm. Consistent behavior was observed from the spectrum analyzer relative to the first Coriolis contact in regards to signal strength and doppler effect.

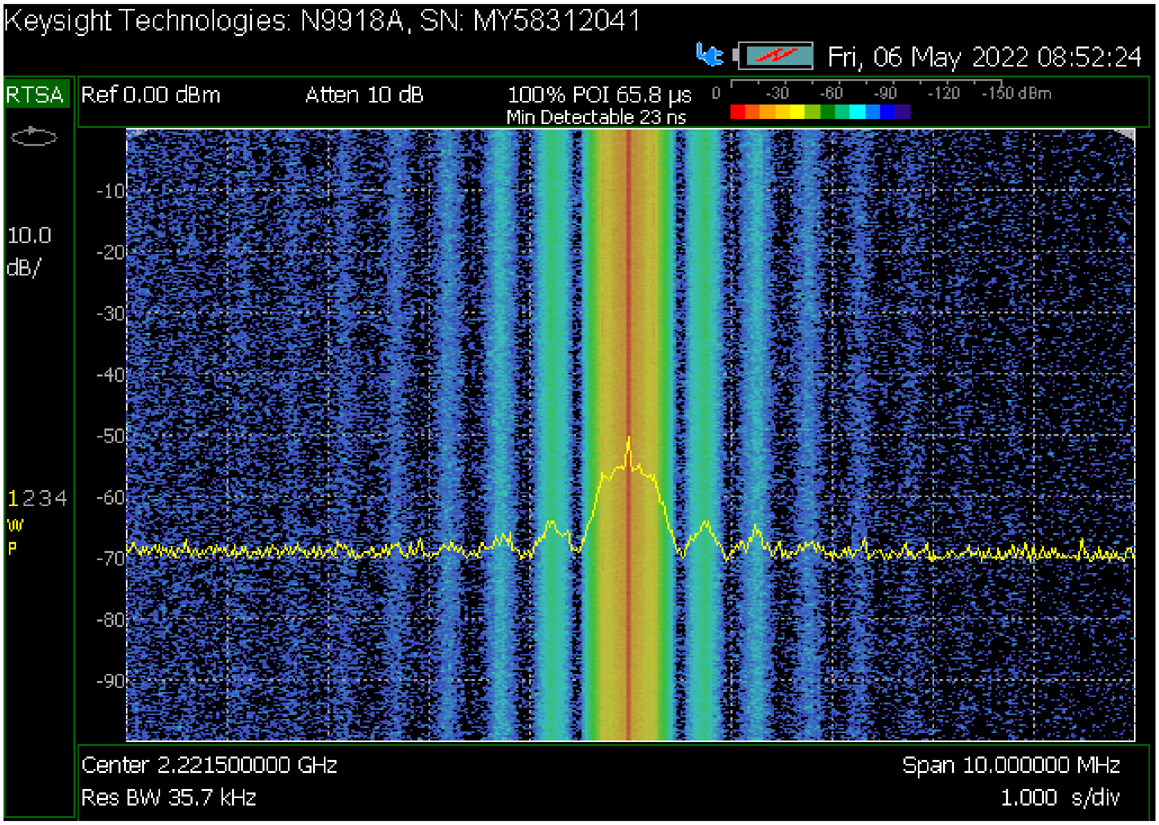


Figure 40. Spectrum analyzer screenshot from Coriolis contact #2

The active satellite tracking test was repeated for a different satellite to provide more data regarding the operation of the new tracking algorithm. The NASA AIM satellite was chosen to be tracked as it had a downlink frequency in the S-band, compatible with the NPS dish. Figure 41 shows the 2-D ground track of the contact from STK.

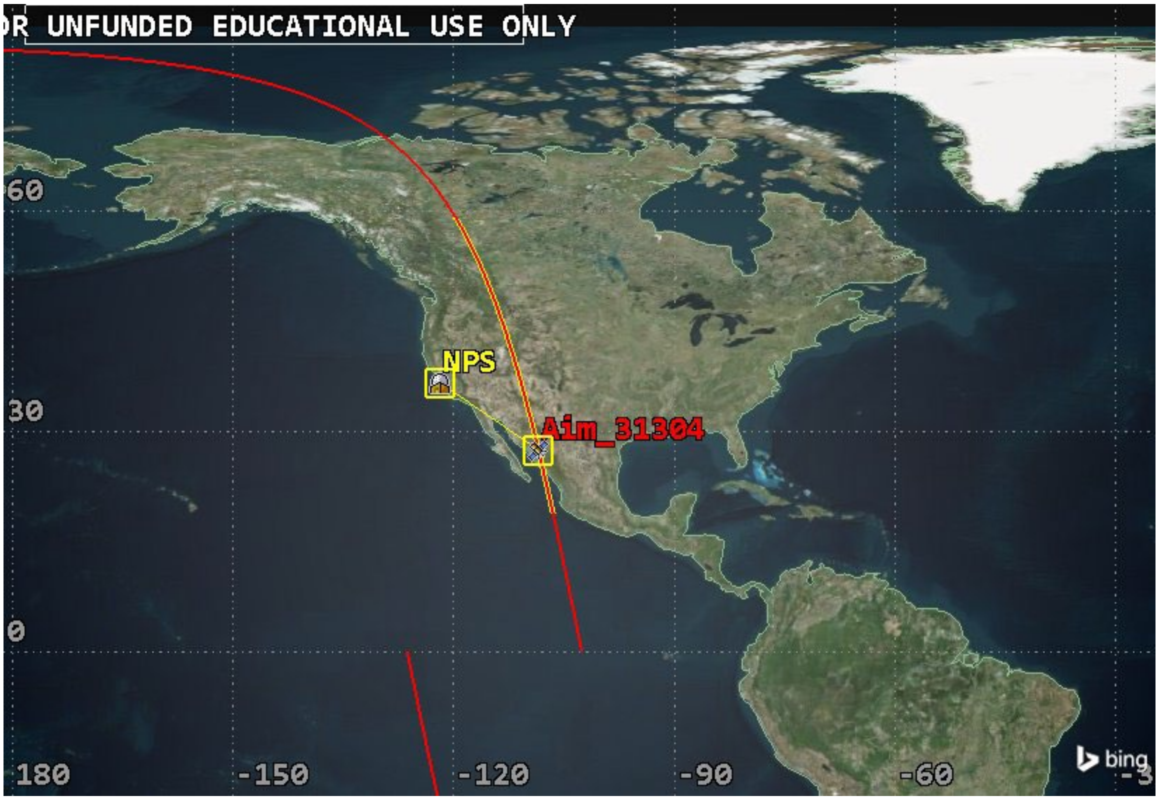


Figure 41. 2-D ground track for AIM-NPS contact

The algorithm and test methodology were kept the same as the previous active satellite tracking tests to validate the consistency and repeatability of the algorithm. Figures 42 through 44 show the results of the AIM tracking test.

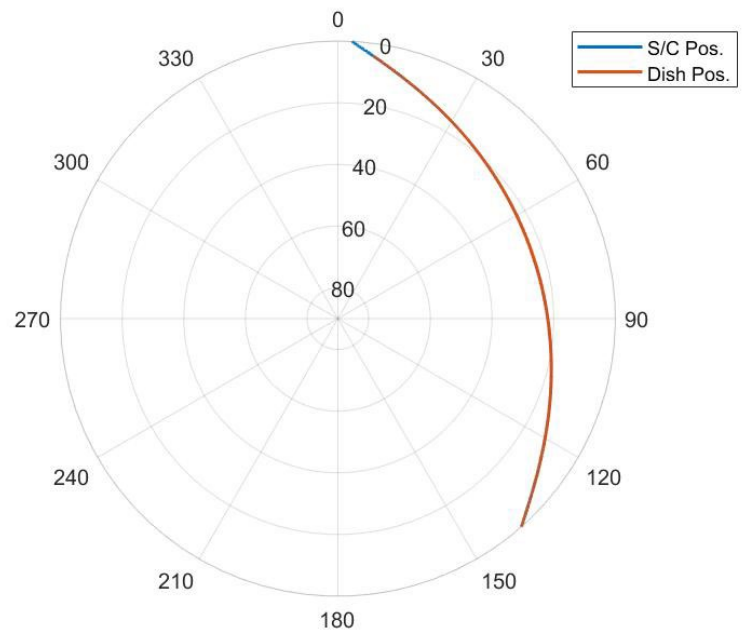


Figure 42. Comparison of AIM and dish trajectory

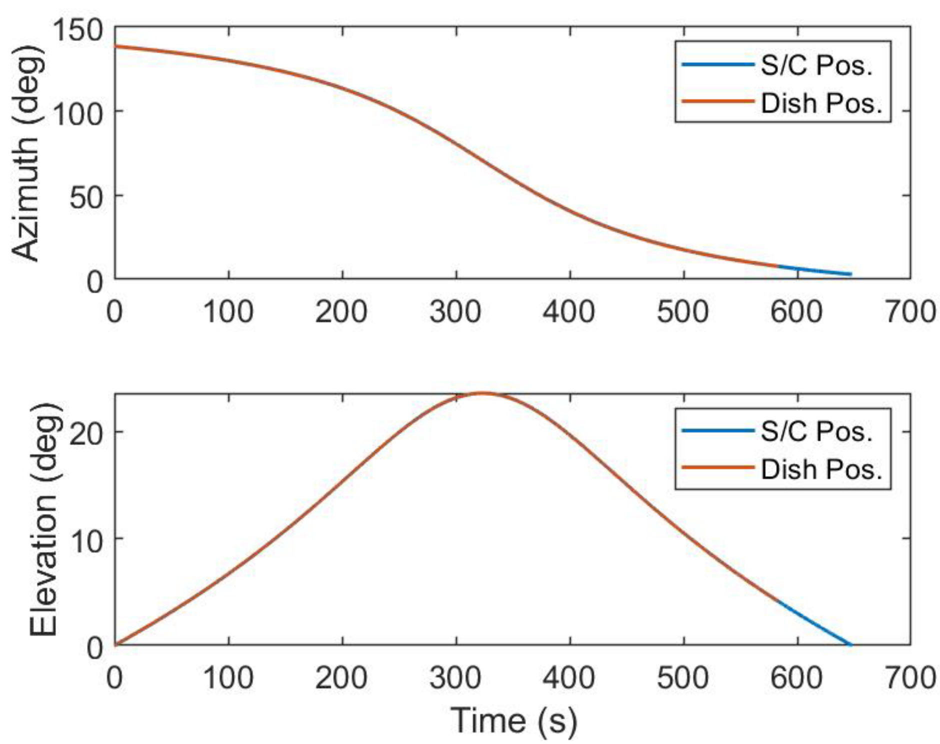


Figure 43. Azimuth and elevation trajectory comparison for AIM contact

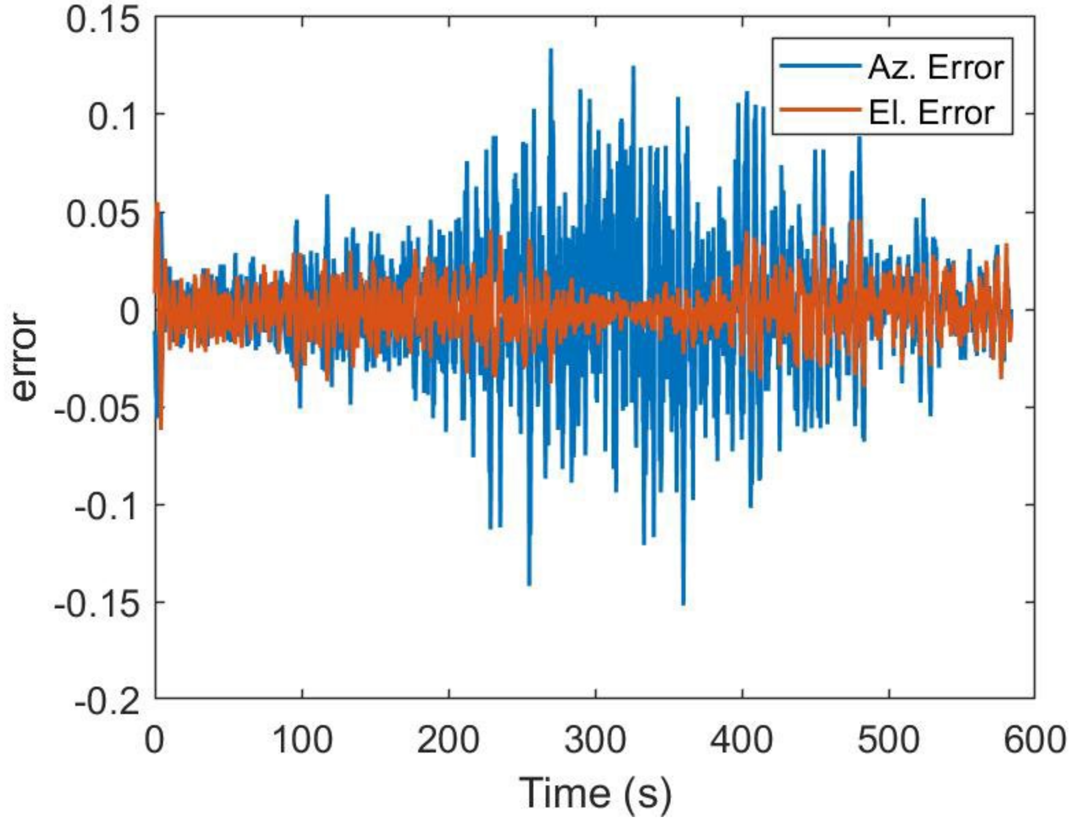


Figure 44. Azimuth and elevation pointing error for AIM contact

The average absolute pointing error and maximum absolute pointing error of the AIM contact was comparable with the results from the second Coriolis contact. The average absolute error was 0.024 degrees for the azimuth axis and 0.0096 degrees for the elevation axis, and the maximum absolute pointing error was 0.152 degrees and 0.062 degrees for azimuth and elevation axes, respectably. From the repeatability of the results, the pointing errors for the velocity tracking algorithm are predictably within the range of +/- 0.15 degrees. Given that the NPS S-band dish has a nominal beam width of 3 degrees, the tracking algorithm's pointing errors are well within the acceptable range to ensure communications with LEO satellites. This is supported by the consistent received signal as demonstrated by the spectrum analyzer for the active tracking tests. Figure 45 presents the received signal from AIM.

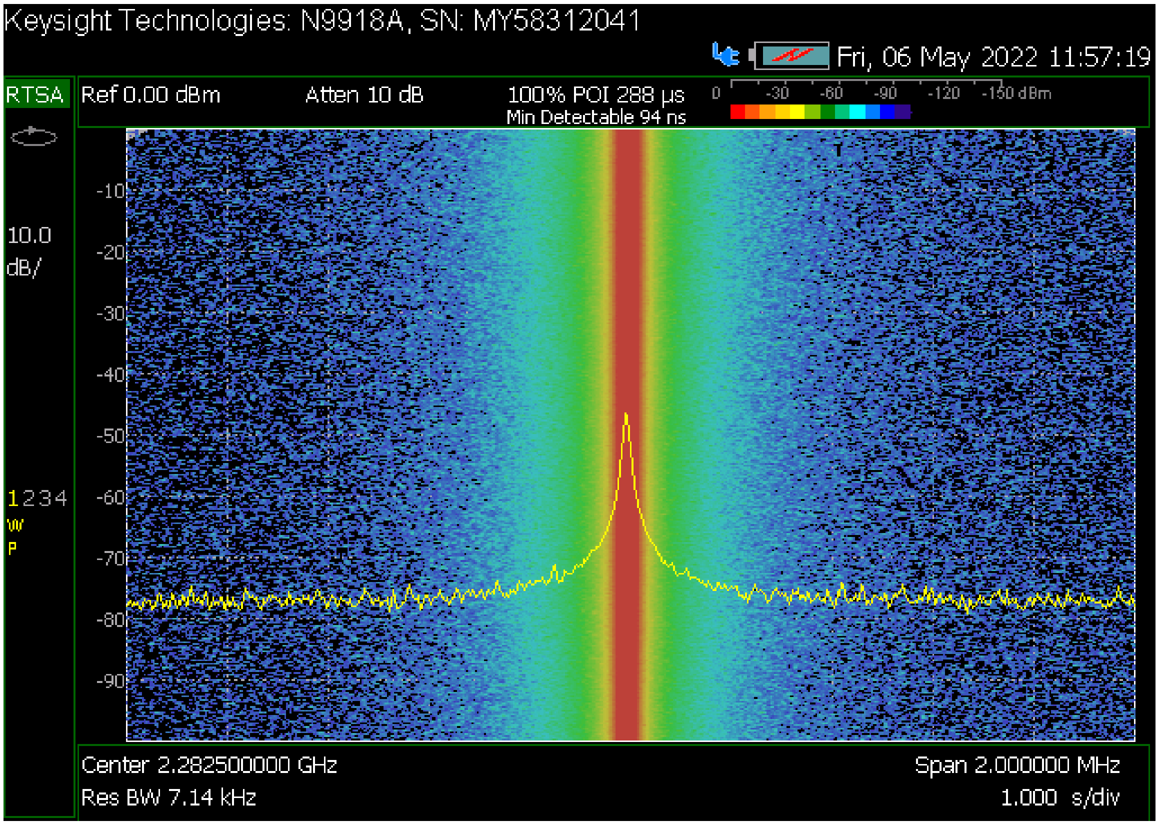


Figure 45. Spectrum analyzer screenshot from AIM contact

During the AIM contact, the same MATLAB error was reported from the second Coriolis contact. Similar to that contact, the algorithm stopped receiving the dish status from the ACU towards the end of the pass, as is evident in Figures 42 and 43. In researching the error message, it appeared that the ACU was sending status messages too quickly—faster than once per second—for the algorithm to keep up with. It is suspected that this caused the receive connection from the ACU to MATLAB to timeout and cause the error. In order to mitigate this error while also keeping the error correction frequency high, the tracking algorithm needs a buffer array that receives and stores all the data from the ACU before it is further acted upon by the tracking algorithm. This constitutes future work to make the tracking algorithm more resilient before it can be considered for operational use by the MC3 network.

The above tests demonstrated the value of the parallel educational station for purposes of research and development of the MC3 operational network, particularly in the tracking algorithm. The full capabilities of the ACU used by MC3, notably velocity control, have not yet been implemented into practice. Velocity control provides smoother tracking of satellites and has the potential of increasing the pointing accuracy of the antenna, which is crucial to MC3 as it becomes a higher performing and heavily relied upon ground network. The velocity feedback control tracking algorithm was developed and tested, successfully tracking an active satellite in orbit with pointing errors averaging in the magnitude of hundredths of a degree. There is interest from the MC3 operators and software engineers to iterate upon the algorithm to make it a suitable replacement for the current operational tracking algorithm. The development and testing that lead to the demonstration of this concept would not have been possible without the use of the parallel ground station.

C. TEST #3: OVER-AIR TEST OF NPS SMALL SATELLITE COMMUNICATIONS HARDWARE

The purpose of this test was to demonstrate the operational benefits of the educational station by performing communication testing of a small satellite in development. This test was also used to verify the transmit and receive capabilities of the educational station. Furthermore, this test demonstrated the capability of the educational station to operate in a complete standalone mode with the full functionality of the SATRN software.

NPS is currently developing a small satellite named Mola to be launched in the spring of 2023. Mola will use MC3 as its ground communications network. As part of its functional testing, Mola requires over-air testing of its communications hardware to ensure it is working properly prior to launch. Mola will use SDRs for TT&C communications with the ground using the S-band frequency. The SDRs used by MC3 and their satellites utilize a software called GNU Radio to create flowgraphs that convert data between digital and RF signals. These flowgraphs are uploaded to the SDRs during a satellite contact to enable communication. As mentioned in Chapter III, SDRs are very beneficial to MC3 as a low-cost RF asset because they are flexible between many different satellites and their associated communications protocols. However, before a flowgraph can be implemented on the operational MC3 network, it must be scanned for cybersecurity purposes, which is a lengthy and cumbersome process. This obstacle could potentially delay the spacecraft's ground segment integration schedule. To avoid the schedule delay, the over-air testing can be accomplished prior to the official approvals through the educational parallel station. The educational station provides a flexible alternative for integration testing as it is not burdened with the approval process of the operational network.

To accomplish the over-air testing, Mola's development kit was used since the full satellite has not been constructed at the time of writing. The development kit consists of Mola's flight computer and S-band SDR, as well as the proper connections between the two. The development kit uses a bench power supply to provide power to the simulated satellite, also known as a FlatSat, since it is a simple set up that incorporates the actual satellite hardware. Two virtual machines (VMs) were also required to perform the over-air test. One VM provided access to SATRN, which was necessary to configure the test pass between the ground station and the FlatSat as well as load the proper flowgraph to the SDR. The other VM served as the simulated missions operation center for the FlatSat, which sends and receives commands through SATRN to communicate with the satellite.

Before over-air communication was tested, the configuration was first tested through a wired connection between the ground SDR and the FlatSat SDR. The ground SDR was connected to a laptop, which served as the educational station CPU and ran the necessary VMs. Figure 46 provides a visualization of the wired test setup. Data was sent from the FlatSat flight computer through its radio to the ground SDR running the proper GNU Radio flowgraph. The data was received by the FlatSat VM, the simulated MOC, but most of the packets were dropped. This indicated that the physical data transfer from the FlatSat to the VM was successful, but the laptop that was running the VM did not have enough computing power to decode the packets being received. The wired communication test was performed again, except the laptop was replaced by a desktop computer with more computing power. No packets were dropped, indicating a functional configuration. The configuration was further tested by running a test pass in SATRN Client through the SATRN VM on the desktop computer. Uplink and downlink to the FlatSat through the wired connection was verified through the SATRN test pass. The test setup was thus ready for over-air testing.

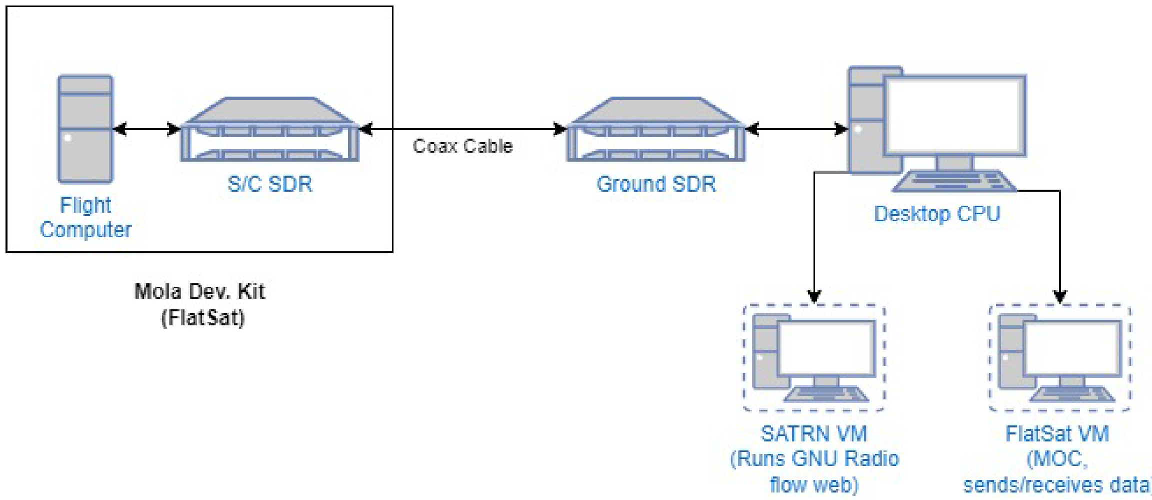


Figure 46. Wired FlatSat communications test setup

Because the over-air test involved radiating from the dish, the educational ground station configuration was slightly different than what has been presented for the previous tests. Figure 47 presents the configuration layout of the educational station for the over-air test. The ground SDR that was used in the wired communications test was injected into the operational station through an open port in the RF switch, which serves as the RF transmit route to the dish. The active port of the RF switch was changed from port 1 to port 3, where the test SDR was connected. The RF receive route was connected from the SDR to the splitter, as shown in Figure 47. The power supply for the SDR was borrowed from the S-band SDR of the operational station. This, along with the RF switch port configuration, was documented as a requirement to return to operational configuration.

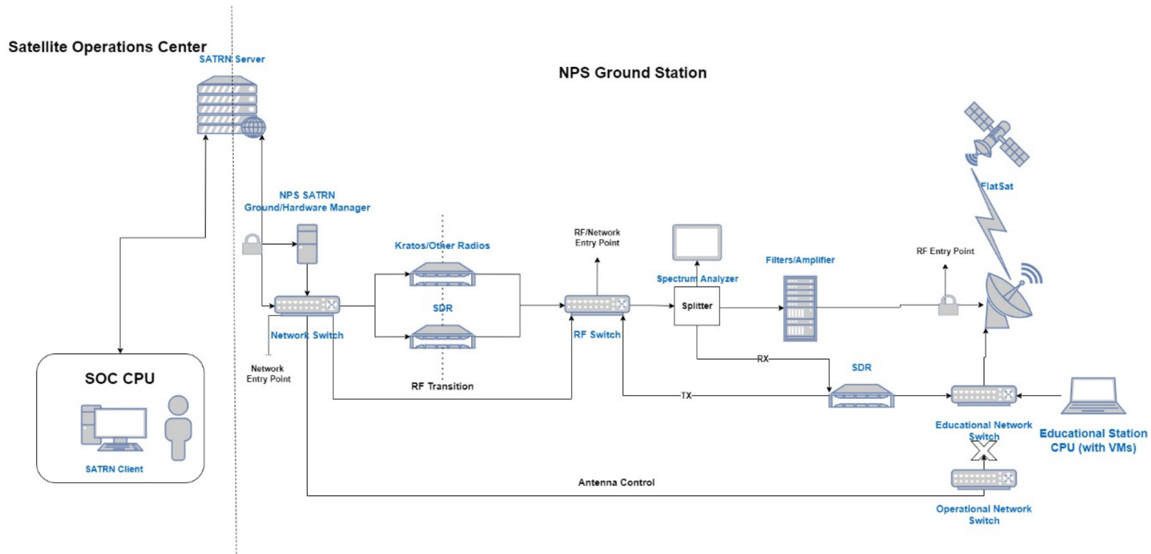


Figure 47. Educational station configuration and injection for over-air test

The SDR was then connected to the educational network switch. The dish was disconnected from the operational network switch and connected to the educational switch. This was documented as a requirement to return to the operational configuration. The educational switch was connected to the educational station CPU through the legacy MC3 port of another network switch, which connected the desktop computer from the wired communication test to the SDR. The legacy MC3 port and computer is completely separated from the operational network, and served as the educational station CPU since the laptop did not have the computing power to perform the test. As demonstrated from the wired communication test, the desktop runs the required VMs. The spectrum analyzer was connected to the splitter and tuned to the Mola downlink frequency of 2.25 GHz to visualize the received RF data. The Mola development kit (FlatSat in Figure 47), was connected to bench power and turned on. Transmit and receive dipole antennas were connected to Mola's radio to enable over-air communications.

An important aspect of successfully injecting hardware into the ground station was properly configuring the IP addresses of the injected hardware to comply with the network addresses of the shared hardware on the operational station (the dish antenna). By properly configuring the IP addresses of the injected hardware, local communication between the injected hardware and shared operational hardware was ensured. The injected hardware IP addresses were reconfigured, vice the operational hardware, in order to minimize the configuration changes of the operational station. For example, the dish at NPS has an IP address of 192.168.XXX.YYY. The first two octets of the IP address represent the MC3 network. The third octet, XXX, represents the network that is unique to the local NPS ground station (each ground station on MC3 has the same first two octets, and a different third). The last octet, YYY, is unique to the particular piece of hardware on the local network. Therefore, to properly inject the hardware for the over-air test, the IP addresses of the educational CPU (including the IP addresses of both VMs) and the SDR were changed to the form 192.168.XXX.YYY with unique final octets for each.

The SATRN VM was used on the educational station CPU to conduct a test pass between the NPS ground station and the FlatSat through SATRN Client. The SDR and NPS dish were configured as the radio-rotator pair for the FlatSat with proper frequencies via SATRN Hardware Manager. On the FlatSat VM, a terminal was open to send commands from the simulated MOC to the FlatSat. Ping commands were used because they were simple and sufficient for the sake of the over-air test.

The SATRN test pass began and showed consistent uplink from the ground station once the ping commands were sent from the MOC. This indicated that the dish was successfully radiating. However, it was observed that no packets were being received from the FlatSat. It was determined that the lack of downlink was because the dish was not pointed at the general relative location of the FlatSat. The dish was moved from its nominal stowage position to point in the direction of the FlatSat through the SATRN VM and SATRN Hardware Manager. This was an important result, as it demonstrated that not only can the educational station transmit and receive through SATRN, but it can also drive the dish through SATRN, fulfilling all the SATRN capabilities that are presented through the operational station. Once the dish was properly pointed, there was successful downlink from the FlatSat, as indicated by the spectrum analyzer and SATRN session shown in Figures 48 and 49.

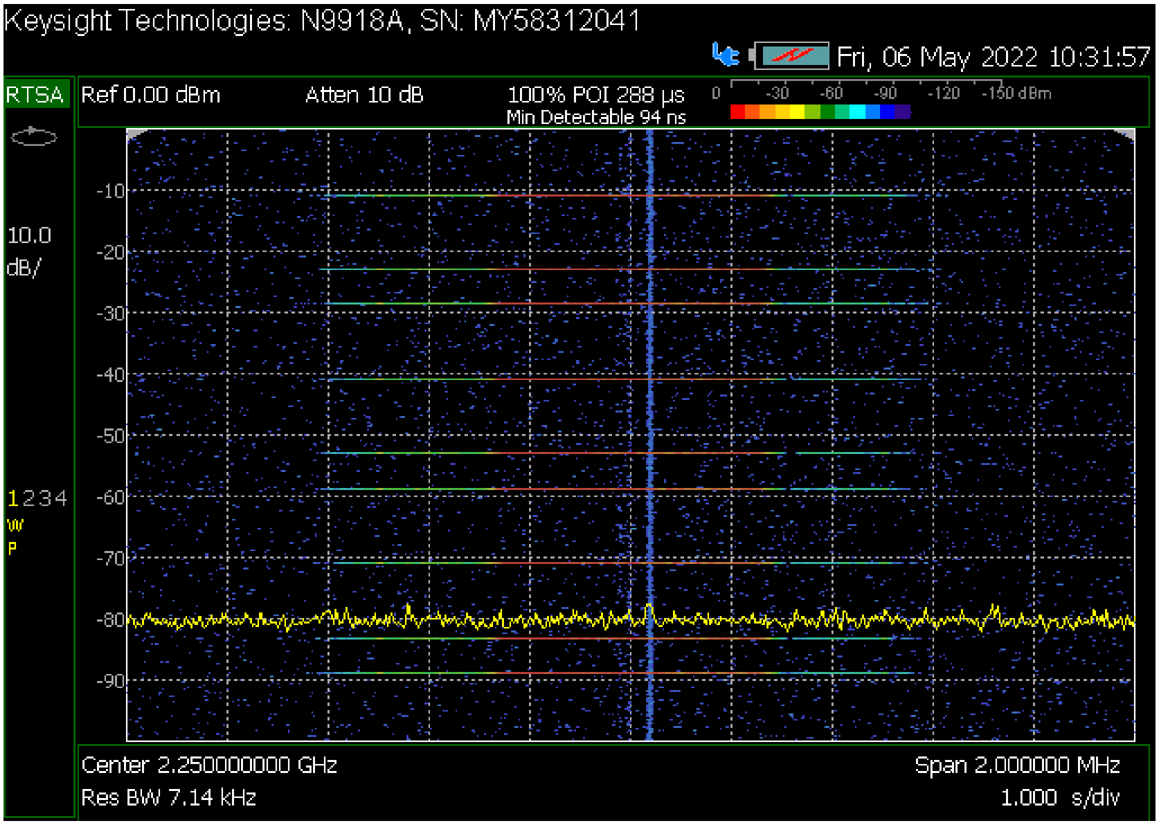


Figure 48. Spectrum analyzer screenshot showing reception of downlinked ping commands from the FlatSat

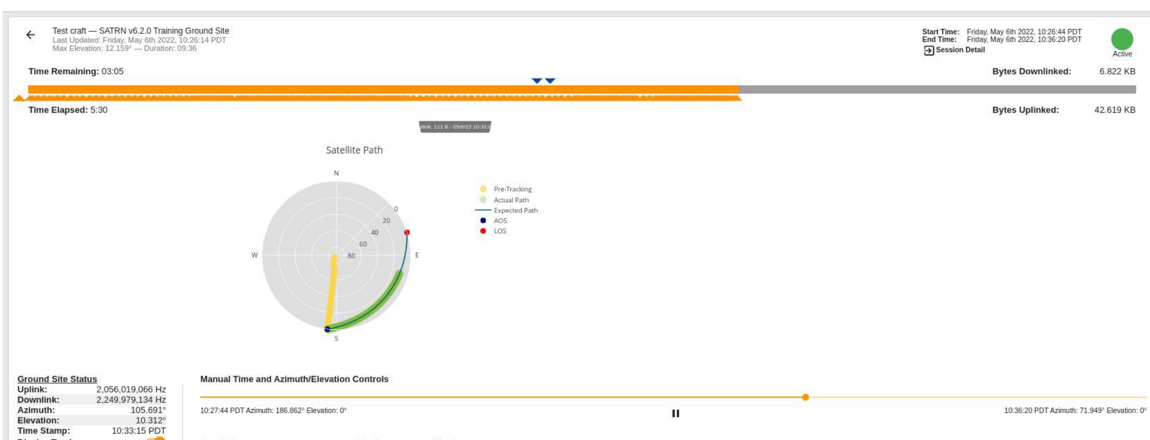


Figure 49. SATRN Client screenshot of the FlatSat test pass

The spectrum analyzer presented in Figure 48 shows a series of parallel lines that indicate the brief RF pulses of the ping commands being received from the FlatSat to the dish over a period of time. Once the dish was pointed approximately in the direction of the FlatSat, the pulses on the spectrum analyzer appeared consistently for the remainder of the test. From the top right corner of the SATRN session in Figure 49, it is demonstrated that the educational station was both transmitting and receiving data during the test pass, with about 42,600 kB of data uplinked and 6,800 kB of data downlinked. The discrepancy in the amount of data uplinked and downlinked was due to the dish not being pointed toward the FlatSat at the beginning of the pass. Once correctly oriented, SATRN demonstrated consistent data rates between uplink and downlink, as was expected for sending simple ping commands.

The over-air test was successful in closing the communications link between the ground station and Mola's communication hardware, validating the GNU Radio flowgraph and communications capability of the satellite. The operational station was reconfigured by returning the S-band SDR power supply, switching the RF switch back to port 1, and reconnecting the dish to the operational network switch. No other configuration changes were made, which demonstrated a simple injection procedure for over-air testing and radiating from the dish from the educational station.

Another trial of the over-air test was completed with a higher-powered laptop—compared to the laptop used in the wired communication test—acting as the educational station CPU to investigate the minimal computing power required to perform over-air communications through the educational station. This is useful for potential educational station users because a laptop provides more flexible injection, given its portable nature, than a dedicated desktop. From the procedural description of the over-air test, the CPU must have the computing power to run two VMs with each running either GNU radio or SATRN.

The over-air test with the laptop as the CPU was successful and produced results similar to the test with the desktop computer. For reference, the laptop's computing specifications are provided in Table 4. These specifications should be considered the minimum required in order to successfully perform over-air communications.

Table 4. Computing specifications of the laptop used in over-air testing

Specification	Value
RAM	16 GB
Processor Frequency	1.9 GHz
Number of Core Processors	4

Future tests are planned, including over-air tests with the complete Mola satellite rather than the development kit. This test will include testing the transmission of real TT&C commands between the ground station and satellite instead of simple ping commands to further validate the integration of Mola with the ground segment

The over-air test of the Mola communications hardware achieved multiple objectives. First, the test validated the injection procedure and configuration of the educational station to transmit and receive RF signals from the dish, as well as run SATRN with its full functionality through a VM. This is arguably the most important result of the test, as it demonstrated that the educational station can be run independently with the same capabilities of the operational station through SATRN, opening a multitude of potential research, instructional, and educational experiments to the host institution. Secondly, the test provided successful preliminary communications results for the development and integration of Mola onto the MC3 network. Lastly, the test demonstrated the operational benefits of the parallel station by having the flexibility to perform integration testing without being inhibited by the approval process that afflicts the operational network.

V. CONCLUSION

The research effort behind this thesis focused on opening opportunities for institutions that host the MC3 satellite ground network to provide more flexible and autonomous use of their valuable hardware beyond nominal MC3 operations. This thesis outlined a general procedure for injecting a parallel ground station that leverages the aforementioned valuable hardware and infrastructure without being restricted by the lengthy approval processes and stringent requirements that accompany an operational government owned and operated ground network. In addition to the general injection procedure, detailed examples of potential use cases of the parallel station were presented, focusing on the educational, R&D, and operational benefits that the parallel station provides.

A. SUMMARY OF RESULTS

Chapter II offered a broad overview of current satellite ground networks, ranging across government, commercial, and educational entities. It was demonstrated that the MC3 parallel station fills a particular niche within the current state of practice of satellite ground stations as a means for the government to freely research and develop new space communications innovations, particularly in the booming small satellite category. Furthermore, the parallel station provides MC3 host institutions the unique ability to provide hands-on training relevant to running a ground station as well as easy access to space systems education.

Chapter III provided an extensive overview of the operational MC3 ground network. From the description of the general network, the infrastructure of the NPS ground station was investigated to inform the design and integration of the parallel station. After consideration of various injection points, it was determined that the network switch along the antenna control pathway of the station provided an ideal balance of valuable shared hardware (the parabolic dish antenna) and ease and feasibility of integration. A general CONOP for proper injection was then presented in Table 2. This CONOP should be used as a reference for MC3 operators interested in utilizing their station autonomously from the operational network. The most crucial aspect of this procedure is ensuring a proper return to the operational configuration as to not disturb nominal MC3 operations.

The majority of this research effort was presented in Chapter IV. This chapter described the various testing and experimentation that the parallel station underwent to verify and validate its proposed functions. Firstly, the brute force injection demonstrated the validity of the design and CONOP presented in Chapter III. The simple parallel station, consisting of a network switch and a laptop computer, was successfully injected and able to control the parabolic dish antenna at the NPS ground station. Most importantly, the operational station was successfully reconfigured, demonstrating proof-of-concept the dual-use ground station.

Section B of Chapter IV demonstrated an example of how the parallel station can be used to develop and test new concepts regarding operation of the ground station. A new tracking algorithm was created that utilized velocity commanding of the dish rather than positional commanding as is current practice for the operational station. The algorithm was first tested by commanding the dish to track a simulated satellite. Once the results were considered satisfactory, the algorithm was modified to track and receive RF signals from operational satellites in orbit. The tracking algorithm allowed for smoother dish movements as well as error correction to adjust for deviations from the satellite's relative overhead trajectory. The algorithm resulted in average pointing errors on the order of hundredths of a degree. It is expected that with further development, the velocity control error correction algorithm can achieve even smaller pointing errors. There is interest from MC3 operators to further develop the algorithm to be implemented as the control algorithm on the operational network. This test also verified the parallel station's ability to receive RF signals.

The final test of the parallel station, presented in Section C of Chapter IV, was an over air test for the Mola spacecraft currently in development at NPS. The over air test demonstrated the benefit that the parallel station provides to integration testing at MC3 stations. The parallel station was injected and configured such that it could both transmit and receive from the dish antenna, verifying its full functionality as a satellite ground station. The parallel station successfully communicated over air with Mola, verifying the communication compatibility between the spacecraft and the ground segment. This provided the Mola team a real and relevant checkpoint for the development of their spacecraft. Furthermore, this test uncovered the ability of the parallel ground station to fully utilize the SATRN software locally at the ground station. This importance of this result cannot be understated, as it demonstrated that the parallel station can access the full functionality of the operational network autonomously. This opens many opportunities for MC3 operators, as will be further explained in Section C of this chapter.

B. FUTURE WORK

From the successful demonstrations of the potential use cases of the parallel station, it would be worth looking into a more seamless transition between the operational and educational stations. That is, determining a method and configuration at the ground station that allows for simple switching between the stations rather than physically unplugging and plugging various hardware. This could be envisioned as the educational station and associated hardware being permanently connected to a network switch that is also connected to the operational station and its associated hardware. An operator could then digitally toggle between the two stations. This approach would require additional research in order to ensure that the stations are separated at a point that does not allow the educational station to access the greater operational network due to cybersecurity concerns.

In regards to the research surrounding the velocity control error correction tracking algorithm, more testing and development is required before it can be considered as a replacement for the current tracking algorithm. Specifically, it needs to be made more resilient. As mentioned in Chapter IV Section B, the use of a buffer array needs to be implemented in order to mitigate the communication error experienced in MATLAB that disallowed the reception of the dish status from the ACU. Furthermore, the algorithm should be modified to eliminate the keyhole effect of near-direct overhead satellite contacts in order to maintain low pointing errors. The stability concerns of the tracking algorithm should be further investigated to potentially achieve lower pointing errors by increasing the error correction frequency.

Separate but related to this research effort, the relevancy of optimal control theory in the control of the NPS dish was conducted as a part of the 299 NPS certificate program. This effort focused on the feasibility of integrating optimal control theory to the control of the NPS dish, particularly how to optimally slew the dish to intercept a spacecraft's trajectory. Figure 50 shows a simulated optimal slew of the dish to intercept a simulated spacecraft, created by the optimal control solver DIDO [22]. In this scenario, the dish was positioned in its nominal stowage configuration. DIDO generated the optimal control to slew the dish to intercept the spacecraft in the minimal amount of time, given the physical constraints on the dish's acceleration and velocity.

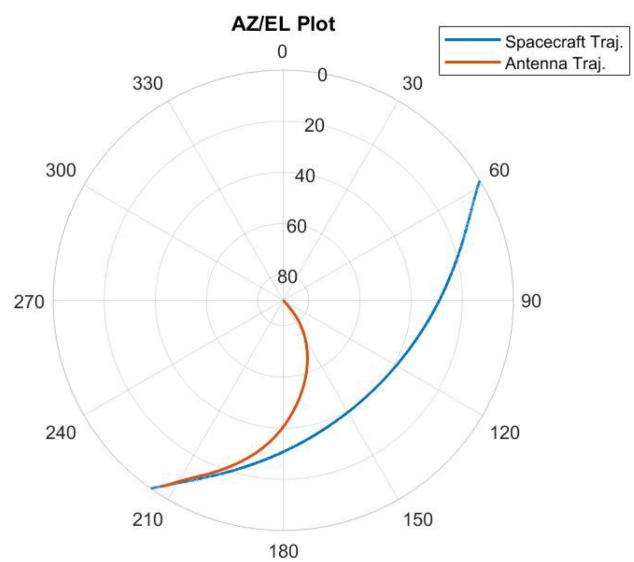


Figure 50. Interception of a simulated spacecraft trajectory using trajectory optimization

It would be beneficial to test the feasibility of the optimal control using the physical dish on the educational station. Generally, there is potentially beneficial work in using optimal control theory to transition the dish control from classical control theory to state-space control theory. Making this transition would greatly modernize the MC3 network, allowing for more efficient and precise tracking of satellites. This is particularly worthwhile as the number of satellites on the MC3 network is expected to quickly grow, causing a ground station to have multiple satellite contacts at the same time.

C. APPLICATION CASE STUDY: USCGA

This section serves to demonstrate the impact of this research beyond NPS and how it can be applied to other MC3 installations with different interests and goals. The Coast Guard Academy is looking to expand their undergraduate space systems education given the relevancy of space operations to the mission of the Coast Guard. Throughout this research effort, USCGA cadets and faculty have shown great interest in the possibility of leveraging their MC3 assets locally to support undergraduate education and research. This includes small satellite design and integration, communication payload testing via high-altitude balloons and drones, and general education in space systems and ground station operation.

A major interest for the use of the parallel ground station is receiving scientific data directly from active satellites. From conversations with Coast Guard cadets, it became clear that finding published, readily available, and relevant spaceborne data was difficult. This presented a barrier for students to conduct research for their capstone projects as well as assigned labs for classes. The parallel ground station removes this barrier, as cadets can receive relevant data directly from the operational satellites while also gaining valuable knowledge on nominal ground station operations. The following hypothetical scenario will be used to demonstrate how the Coast Guard Academy can use the injection procedure presented in Chapter III and tested in Chapter IV to accomplish their unique academic goals.

A cadet is interested in researching the effects of changing ocean patterns on Coast Guard search and rescue operations. Through their research, the cadet has found a NOAA satellite that provides relevant scientific data that allows them to track these ocean patterns over an extended period of time. Using the over-air test presented in Chapter IV, Section C, as a reference, the cadet works with the MC3 faculty at USCGA to devise an experimental setup to use the UHF antenna at the USCGA ground station to receive the data directly from the NOAA satellite. In addition to the UHF antenna, the cadet plans on using an SDR to process the received RF data into useable information. The ground station will be configured similar to what is presented in Figure 47 in Chapter IV, using the SATRN VM to track the satellite and receive downlinked data through the SDR. Since there is no need for uplink, the SDR will solely be connected to the receive port at the splitter.

From the above experimental setup, the cadet performs a test readiness review with the MC3 faculty at USCGA, beginning the injection process presented in Table 2 in Chapter III. Once the experimental design is considered satisfactory, the USCGA MC3 personnel notify the NPS MC3 personnel on their experiment and potential schedule to deconflict any scheduled operational contacts using the USCGA ground station. Once the experiment is deconflicted with the operational schedule, station downtime is scheduled at USCGA through their SATRN Hardware Manager. The educational station is injected per the experimental setup, ensuring that any configuration changes are well documented. When the experiment is executed and scientific data is received, the station is returned to its operational configuration, and the station downtime is ended via SATRN Hardware Manager. The USCGA team conducts a simulated test contact through SATRN Client to ensure that the operational station is back online. To complete the procedure, USCGA personnel notify NPS personnel of their status and that the node is online for the operational network.

Because the presented scenario is hypothetical, it does not go into the high-level details of what is required for its execution. However, it provides a guide to interested MC3 installations on how the educational station CONOP is used in practice outside of NPS. As the originators of this research, NPS MC3 personnel are available for training, troubleshooting, and questions regarding the use of the parallel educational ground station.

APPENDIX. VELOCITY CONTROL ERROR CORRECTION MATLAB CODE

```
clear
clc
close all

[data, txt]=xlsread('Aim Test.xlsx');
AZ=data(:,1);

% %correct AZ for gearbox lockout at 0-360 deg transition
% AZ=AZ-180;
% for m=1:numel(AZ)
%     if AZ(m)<352
%         AZ(m)=AZ(m)+360;
%     end
% end

%Correct trajectory of dish azimuth limits at 354 degrees
for m=1:numel(AZ)
   if AZ(m)>354
       AZ(m)=354;
   end
end

EL=data(:, 2);
% EL=180-EL;

time=datetime((txt(2:end,1)), 'InputFormat', 'yyyy:MM:dd:HH:mm:ss.SSS',
'Format', 'dd-MMM-uuuu HH:mm:ss.SSS');
time = time - hours(7); %Set time to local time

%generate velocity commands
dt=.2;
AV=diff(AZ)./dt;
EV=diff(EL)./dt;
AV=[AV; 0];
EV=[EV; 0];

% %For MATLAB testing of code, comment out for active testing
% diff_time=time(1)-datetime('now')-seconds(5)
% time=time-diff_time;
% %------------


%TCP/IP communication
dish=tcpclient("192.168.151.98," 5003)
pause(1)

%Preposition Dish
AZ_0=num2str(AZ(1));
```

```
EL_0=num2str(EL(1));
writeline(dish, [‘am’, AZ_0, ‘; ‘, ‘em’, EL_0, ‘;’])

i=1;

waittime=time(1)-datetime(‘now’) %time until pass begins
pause(seconds(waittime))

while datetime(‘now’)<=time(end) %repeat loop until contact ends
   k=find(time>=datetime(‘now’,’Format’, ‘dd-MMM-uuuu HH:mm:ss.SSS’),1);

   writeline(dish, [‘av’, num2str(AV(i)),’; ‘, ‘ev’, num2str(EV(i)), ‘; ‘,
‘SQ;’]) %Send velocity commands and request status to dish from telnet

   message=char(readline(dish));
   while numel(message)<23 %ensure read of dish status
       message=char(readline(dish));
   end
   SQ1=message; %read current position from dish

   t_DISH(i)=time(k);
   AZ_DISH(i)= str2num(SQ1(7:13)) %current dish position
   EL_DISH(i)= str2num(SQ1(17:23))

   %calculate position error
   AZ_error(i)=AZ(k)-AZ_DISH(i);
   EL_error(i)=EL(k)-EL_DISH(i);

   %generate new velocity cmd (every c time steps)
   c=5;
   if mod(i, c)==0 %every c steps
       if k<(length(AV)-c)
           AV(i+1:i+c)=(AZ(k+c)-AZ_DISH(i))/(c*dt); %error correction
           if abs(AV(i+1))>3 %impose dish velocity limits
               AV(i+1:i+c)=sign(AV(i+1))*3;
           end

           EV(i+1:i+c)=(EL(k+c)-EL_DISH(i))/(c*dt); %error correction
           if abs(EV(i+1))>3 %impose dish velocity limits
               EV(i+1:i+c)=sign(EV(i+1))*3;
           end

       end
   else
       AV(i+1)=AV(i);
       EV(i+1)=EV(i);
   end

i=i+1;
   pause(dt)

end

%Stop dish
```

```
writeline(dish, ['av0; ev0;'])

%% Postprocessing
dt1=milliseconds(diff(t_DISH));
for i1=2:(length(dt1)+1)
   t1(1)=0;
   t1(i1)=t1(i1-1)+dt1(i1-1);
end
t1=t1'/1000;

dt2=milliseconds(diff(time));
for i2=2:(length(dt2)+1)
   t2(1)=0;
   t2(i2)=t2(i2-1)+dt2(i2-1);
end
t2=t2'/1000;

figure;
subplot(2,1,1)
plot(t2, AZ, t1, AZ_DISH)
ylabel('Azimuth (deg)')
legend('S/C Pos.', 'Dish Pos.')
subplot(2,1,2)
plot(t2, EL,t1, EL_DISH)
ylabel('Elevation (deg)')
legend('S/C Pos.', 'Dish Pos.')
xlabel('Time (s)')

figure;
polarplot(AZ./180*pi, EL, AZ_DISH./180*pi, EL_DISH)
legend('S/C Pos.', 'Dish Pos.')
pax=gca;
pax.ThetaZeroLocation='top';
set(gca, 'Rdir', 'reverse')
pax.RLim=[0 90];
pax.ThetaDir='clockwise';

figure;
plot(t1, [AZ_error; EL_error])
xlabel('Time (s)')
ylabel('error')
legend('Az. Error','El. Error')



function connectionFcn(src,~)
if src.Connected
  disp("This message is sent by the server after accepting the client
connection request.")
else
  disp("Client has disconnected.")
end
end
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] N. Saeed, A. Elzanaty, H. Almorad, H. Dahrouj, T. Y. Al-Naffouri, and M.-S. Alouini, "CubeSat communications: Recent advances and future challenges," *IEEE Communications Surveys Tutorials*, vol. 22, no. 3, pp. 1839–1862, 2020, doi: 10.1109/COMST.2020.2990499.

[2] M. Bazdresch, S. Velayudhan, and W. Johnson, "A satellite ground station for teaching digital and wireless communications," in *2016 IEEE Frontiers in Education Conference (FIE)*, Oct. 2016. doi: 10.1109/FIE.2016.7757672.

[3] "Global Educational Network for Satellite Operations." Accessed Sep. 16, 2021 [Online]. Available: https://www.esa.int/Education/Global_Educational_Network_for_Satellite_Operations

[4] G. Minelli, L. Magallanes, N. Weitz, D. Rigmaiden, J. Horning, J. Newman, D. R. Bu, M. Scott, S. Brady, C. Watkins, J. Christensen, C. Buttars, R. Beus, R. Oakden, "The Mobile CubeSat Command and Control (MC3) Ground Station Network: An Overview and Look Ahead," presented at the 33rd Annual AIAA/USU Conference on Small Satellites, Logan, UT, Aug. 2019.

[5] R. C. Griffith, "Mobile Cubesat Command and Control (MC3)," M.S. thesis, Space Systems Academic Group, NPS, Monterey, CA, USA, 2011. Accessed: Sep. 16, 2021. [Online]. Available: https://calhoun.nps.edu/handle/10945/5591

[6] G. Minelli, L. Magallanes, N. Weitz, D. Rigmaiden, J. Horning, J. Newman, R. Phelps, "The Mobile CubeSat Command and Control (MC3) Ground Station Network," presented at the Small Spacecraft Virtual Institute, May 20, 2020.

[7] H. Monaghan, "About the Deep Space Network," NASA, Mar. 30, 2020 [Online]. Available: http://www.nasa.gov/directorates/heo/scan/services/networks/deep_space_network/about

[8] J. Berner and T. Pham, "Deep Space Network Services Catalog." California Institute of Technology, Sep. 06, 2020 [Online]. Available: https://deepspace.jpl.nasa.gov/files/820-100-G1.pdf

[9] J. Foust, "Increasing demands putting pressure on Deep Space Network," SpaceNews, Jul. 10, 2021 [Online]. Available: https://spacenews.com/increasing-demands-putting-pressure-on-deep-space-network/

[10] D. Baird, "NASA to commercialize near-Earth communications services," NASA, Oct. 26, 2020 [Online]. Available: http://www.nasa.gov/feature/Goddard/2020/nasa-to-commercialize-near-earth-communications-services

[11] NASA, “Services and Scheduling.” Accessed Dec. 01, 2021 [Online]. Available: https://www.nasa.gov/directorates/heo/scan/services/services/index.html

[12] Y. Wong, S. Schaire, C. Patel, L. Ambrose, O. Kegege, and D. Thorsen, “NASA leveraging commercial communication ground stations for small satellites,” presented at the 33rd Annual AIAA/USU Conference on Small Satellites, Logan, UT, Aug. 2019.

[13] Kongsberg, “KONGSBERG.” Accessed Jan. 04, 2022 [Online]. Available: https://www.kongsberg.com/

[14] “KSAT adds South Korean station to its already extensive ground network,” KSAT: Kongsberg Satellite Services, Sep. 04, 2020 [Online]. Available: https://www.ksat.no/news/news-archive/2020/ksat-is-expanding-their-global-ground-network/

[15] “KSATlite,” KSAT: Kongsberg Satellite Services. Accessed Oct. 21, 2021 [Online]. Available: https://www.ksat.no/ground-network-services/ksatlite/

[16] D. White, C. Shields, P. Papadeas, A. Zisimatos, M Surligas, M. Papamatthaiou, D. Papadeas, E. Kosmas, “Overview of the Satellite Networked Open Ground Stations (SatNOGS) Project,” presented at the 32nd Annual AIAA/USU Conference on Small Satellites, Utah, USA, Aug. 2018.

[17] “SatNOGS Network,” SatNOGS Wiki, Feb. 21, 2020 [Online]. Available: https://wiki.satnogs.org/SatNOGS_Network

[18] “Morehead State University :: Laboratories and Facilities- Space Science Center.” Accessed Feb. 16, 2022 [Online]. Available: https://www.moreheadstate.edu/college-of-science/earth-and-space-sciences/space-science-center/laboratories-facilities/21m-space-tracking-antenna

[19] B. Malphrus, J. Kruth, M. Combs, N. Fite, B. Twiggs, and R. Schulze, “A university-based ground station: The 21 M antenna at Morehead State University,” presented at the SpaceOps 2010 Conference: Delivering on the Dream (Hosted by NASA Marshall Space Flight Center and Organized by AIAA), Huntsville, Alabama, Apr. 2010. doi: 10.2514/6.2010-2156.

[20] J. H. Sevillaparra, “Integration of Mobile Cubesat Command and Control (MC3) operationas with Amazon Web Services,” M.S. thesis, Space Systems Academic Group, NPS, Monterey, CA, USA, 2020.

[21] H. Kramer, “Coriolis/WindSat,” eoPortal. Accessed Apr. 11, 2022 [Online]. Available: https://directory.eoportal.org/web/eoportal/satellitemissions/content//article/coriolis

[22] I. M. Ross, “Enhancements to the DIDO Optimal Control Toolbox.” arXiv, Jun. 14, 2020 [Online]. Available: http://arxiv.org/abs/2004.13112

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California