

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

DISSERTATION

RESOURCE-CONSTRAINED AUTONOMOUS OPERATIONS OF SATELLITE CONSTELLATIONS AND GROUND STATION NETWORKS

by

Giovanni Minelli

September 2018

Dissertation Supervisor: Isaac M. Ross

Approved for public release. Distribution is unlimited.

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** September 2018	**3. REPORT TYPE AND DATES COVERED** Dissertation

4. TITLE AND SUBTITLE RESOURCE-CONSTRAINED AUTONOMOUS OPERATIONS OF SATELLITE CONSTELLATIONS AND GROUND STATION NETWORKS	**5. FUNDING NUMBERS**
6. AUTHOR(S) Giovanni Minelli	
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)

To address the growing population of small satellite constellations that rely on distributed ground station networks, a dynamic optimization problem is formulated and solved. Specifically, this dissertation addresses the problem formulation, algorithm implementation, and experimental techniques developed to optimally slew ground-based antennas between multiple satellites that are simultaneously in view of one or more earth stations. The problem is solved using DIDO, a MATLAB optimal control solver, to produce deconflicted ground antenna slew trajectories. The deconfliction parameters include space-to-ground link budgets, mission priority, asset availability, and onboard health. Traditional methods employ heuristics to generate a subset of available targets and a separate process to check feasibility of the solution. The method described in this dissertation deterministically solves the problem in a single step. The approach is experimentally validated and tested using a small constellation of low-Earth-orbiting CubeSats operated by the Small Satellite Laboratory at the Naval Postgraduate School, using the Mobile CubeSat Command and Control (MC3) ground station network.

14. SUBJECT TERMS optimization, CubeSat, mission operations, MC3, Gaussian targeting, benefit value function, dynamic optimization, DIDO	**15. NUMBER OF PAGES** 227
	16. PRICE CODE

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

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

RESOURCE-CONSTRAINED AUTONOMOUS OPERATIONS OF SATELLITE CONSTELLATIONS AND GROUND STATION NETWORKS

Giovanni Minelli
Civilian, Department of the Navy
BS, Santa Clara University, 2006
MS, Santa Clara University, 2011

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN ASTRONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2018

Approved by:

Isaac M. Ross
Department of Mechanical and Aerospace Engineering
Dissertation Supervisor

Francis X. Giraldo
Department of Applied Mathematics

Mark Karpenko
Department of Mechanical and Aerospace Engineering

James H. Newman
Space Systems Academic Group

Marcello Romano
Department of Mechanical and Aerospace Engineering

Approved by:

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

Douglas Moses, Vice Provost of Academic Affairs

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

To address the growing population of small satellite constellations that rely on distributed ground station networks, a dynamic optimization problem is formulated and solved. Specifically, this dissertation addresses the problem formulation, algorithm implementation, and experimental techniques developed to optimally slew ground-based antennas between multiple satellites that are simultaneously in view of one or more earth stations. The problem is solved using DIDO, a MATLAB optimal control solver, to produce deconflicted ground antenna slew trajectories. The deconfliction parameters include space-to-ground link budgets, mission priority, asset availability, and onboard health. Traditional methods employ heuristics to generate a subset of available targets and a separate process to check feasibility of the solution. The method described in this dissertation deterministically solves the problem in a single step. The approach is experimentally validated and tested using a small constellation of low-Earth-orbiting CubeSats operated by the Small Satellite Laboratory at the Naval Postgraduate School, using the Mobile CubeSat Command and Control (MC3) ground station network.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION....1
A. SMALL SATELLITES AND CUBESATS....1
B. LARGE CONSTELLATIONS....4
C. COMMUNICATIONS ARCHITECTURES....7
D. SIMPLIFIED PROBLEM DESCRIPTION AND HEURISTIC METHODS....12
1. One Satellite, One Ground Station....13
2. Two Satellites, One Ground Station....15
3. Three Satellites, One Ground Station....18
4. Two Satellites, Two Ground Stations....20
5. Three Satellites, Two Ground Stations....23
E. SCHEDULING CHALLENGES....25
F. DISSERTATION OBJECTIVES AND SCOPE....28
G. DISSERTATION OUTLINE....30

II. GROUND-BASED SATELLITE TRACKING....31
A. LITERATURE REVIEW....31
1. Graph-Theoretic Problem Formulation....34
2. Optimal Control Fundamentals....36
3. Smooth Target Model Development....39
B. MATHEMATICAL DEFINITIONS....42
1. Classical Orbital Elements....42
2. Satellite Observations from Earth's Surface....47
3. Two Line Elements....51
4. Target Representation as Polar Plot in Cartesian Space....52
5. Link Strength Analysis....54
6. Kinematics and Dynamics....58
7. Target Model Development....61
C. CHAPTER SUMMARY....64

III. PROBLEM FORMULATION AND VALIDATION....65
A. INITIAL PROBLEM FORMULATION....65
1. Benefit Value Function....65
2. Problem Formulation....66
B. PROBLEM FORMULATION WITH TORQUE SMOOTHING....75
C. ONE SATELLITE, ONE GROUND STATION....78

1. Analytic vs. Numerical Orbit Propagation 82
2. Numerical Propagator Double Integrator with Torque Smoothing 85
D. CHAPTER SUMMARY 88

IV. MULTI-SATELLITE SCENARIOS 89
A. TWO SATELLITES, ONE GROUND STATION 89
1. Double Integrator 92
2. Double Integrator with Torque Smoothing 95
3. Real-World Application 96
4. Numerical Integration V&V 99
B. THREE SATELLITES, ONE GROUND STATION 100
C. FOUR SATELLITES, ONE GROUND STATION 106
D. TWO SATELLITES, TWO GROUND STATIONS 108
E. THREE SATELLITES, TWO GROUND STATIONS 112
F. TEN SATELLITES, ONE GROUND STATION 118
G. CHAPTER SUMMARY 122

V. EXPANDED EXPERIMENTATION 123
A. MC3 OPERATIONAL TESTING ENVIRONMENT 126
1. At the NPS SOC 126
2. At the Remote MC3 Node 129
3. Pass Planning and Deconfliction 133
B. REAL-WORLD SCENARIOS 137
1. Two Satellites, One Ground Station 138
2. Two Satellites, Two Ground Stations 140
3. Crossing Orbits 145
4. Three Satellites, Two Antennas 149

VI. CONCLUSION 151
A. FUTURE WORK 152
1. Automatic Scaling and Balancing 152
2. Binary Variables and Extended Testing with Narrow-Beam Dishes 152
3. Mapping of Mission Parameters into Target Gaussian Modeling 155
B. PROPOSED CONOPS FOR FUTURE APPLICATIONS 158
1. Satellite Tracking Problem 158
2. Satellite Imaging Problem 160

3. Autonomous Vehicle Guidance.....160
C. FINAL SUMMATION 161

APPENDIX A. 163

APPENDIX B. 179

APPENDIX C. 187

LIST OF REFERENCES 193

INITIAL DISTRIBUTION LIST 201

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 1. CubeSats in 1U, 3U, and 6U sizes. Source: [11]. 3

Figure 2. The Iridium constellation. 4

Figure 3. Two Planet *Dove* satellites deployed from the International Space Station. 5

Figure 4. SN and NEN proposed architecture to support CubeSats and CubeSat constellations. Source: [28]. 9

Figure 5. KSAT commercial ground station locations. Source: [30]. 10

Figure 6. MC3 network ground station locations. 12

Figure 7. Simplified one-satellite one-ground station contact scenario. 14

Figure 8. Schematic of GS1 schedule allocation and signal strength/ benefit curve. 14

Figure 9. Idealized two-satellite one-ground station case. 16

Figure 10. GS1 allocation plot for exclusive, greedy, equal, and benefit value schemes. 16

Figure 11. Three satellite contact scenario for one ground station. 18

Figure 12. Allocation curves for three satellite, one ground station example. 19

Figure 13. Two satellite, two ground station case. 20

Figure 14. Resource allocation curves for two satellite/two ground station case. 22

Figure 15. Three satellite/two ground station scenario. 23

Figure 16. Resource allocation curves for three satellites, two ground stations. 24

Figure 17. Simplified network diagram for SATRN connections on MC3. ...26

Figure 18. Example of a directed graph. 35

Figure 19. Unit p-norms for p = 1, 2, and 100—shapes are continuous and smooth. Source: [80]. 40

Figure 20. Example of Gaussian function centered at the origin. Source: [49]. 41

Figure 21. Circular orbit in two dimensions 44

Figure 22. Elliptical orbit in two dimensions. 44

Figure 23. Three-dimensional representation of Earth orbit. Source: [83]. 45

Figure 24. Topocentric-horizon coordinate system. 47

Figure 25. ECEF to ENU transformation. 48

Figure 26. Cartesian space representations of traditional Unit Circle (left) and Compass circle with Azimuth angles (right) 53

Figure 27. Elevation angles with respect to the horizon (left) and represented in Cartesian space (right) 54

Figure 28. Typical polar diagram for a Yagi antenna. Source: [93]. 58

Figure 29. Three different satellite communications apertures. 59

Figure 30. Satellite flyby of the NPS ground station at three time points: acquisition of signal (top left), point of closest approach (top right), and loss of signal (bottom) 61

Figure 31. Idealized link margin calculation for the CubeSat as it passes the NPS ground station (assumes no spacecraft rotation/antenna pointing losses, and no environmental noise). 62

Figure 32. Azimuth/elevation polar plot representation of ground station and satellite Gaussian 63

Figure 33. One satellite, one ground station simulation. 79

Figure 34. Benefit plots (top) and slant range distance to ground station (bottom). 80

Figure 35. Animated results from simulation, with progression from left to right. 81

Figure 36. Optimal control solutions from analytic (AN) and numerical (NU) trials for one satellite and GS1 station with identical orbits and start/stop times. 83

Figure 37. Torque control V&V plots for analytic and numerical double integrator models. 85

Figure 38. Outputs from double integrator with torque smoothing. 86

Figure 39. Torque control KKT verification. 87

Figure 40. Velocity state KKT verification for double integrator with smoothing....88

Figure 41. Two satellite, one ground station scenario. Top-down ground track view (left) and reference from Chapter I (right). Satellites move from left to right. 89

Figure 42. Two-satellite animation progression starting at top left. Satellite1 link margin in blue (dB), Satellite2 link margin in green (dB), combined benefit in cyan (CU). 90

Figure 43. Benefit value curves for two satellite/one ground station scenario. 91

Figure 44. Angular positions and velocities for double integrator problem. ..92

Figure 45. Double integrator torque controls, two satellites/one ground station. 93

Figure 46. Hamiltonian and costates for double integrator solution. 94

Figure 47. Double integrator without (left) and with torque smoothing (right), for two satellites/one ground station. 95

Figure 48. Satellite positions vs. DIDO solution and corresponding interpolated outputs for azimuth (left) and elevation (right) axes. 97

Figure 49. Antenna pointing accuracy from propagated satellite positions and solution outputs for azimuth (top) and elevation (bottom) degrees of freedom with and without torque smoothing. 98

Figure 50. Velocity and position trajectory integration for azimuth (left column) and elevation (right column). 100

Figure 51. Three satellite/one ground station scenario....101

Figure 52. Three satellite animation plots, progressing from top left to bottom right.....102

Figure 53. Benefit plot (*a.)*, azimuth (*b.*), and elevation (*c.*) plots and corresponding solution outputs.....103

Figure 54. Controls, states, costates, and Hamiltonian for three-satellite problem.....105

Figure 55. Four satellite, one ground station case. Satellite4 approaches GS1 (top) and crosses the Satellite3 ground track but remains further away, even at closest approach (bottom); satellite3 therefore remains GS1's target.....106

Figure 56. Four satellite/one ground station solution benefit curves and overlapping Gaussian snapshot (inset).....107

Figure 57. Two satellite, two ground station scenario.....108

Figure 58. Four panel progression of two-satellite scenario.....109

Figure 59. Benefit curve for two satellite two ground station problem 110

Figure 60. Azimuth and elevation trajectories for the satellites (blue, green) and interpolated outputs (red).....111

Figure 61. Three satellite/two ground station scenario.....112

Figure 62. Scenario start (top) and first transition at GS1.....113

Figure 63. First transition at GS2 (top) and second transition at GS1 (bottom).....114

Figure 64. Approaching second transition at GS2 (top) and near LOS at GS2 (bottom).....115

Figure 65. Benefit plot and link margins for three satellite/two station simulation.....116

Figure 66. Azimuth (top two) and elevation plots for each ground station, passing satellites, and superimposed interpolated output.....117

Figure 67. Ten satellite, one ground station scenario. This simulation is meant to saturate the station and force the tracking of the highest benefit targets only.....118

Figure 68. Benefit plot for ten-satellite case.....119

Figure 69. PROPCUBE satellites undergoing preparations for flight.....123

Figure 70. Illustration of MC3 stations and three PROPCUBE satellites.....124

Figure 71. MPIPE web-based user interface.....127

Figure 72. MC3 block diagram for SOC and remote station. 128

Figure 73. Typical PROPCUBE-compatible UHF configuration for an MC3 station.....130

Figure 74. National Instruments USRP 2922.....131

Figure 75. GNU Radio flowgraph for PROPCUBE downlink. 132

Figure 76. Waterfall plot showing received RF energy of PROPCUBE satellite (yellow box). Source: [102].....132

Figure 77. Pass planning flowchart.....134

Figure 78. Example of overlapping contact windows.....135

Figure 79. Flowchart for DIDO-integrated system on MC3.....136

Figure 80. Scenario start.....138

Figure 81. Scenario midpoint.....139

Figure 82. Scenario end.....139

Figure 83. Link margin and benefit plot for the two satellite, one station example.....140

Figure 84. Scenario start: Merryweather in view for both sites. Flora is on horizon for SDL and out of view for UNM. Both sites are tracking Merryweather.....141

Figure 85. First transition: The SDL site slews from the Merryweather to Flora target. UNM continues tracking Merryweather. 142

Figure 86. Second transition: Immediately after Flora was acquired at SDL, the UNM site began its slew from Merryweather to Flora.....143

Figure 87. Link margin and benefit plot for the two-satellite, two-station example.....144

Figure 88. Screenshot of actual PROPCUBE operations performed using SDL and UNM ground stations. 145

Figure 89. Two satellite, one ground station example with crossing orbits. .146

Figure 90. Trajectory plots for azimuth and elevation axes in crossing orbit scenario with automatic pointing inversion. 147

Figure 91. Four-panel progression of crossing orbit scenario. 148

Figure 92. Three satellite/two antenna test at HSFL. 149

Figure 93. Infinitely differentiable bump (left) and step (right) functions. Source: [54]. 154

Figure 94. Flora/Merryweather predictive model for PTSUR ground station based on initial azimuth angle and historical downlink performance. Source: [103]. 157

LIST OF TABLES

Table 1. Tracking progression of ten-satellite case. 120

Table 2. Solution for horizon-to-horizon contacts with no overlap. 121

Table 3. Symbol to MATLAB variable mapping. 163

Table 4. Symbol to MATLAB variable mapping 179

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ADCS	attitude determination and control system
AER	azimuth, elevation, and range
AFSCN	Air Force Satellite Control Network
AOS	acquisition of signal/satellite
API	Application Programming Interface
AST	apparent sidereal time
bps	bits per second
BVF	benefit value function
C2	command and control
COE	classical orbital elements
CONOP	concept of operations
COTS	commercial off-the-shelf
CU	cost units
DoD	Department of Defense
ECEF	Earth Centered Earth Fixed
ECI	Earth Centered Inertial
EIRP	equivalent isotropic radiated power
ENU	East North Up
EO	Earth observation
EOS	Earth observing satellite(s)
ESA	European Space Agency
FEP	front-end processor
GEO	Geosynchronous Earth Orbit
GOTS	government off-the-shelf
GSFC	Goddard Space Flight Center
HSFL	Hawaii Spaceflight Laboratory
ISRO	Indian Space and Research Organization
ISS	International Space Station

ISTRAC	ISRO Telemetry Tracking and Command
KKT	Karush-Kuhn-Tucker
LOS	loss of signal/satellite
MATLAB	Matrix Laboratory
MC3	Mobile CubeSat Command and Control
MEO	Medium Earth Orbit
MIP	Mixed Integer Programming
MLB	Malabar transmitter annex station
MOC	Mission Operations Center
MPIPE	MC3 Picosat Interface Pipeline Extension
NASA	National Aeronautics and Space Administration
NP	nondeterministic polynomial time
NPS	Naval Postgraduate School
NRL	Naval Research Laboratory
NSF	National Science Foundation
OPSPLAN	operations planners
OPTimIS	Operations Planning Timeline Integration System
OP	orienteering problem
OSTPV	On-board Short Term Plan Viewer
PROPCUBE	Picosats Realizing Orbital Propagation Calibrations Using Beacon Emitters
R&D	research and development
RF	radio frequency
RPE	real-time planning engineers
SATRN	SmallSat Agile Transmit Receive Network
SCaN	Space Communications and Navigation
SDL	Space Dynamics Laboratory
SDR	software-defined radio
SEM	space environment monitoring
SGP4	Simplified General Perturbations Satellite Orbit Model 4
SOC	Satellite Operations Center

SSA	space situational awareness
TCP/IP	Transmission Control Protocol/ Internet Protocol
TDRS	Tracking and Data Relay System
TLE	two-line element
TSP	traveling salesman problem
TT&C	telemetry, tracking, and control
UAF	University of Alaska, Fairbanks
UDP	User Datagram Protocol
UNM	University of New Mexico
USCGA	United States Coast Guard Academy
USNA	United States Naval Academy
UTC	Universally Coordinated Time
V&V	verification and validation
VPN	virtual private network

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

Though the PhD might seem like a largely individual effort for the student, in practice, reaching this milestone could only have been possible thanks to the dedicated guidance and encouragement from many teachers, mentors, colleagues, and loved ones.

First, I would like to thank advisor and mentor Dr. Jim Newman who provided the forum in the Space Systems Academic Group within which I could learn and grow both professionally and personally. To Dr. Mark Karpenko, thank you for the countless hours spent in the weeds hashing out the many details that made their way into this dissertation. Dr. Mike Ross, thank you for bringing me up to speed on the compelling world of optimal control and its myriad applications. Committee members Dr. Marcello Romano and Dr. Frank Giraldo offered their guidance both in the initial coursework and as the dissertation progressed toward the final product. Their expertise is deeply appreciated. Alison Scharmota in the Graduate Writing Center graciously helped with edits.

Working alongside the staff in our lab has been nothing short of a privilege. Friends and colleagues Jim Horning, David Rigmaiden, Lara Magallanes, Wenschel Lan, the ever-energetic Prof. Rudy Panholzer, and many others made each day great fun. The dozens of military thesis students who came through our lab over the years are remarkable individuals. I hope that they found their time in the lab as rewarding as I found my time with them.

My appreciation also goes to the many sponsors, collaborators, and organizations who have shown their support for the MC3 project from its inception several years ago. Together we have built a diverse community of users who all agree that small satellites offer an exciting future worth fighting for.

Finally, my heartfelt thanks to the dear family and friends whose kind words of encouragement helped carry this work to the finish line. Thank you for supporting, and occasionally putting up with, these lofty dreams of advancing spaceflight to make life better for all on Earth.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Small spacecraft have recently emerged as an attractive and highly capable platform that enable scientists and engineers to perform missions of ever-increasing complexity [1]. The popularity of these spacecraft can be attributed in part to the falling cost of development, miniaturization of complex electronics, shortened development life cycles, and increasingly routine access to space as rideshare/secondary payloads [2]–[4]. The total cost of flying a small satellite mission is often orders of magnitude lower than a traditional, large, albeit more capable, counterpart. The lowered barriers to entry allow a growing number of worldwide participants from academia, government, and commercial industries. Given the increased popularity of small spacecraft, lowering costs, and routine access to space, the small satellite industry is poised to produce spacecraft in large quantities and field constellations of considerable size [5].

Traditional constellations composed of large satellites have been typically operated by dedicated ground networks; however, the low-cost methodology of the small satellite industry preclude_s many institutions from building such dedicated networks. Rather, the majority will share ground-based infrastructure requiring flexibility for mission-specific details and considerable automation to reduce operating costs. This research begins to address the problem of operating satellite populations that number in the hundreds or thousands from a small network of distributed ground stations by developing a scalable approach for automating and optimizing the mission operations component. This dissertation reviews the current state-of-the-art for optimization techniques, presents a unique mathematical problem formulation, and demonstrates its implementation in simulation and with the use of orbiting CubeSats and the ground stations that operate them.

A. SMALL SATELLITES AND CUBESATS

The traditional definition of a small satellite varies somewhat, though it generally refers to a spacecraft with a total mass less than 500 kg [6]. The first satellites launched between 1957–58 (Sputnik-1 and -2, Explorer-1 and -3, and Vanguard-1) all weighed under 100 kg. As launch vehicles gained more lift capacity, so too did spacecraft increase in size,

mass, and complexity. The primary payloads launched for many decades after these first missions could no longer be classified as small satellites. The cost to develop these spacecraft and launch vehicles also remained high; only governments such as the United States and Soviet Union initially launched and proliferated space-based technologies.

As launches have become more commonplace in recent decades, small satellites began reemerging in popularity. These satellites were often launched as secondary payloads—hitchhikers on a rocket already scheduled to fly. Since a dedicated launch could cost tens, if not hundreds, of millions of dollars [7], utilizing rideshares for small, less expensive payloads became an increasingly attractive option for missions that sought to attain Earth orbit, but did not necessarily require particular orbital characteristics (i.e., periods, inclinations, eccentricities, etc.). A lower cost to orbit also allowed programs to accept a risk-tolerant posture. For example, flying a small satellite became cost-effective enough to increase the Technology Readiness Level of an instrument or subsystem slated to fly on a larger, more expensive spacecraft in the future. A loss of this pathfinder small satellite or its instrument could be considered acceptable since the primary, more expensive mission was not jeopardized.

In the 1990s, the concept of low-cost and high-risk missions was a natural fit for universities seeking to accomplish unique science missions while providing their students opportunities for training in fields such as spacecraft design, testing, integration, and operations. The cost, mass, volume, power, and data constraints typically associated with small satellites often posed rigorous design challenges while mission life cycles fell within the timespan of a graduate or doctoral degree. University nanosatellite missions such as Stanford's OPAL [8] and the Naval Postgraduate School's PANSAT [9] were developed by staff and students at universities and flown at the turn of the 21st century as secondary payloads.

Early university small satellite missions, though successful, highlighted the need to develop a set of common standards for integrating secondary payloads to launch vehicle to further reduce the cost and lead time for flight. As part of a collaboration between Stanford University and California Polytechnic State University, San Luis Obispo, the CubeSat

standard was first published in 1999 to help universities worldwide better perform their engineering and science experiments with these very small satellites [10]. The standard primarily determined the satellites' size and mass constraints. The constraints were defined in units where each unit was represented by a 10 cm cube weighting 1 kg. A satellite's size could, therefore, be described in a unit quantity, with one unit (1U), three units (3U), and six units (6U) being the most common (Figure 1). The primary purpose of the CubeSat standard was to define a generic launch vehicle adapter into which any satellite could integrate.

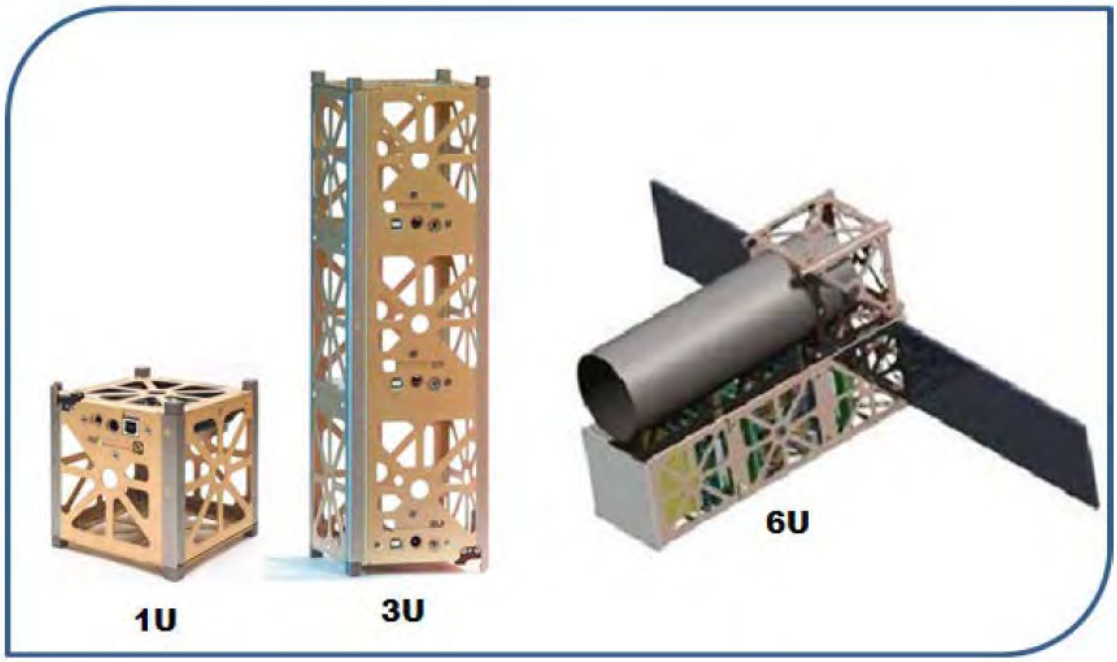


Figure 1. CubeSats in 1U, 3U, and 6U sizes. Source: [11].

Launch vehicle adapter standardization is one reason for the rising popularity of CubeSats. Additionally, the miniaturization of low-power, custom and commercial off-the-shelf (COTS) electronic components greatly enabled sophisticated on-orbit technologies in the highly constrained CubeSat form factor, such as attitude determination and control systems (ADCS), remote sensing, powerful onboard processing, abundant solid state memory, deployable solar cells, and high data rate communications. Emerging

technologies such as 3D printing plastics and metals have further enabled complex mechanical designs at reduced costs.

B. LARGE CONSTELLATIONS

Iridium operates one of the largest existing constellations which consists of 66 satellites and several active spares in 6 equally spaced circular orbital planes at a 780 km altitude as shown in Figure 2. The satellites that make up the Iridium constellation are relatively large (approximately 680 kg each) and required 21 launches between 1997 and 2002 [12]. At the time of writing, this constellation is being repopulated with newer satellites called Iridium NEXT.

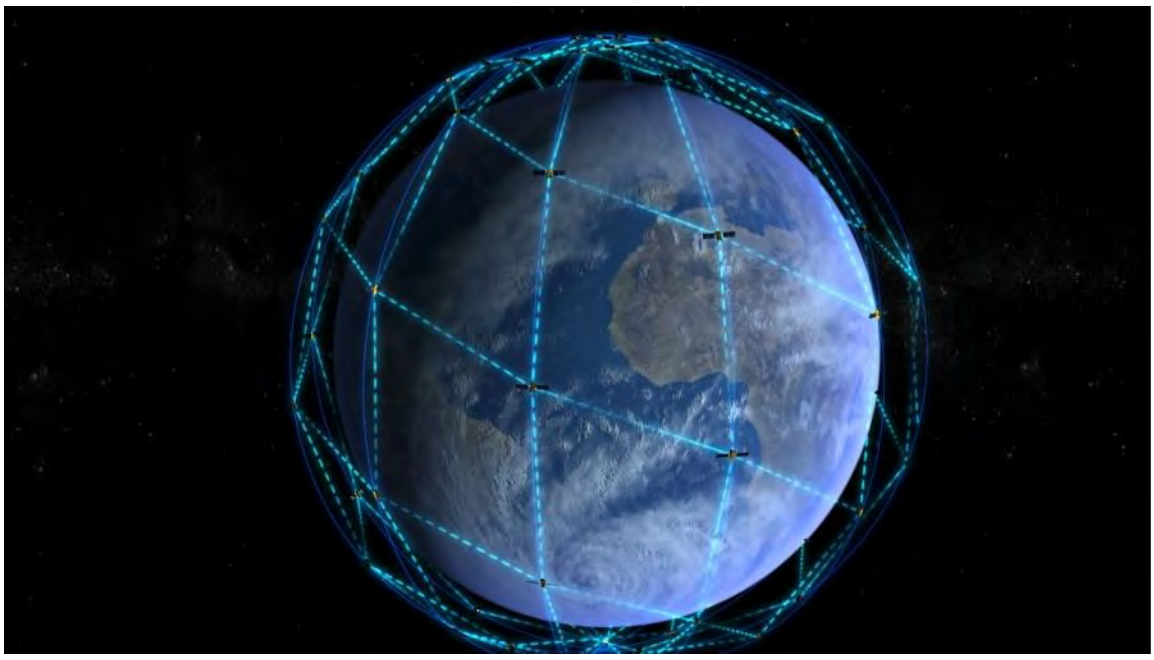


Image retrieved from https://www.thalesgroup.com/en/worldwide/space/news/space-qa-all-about-iridium-next

Figure 2. The Iridium constellation.

The substantial cost of building, fielding, and operating an Iridium-class constellation had been generally prohibitive for all but a handful of private companies and governments. Due to miniaturization of flight hardware and decreasing launch costs, large constellations for

applications such as telecommunications or remote sensing have now become viable concepts. As the decreasing cost to develop and launch hardware has become increasingly capable, industry and governments are poised to create large constellations of small satellites.

The recently formed company Planet (formerly Planet Labs) [13] has, as of this writing, launched over 200 CubeSats into Low Earth Orbit (LEO) for Earth observation (EO) applications and has quickly become the operator of the largest satellite constellations ever fielded (in terms of quantity of spacecraft simultaneously operated). Planet's ability to leverage rideshare capabilities to the International Space Station (ISS), as shown in Figure 3, and low-cost dedicated launches has allowed the company to replenish and expand its constellation multiple times per year.



Image retrieved from https://www.nasa.gov/content/set-of-nanoracks-cubesats-deployed-from-space-station

Figure 3. Two Planet *Dove* satellites deployed from the International Space Station.

Several companies are attempting to field constellations that will provide a near real-time snapshot of global data products based on EO applications. For example, BlackSky [14] will capture global imagery with a constellation of 60 satellites. Continuous GPS radio occultation weather monitoring will be done by 12 satellites developed by planetIQ, and Spire has begun conducting a wide range of radio signal exfiltration using over 50 satellites [15]. Additionally, O3b, OneWeb, and SpaceX [16], [17] are developing the capability to provide broadband internet coverage from space using large satellite constellations.

The U.S. government also maintains an active interest in small satellite constellations. In particular, the National Aeronautics and Space Administration (NASA) [18]–[20], National Science Foundation (NSF) [21], and Department of Defense (DoD) [22] look to further advance space environment monitoring (SEM), EO, space situational awareness (SSA), and communications.

Following guidance in the 2010 National Space Policy, the 2011 U.S. National Security Space Strategy, and the 2015 Space Domain Mission Assurance white paper [23], the National Security Space community has been directed to ensure cost-effective survivability of space capabilities, which includes defensive operations, reconstitution, and resilience. Of these three capabilities, operational resilience is specified as the end goal, allowing National Security systems to accomplish their mission with greater likelihood, shorten downtime, and access a wider range of scenarios, conditions, and threats in spite of hostile actions or adverse conditions. To achieve resilience, six characteristics were identified as primary enablers: disaggregation, protection, distribution, proliferation, diversification, and deception. All six of these attributes are possible with small satellite constellations. For example, the ability to disaggregate key sensors means that no one spacecraft carries all assets critical for mission success. Distribution, diversification, and proliferation are all attainable with increasing quantities of low-cost satellites launched into varied orbits. The ability to protect these assets is also improving with miniaturized electronics and increased computational power enabling sophisticated forms of onboard encryption. Finally, the small size of these satellites provides an inherent ability to deceive

adversaries, given the immense technological requirements needed for effective SSA to remotely characterize the satellites' missions. Though not always a replacement for larger assets, small satellite constellations have attracted attention as possible solutions for more resilient operational capabilities [7].

Access to capacity-constrained ground stations, staffing of mission operators, limited onboard power, and increased data products all represent potential bottlenecks for effective command and control (C2) of large constellations of small satellites [24]. Consequently, large constellations of small satellites could encounter significant ground station resource constraints, which may altogether eliminate their advantages. Given these bottlenecks, traditional approaches to conduct C2 involving many operators do not scale to large constellations, even if cost is not a primary constraint [25]. Automation will be a requirement for managing each small satellite's power and data budgets, system health, anomaly resolution [26] and ground-based resources enabling the two-way communications link. Each contact event may only last a minute or two before the ground station resources divert to the next passing spacecraft. The data transfer would be resumed automatically over the next available station, producing sporadic bursts based on ground resource availability. The flow of information could be more likened to a stream of information instead of discrete satellite contacts staffed by operators. Much how cellular telephones transmit digital information while seamlessly transferring between terrestrial towers without any user input, so too should the various constellations of the future transition between diverse ground and space-based communications infrastructure.

C. COMMUNICATIONS ARCHITECTURES

There exist several distributed ground station networks servicing a growing population of satellites. Traditional architectures have been established for many decades and consist of globally-dispersed radio frequency (RF) apertures that service spacecraft in Low Earth Orbit, Medium Earth Orbit (MEO), Geosynchronous/Geostationary Earth Orbit (GEO), and deep space.

The Air Force Satellite Control Network (AFSCN) [27] is an example of a high-cost, high-reliability ground segment supporting a diverse population of DoD satellites for operational uses. There are several satellite operation centers (SOCs) that are permanently staffed with operators tasking the satellite population through the worldwide remote tracking stations. Though automation exists, a significant number of military and civilian staff plan, execute, and maintain the daily activities of the network. The network has two barriers that limit usage of these assets by U.S. government small satellite missions: high cost and low priority. The operational cost prohibits the use this network for many low-cost small satellite missions, the primary purposes of which have been technological demonstrations and research and development (R&D). Given the nature of these R&D missions, their tasking priority is often much lower than those of operational national security assets. In the future, it is possible that small satellite constellations charged with operational tasks critical to national security and the safety of military service members will routinely use the AFSCN assets.

The NASA Space Communications and Navigation (SCaN) program's Near Earth Network (NEN) and Space Network (SN) are also a well-established and highly capable collection of assets that enable high-reliability communications for missions supporting Earth science, remote sensing, and human spaceflight operations. The NEN operates a global network of government and commercial 11-meter-class parabolic antennas whose SOC is based at the NASA Goddard Space Flight Center (GSFC) [28]. Mission-specific activities are performed at mission operation centers (MOCs) that do not have to be co-located with the SOC. The SN primarily consists of the Tracking Data Relay System (TDRS) satellites. The satellites relay communications from other spacecraft to the primary ground station located at the White Sands Test Facility, acting as a bent pipe for mission operators. Researchers propose using these assets with small satellites, both individual free-flyers and constellations. Efforts are underway to characterize network loading and schedule optimization to support this increase in usage. The cost model is also under evaluation so as not to pose significant barriers to entry for comparatively low-cost missions. A primary enabler for lowered costs will be the optimization of tasking and

resource allocation. The concept of operations (CONOP) for such an architecture is shown in Figure 4.

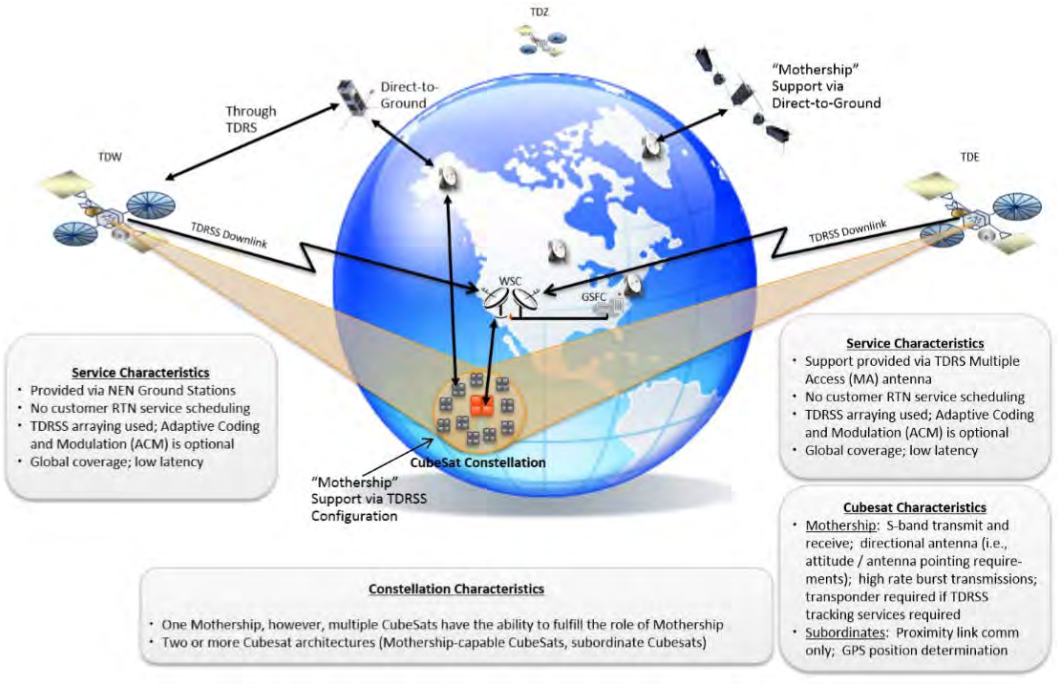


Figure 4. SN and NEN proposed architecture to support CubeSats and CubeSat constellations. Source: [28].

Several commercial solutions have been created to support the increase in small satellite mission operations. SSC Space U.S. Inc., formerly Universal Space Network (USN) [29], Kongsberg Satellite Services (KSAT) [30], and ATLAS Space Operations [31], [32] all consist of globally-distributed, remotely-operated ground stations that are progressively adapting to support small satellite constellations. For example, Figure 5 shows a layout of KSAT's global network. Each network has a cost model that typically charges users per contact or per unit of data passed through the network [33]. The typical CONOP involves a user scheduling time on an asset and using the ground station as a bent pipe for performing remote mission operations with their spacecraft. Though some automation and schedule optimization exists, it will be important for these companies to continue finding ways to maximize network capacity for an increasing user population.



Figure 5. KSAT commercial ground station locations. Source: [30].

To support educational and amateur satellite activities there have been several efforts to stand up privately owned, loosely organized ground stations that can be used at little to no cost. Called Federated Ground Station Networks (FGSNs) [34], [35], these assets are typically owned by individuals or academic institutions involved in the development and operation of small satellites. The missions typically come with very little funding and can accept a high-risk, low-reliability CONOP for the ground segment. An effort to bring together many amateur stations into an FGSN with pre-established standards is the Global Educational Network for Satellite Operators (GENSO) [36], [37]. The program gained little traction in the United States, though it has been relatively successful in Europe and Asia.

The Mobile CubeSat Command and Control (MC3) [38]–[43] ground station network was created by the DoD as an alternative to high cost assets such as the AFSCN for U.S. government small satellite R&D missions. The network SOC is located at the Naval Postgraduate School and primarily consists of low-cost COTS components and government-off-the-shelf (GOTS) software hosted at partner institutions in geographically diverse locations. There are currently nine remotely-operated stations fielded around the United States that constitute the MC3 network. Figure 6 illustrates the ground station locations for MC3. The site locations are as follows:

- Hawaii Space Flight Laboratory (HSFL)—Honolulu, HI
- University of Alaska, Fairbanks (UAF)—Fairbanks, AK
- Naval Postgraduate School (NPS)—Monterey, CA
- Space Dynamics Laboratory (SDL)—Logan, UT
- University of New Mexico (UNM)/ Cosmiac—Albuquerque, NM
- Air Force Institute of Technology (AFIT)—Dayton, OH
- United States Naval Academy (USNA)—Annapolis, MD
- Malabar Transmitter Annex (MLB)—Melbourne, FL
- United States Coast Guard Academy (USCGA)—New London, CT

The primary mission of MC3 is to develop and operate ground segment infrastructure that can support a wide array of R&D satellites at reduced costs. The MC3 network and the satellites it operates act as a real-world testbed for many educational and research applications, including the algorithms developed as part of this research. Consequently, many of the results presented in this dissertation involve MC3 assets, but can be generalized to other ground station networks and architectures.



Figure 6. MC3 network ground station locations.

The ground station networks described previously will be responsible for operating a disaggregated population of small satellites with unique requirements such as custom-tailored C2 software, different radio frequencies, waveforms, and data protocols, and varying constraints on timeliness for the mission stakeholders. Traditionally, a large team of operators would be trained to take these constraints into account and manually deconflict assets to keep the missions on track. This process is labor-intensive and does not scale to populations of satellites in the hundreds or more, given the number of deconfliction events that would need to be considered every day. The nature of small satellite missions is that they are cost-constrained, and as such, automated mission operations show the greatest promise for keeping costs down while servicing a vast number of diverse satellites performing separate missions.

D. SIMPLIFIED PROBLEM DESCRIPTION AND HEURISTIC METHODS

The size and onboard power constraints of modern, very small satellites typically limit the spacecraft transmitters' radiated power output. This will consequently require

ground-based directional antennas with corresponding signal amplification to receive downlinks while tracking the satellites. Described below are a few simplified scenarios that are common to the multiple small spacecraft communications problem when utilizing ground-based directional antennas. The scenarios make the following assumptions:

- The ground stations utilize directional antennas that are limited to only tracking one satellite at a time.
- Only one station can transmit to a particular satellite at any point in time. The authority to transmit can be passed between multiple stations during a single contact if they are simultaneously in view of the satellite. This mitigates self-inflicted uplink interference that may prevent commands from being received by the spacecraft.
- Multiple stations can track a single satellite and receive its downlink. This scenario aids in receiving telemetry when marginal signal performance is being experienced. It is assumed that there exists a ground-based solution to detect and handle duplicate data sets.
- The satellites are in LEO and MEO, where communications windows are finite, as they pass over points of interest on Earth. Satellites in GEO can also be abstracted into this model, however their persistent availability makes for a less generalized case.

1. One Satellite, One Ground Station

The simplest communications problem involves a single satellite passing over a single ground station. The common terminology associated with these contact events is the acquisition of signal/satellite (AOS) and loss of signal/satellite (LOS), corresponding to the beginning and end of a contact window, respectively. When involving LEO and MEO satellites, this often corresponds to the times at which the spacecraft rises above and descends below the horizon for a ground-based observer. GEO and interplanetary missions also use AOS and LOS terminology. However, contacts are scheduled based on other

factors in addition to line-of-sight visibility from a ground station. Figure 7 shows a simplified contact for three snapshots in time.

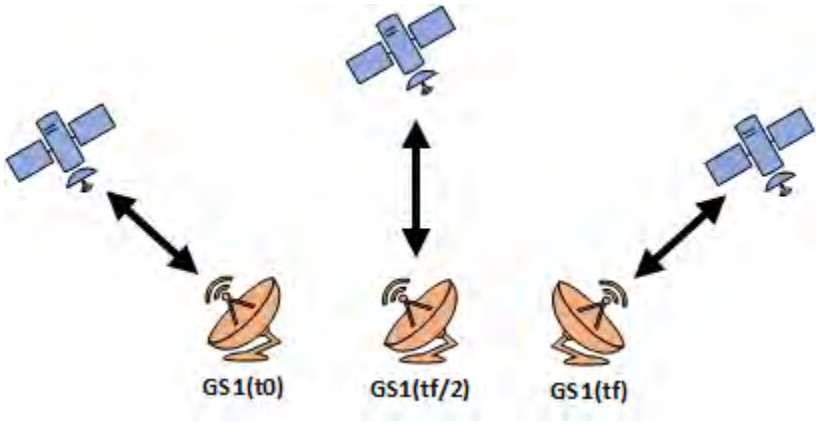


Figure 7. Simplified one-satellite one-ground station contact scenario.

Figure 7 depicts three snapshots as the satellite passes from left to right at the initial (t0), middle (tf/2), and final (tf) time points. The ground station (GS1) antenna tracks the satellite from AOS to LOS. Communications are bidirectional as commands are sent from the ground and data is downlinked from space. Figure 8 shows two ways of illustrating the GS1 resource allocation for the duration of the contact.

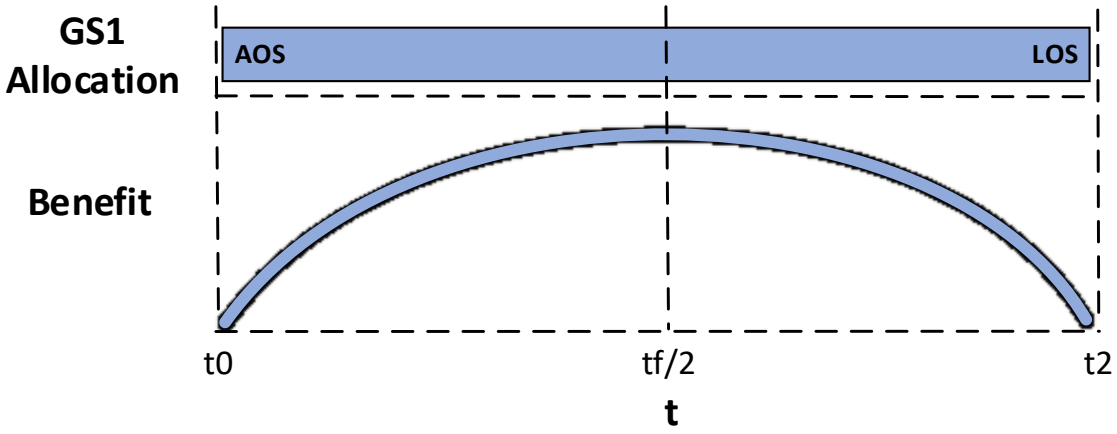


Figure 8. Schematic of GS1 schedule allocation and signal strength/ benefit curve.

Given the assumptions just described, the GS1 resource is constrained to only communicate with a single satellite from AOS to LOS. This is denoted as the horizontal blue AOS/LOS bar in Figure 8. Without other satellites competing for ground-based resources, and having not considered any additional performance parameters, the entire AOS to LOS window could be an appropriate allocation for GS1.

The time-varying Benefit curve shown in Figure 8 offers another perspective for looking at the GS1 schedule allocation. The ability to successfully transmit digital information over long distances is directly influenced by the signal strength at the receiver. Affecting the strength are such factors as transmitting power, free-space path loss, environmental noise at the receiver, component losses, and antenna orientation and gain at both transmitting and receiving locations. The equations governing these relationships are further explored in Chapter II. For this simplified discussion, we shall assume no antenna pointing losses, constant component losses, and no environmental noise. The only time-varying aspect of the space-to-ground link is the free-space path loss, which is directly related to distance. As the satellite approaches the station, the signal strength for both uplink and downlink is increased at the respective receiver. Conversely, as the satellite recedes from the station, the signal strength diminishes. It can therefore be said that the most advantageous opportunity to transmit and collect data from the passing satellite would be at the point of closest approach, where signal strength is highest and the probability of successfully transmitting messages is greatest. Reasoning that data transfer is of interest to the mission, the greatest benefit to the mission is at a point of the contact, which enables the greatest throughput of information. This is represented as the highest point of the curve shown in Figure 8, corresponding to the point of closest approach for the satellite in our idealized case.

2. Two Satellites, One Ground Station

The next idealized scenario involves two satellites in the same orbit with a small position offset passing over a single ground station, shown in Figure 9. The goal of this scenario is to show various strategies for allocating the ground resource to serve both spacecraft.

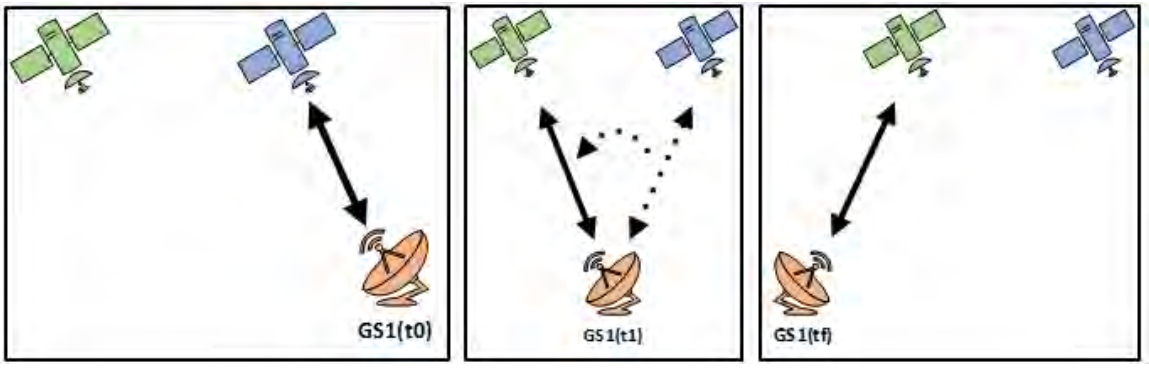


Figure 9. Idealized two-satellite one-ground station case.

The GS1 ground station is now tasked with serving two satellites (blue and green). Three snapshots are shown in Figure 9 for the AOS of the blue satellite (t_0), transition from blue to green (t_1), and LOS of the green satellite (t_f). While t_0 and t_f are fixed due to the orbital mechanics of the pass, the transition point t_1 is flexible, depending on the method determining when to switch from the blue satellite to the green, as expressed in Figure 10.

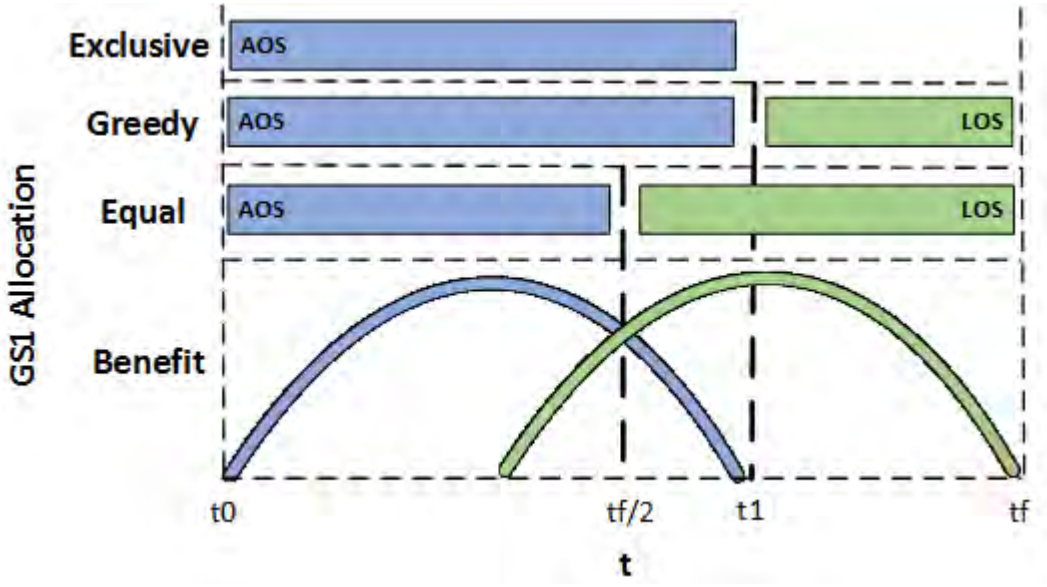


Figure 10. GS1 allocation plot for exclusive, greedy, equal, and benefit value schemes.

The resource allocations in Figure 10 show four approaches that could be implemented for serving the two satellites. The *exclusive* method is commonly used in networks where starvation [44] is not a major concern. The first satellite to come into view is serviced from AOS to LOS, and conflicting satellites will be deferred to a future contact. Variations on this method involve prioritization schemes (of the two, choose the one assigned a higher priority value), or elevation angle/range filters which choose the contact that has the best predicted signal strength and data downlink potential.

The *greedy* method seeks to choose a locally optimal solution. In this application, an unused station provides less benefit than one communicating with a satellite, regardless of performance. Therefore, the greedy method employs a first-come-first-served approach that tracks the blue satellite from its AOS to its LOS with no consideration for link quality. It then tasks GS1 to reorient its antenna at t_1 and track the green satellite for the remainder of the contact window. While this approach serves both satellites, it spends a portion of the contact tracking a fading signal from the blue satellite instead of acquiring the approaching green satellite. This method may be adequate for certain missions with minimal data requirements, but it is not globally optimal.

If using an *equal time distribution* method, the transition point between blue and green satellites would occur exactly in the middle of the contact window at $t_f/2$. This method assumes that the greatest benefit to the overall mission can be achieved by providing identical contact windows for each GS1 user.

The time-varying benefit case of Figure 10 shows an idealized signal quality metric for both satellites. The point of overlap at $t_f/2$ represents the point at which the signal quality from the green satellite begins to exceed that of the blue satellite. This could be a natural transition point for the GS1 antenna to provide the most benefit for both missions in the contact window. In contrast, the equal time distribution method does not factor in data throughput or signal performance parameters. For an idealized case such as the one shown in Figure 10, the ideal transition point for the equal time distribution case is identical to that suggested by the time-varying benefit plots. Once non-idealized situations would

arise, the equal time distribution plot would become sub-optimal with respect to data throughput for the two satellites.

3. Three Satellites, One Ground Station

Figure 11 now introduces a third satellite serviced by GS1. This additional satellite (shown in yellow) is in the same idealized orbit as the previous examples, however its spacing is non-uniform with respect to the blue and green satellites. There are now two transition points for GS1 to consider for its resource allocation schedule. The first transition point at t_1 selects between the blue and green satellites and the second transition point at t_2 corresponds to the switch between blue and yellow. Figure 12 shows the corresponding allocation curves for the exclusive, greedy, equal, and benefit value scenarios.

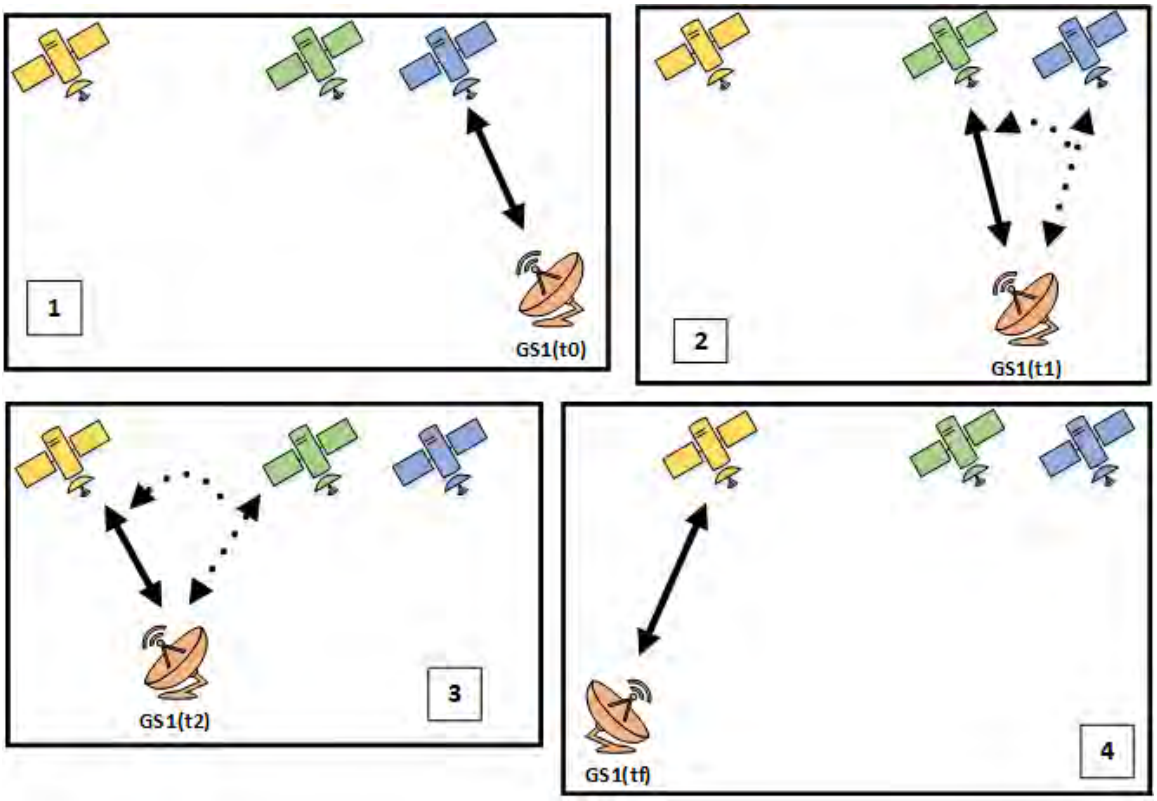


Figure 11. Three satellite contact scenario for one ground station.

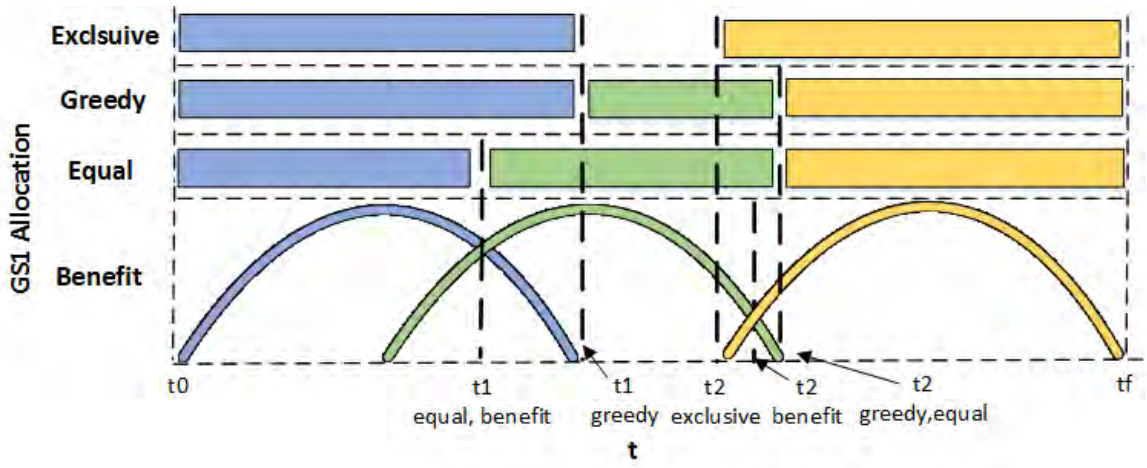


Figure 12. Allocation curves for three satellite, one ground station example.

The exclusive method is the easiest to implement and is often based on a simple scheduling *heuristic*, or shortcut/simplification, of starting a task at the AOS of a contact. The GS1 resource is allocated until the pass ends at LOS and any other overhead satellites will not be considered. In the above example, the blue satellite is serviced for its entire contact window. The next available contact is when an AOS occurs with GS1 unallocated, which is the yellow satellite. The green satellite is bumped from the schedule entirely, to be serviced at a future time.

The greedy approach starts with having tracked the blue satellite for the entire duration of its contact, and then switches to the green satellite. However, the timing of the switch is irrespective of the position of the green satellite, so the contact occurs at the peak, after having missed a portion of usable contact time. The green satellite is tracked until its LOS, at which point a transition is made to the yellow satellite. The total contact window for the green satellite is significantly shortened.

The equal time method provides identical contact windows for both the blue and green satellites. The limiting factor is the equal time distribution before the first two satellites. The third (yellow) satellite contact start is delayed slightly, but still receives more contact time than the first two satellites based on its offset position. The equal time method

seeks to spread contact time as uniformly as possible across the available satellites for the GS1 resource.

Lastly, the benefit value functions vary as a function of satellite distance with respect to the ground station. The transition points would therefore occur at the thresholds where the benefit of the subsequent satellite exceeds that of the one currently being serviced.

4. Two Satellites, Two Ground Stations

The next example introduces a second ground station, denoted as GS2, illustrated in Figure 13. The mechanics of greedy, equal, and benefit value allocations remain identical to those of the one ground station example, however the second ground station now includes time allocations offset in proportion to its geographical location, with respect to GS1.

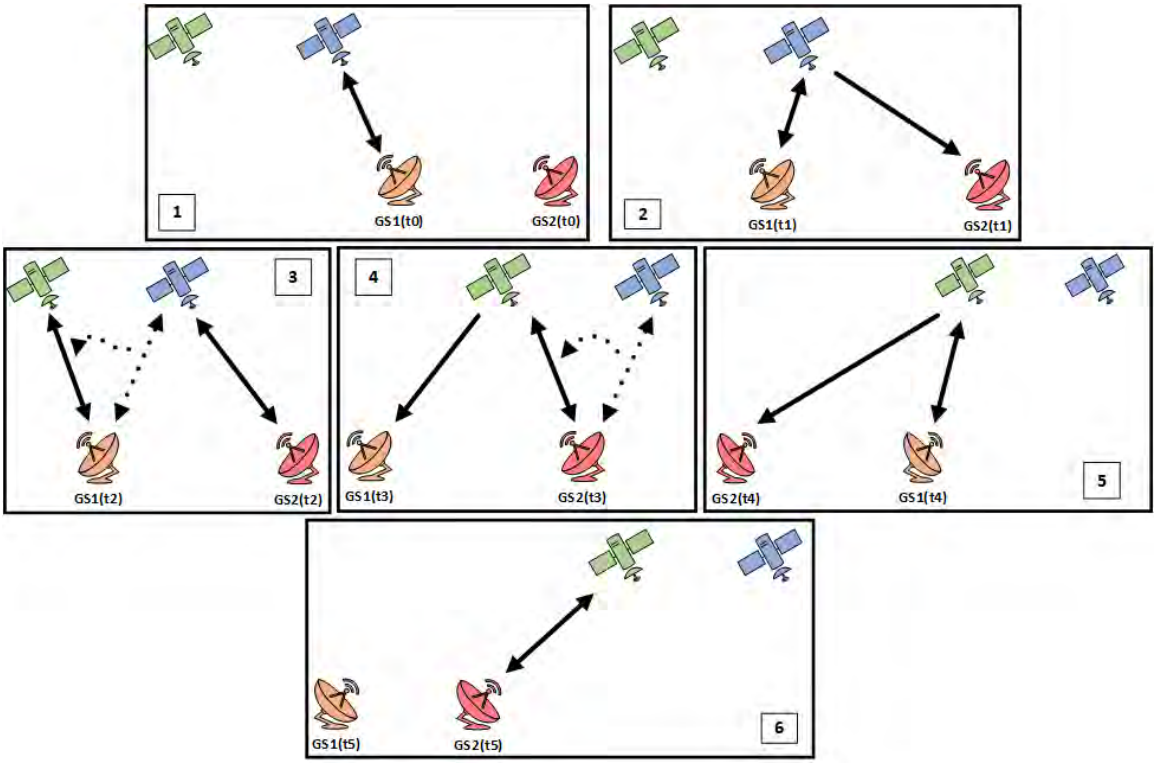


Figure 13. Two satellite, two ground station case.

This scenario involves multiple AOS and LOS events (at t_0, t_1 and t_4, t_5, respectively) as well as two transition points at t_2 and t_3. At t_0, the blue satellite comes into view for GS1, which initiates bidirectional communications. Sometime later, at t_1, the blue satellite also comes into view of GS2. However, due to the longer distances, GS1 retains custody while GS2 is only receiving downlink. At t_2, GS1 transitions from the blue to the green satellite, which allows GS2 to radiate to the blue satellite. At t_3, the blue satellite contact has finished and GS2 transitions to the green satellite. GS1 is now in a listen-only mode for t_4 and t_5 until the contact has terminated. Much like the previous examples, the satellite positions with respect to the ground stations are determined by the orbital mechanics of the scenario, however the transition points during the contact can be specified algorithmically, as shown in Figure 14.

We first revisit the exclusive contact method, which initiates a satellite contact at AOS when the ground station resource is unallocated. In the two ground station case, some logic can be applied to better allocate the ground resources and reduce starvation in satellite servicing. Figure 14 shows two exclusive strategies, A and B. In Exclusive A, GS1 contacts the blue satellite and drops the green satellite contact. Since GS2 is unallocated, it can pick up the green satellite contact after ignoring the initial blue contact. Exclusive B is the inverse case, where GS1 takes the green satellite and GS2 contacts the blue satellite. This method reduces starvation in a constellation that has approximately the same number of satellites as ground stations. If the satellite population outgrows the number of ground stations, this method loses the ability to service all of its satellites in a timely manner, particularly when the satellites are in diverse, randomly distributed orbits.

The greedy method disproportionately allocates ground station time to the blue satellite, as it arrived first. The unequal distribution is now compounded with the addition of a second ground station. While the blue satellite receives significant contact time from both ground stations, the green satellite receives markedly less. This example illustrates how the sub-optimal effects of the greedy method compound as the number of ground stations increase.

Dividing contact time equally between two satellites generates symmetric allocations at each station. The transition points also correspond directly to the benefit value curves. We can begin assessing the total mission benefit from servicing both satellites by summing the individual benefit contributions into a single curve, shown at the bottom of Figure 14.

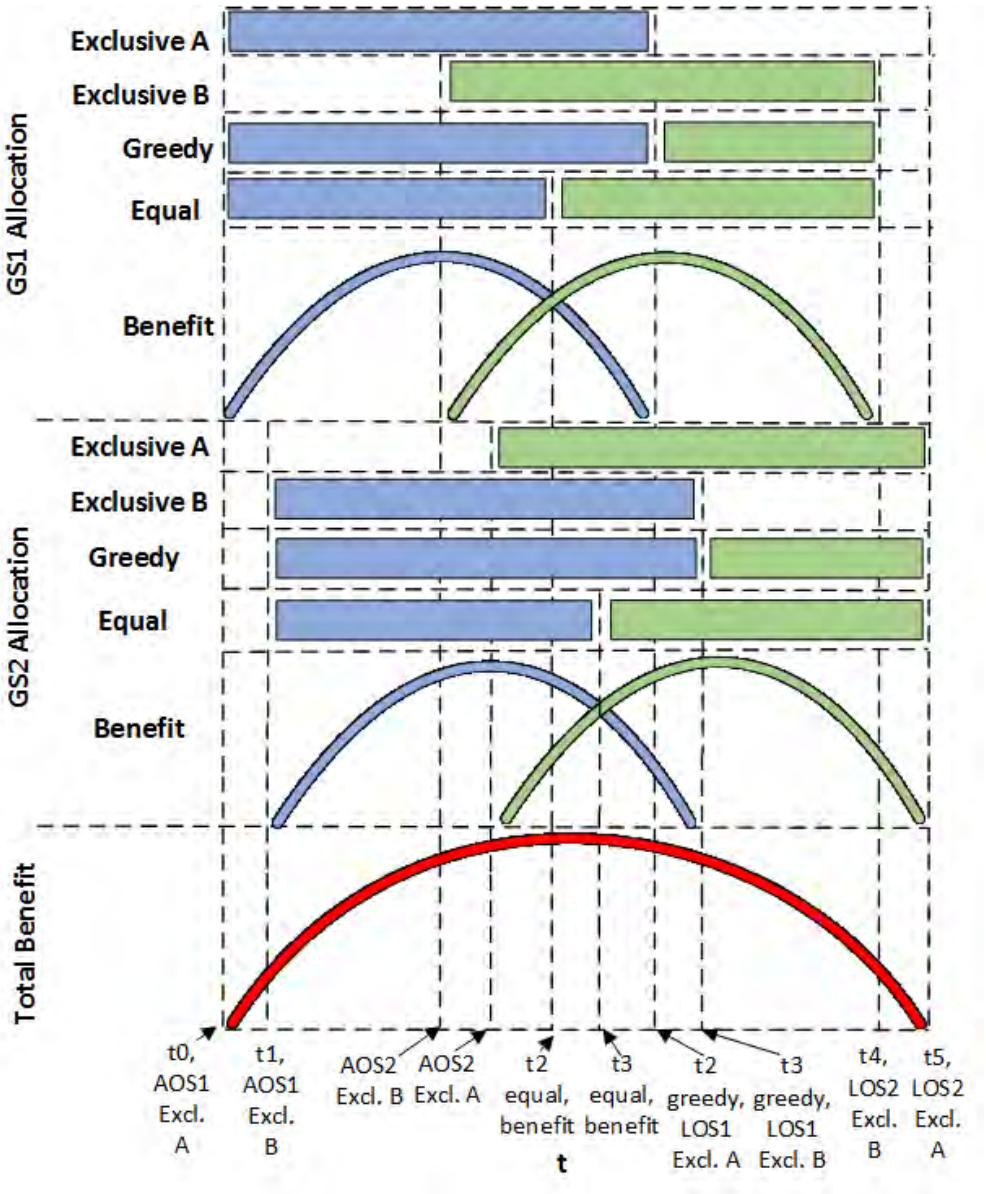


Figure 14. Resource allocation curves for two satellite/two ground station case.

This curve shows the combined value of servicing the passing satellites when considering multiple stations. The peak total benefit is centered between transition points at t_2 and t_3 for the individual benefit value curves. The mechanics of using this total benefit summation are described in Chapter II.

5. Three Satellites, Two Ground Stations

The final simplified example considered in this section, illustrated in Figure 15, involves three satellites and two ground stations. This scenario involves four transitions at t_2, t_3, t_4, and t_5 as the blue, green, and yellow satellites pass over GS1 and GS2. Much like the two satellite-two ground station example, Figure 16 shows the compounding effects involved with multiple stations, which accentuate the disproportionate distribution of contact time experienced in the greedy method.

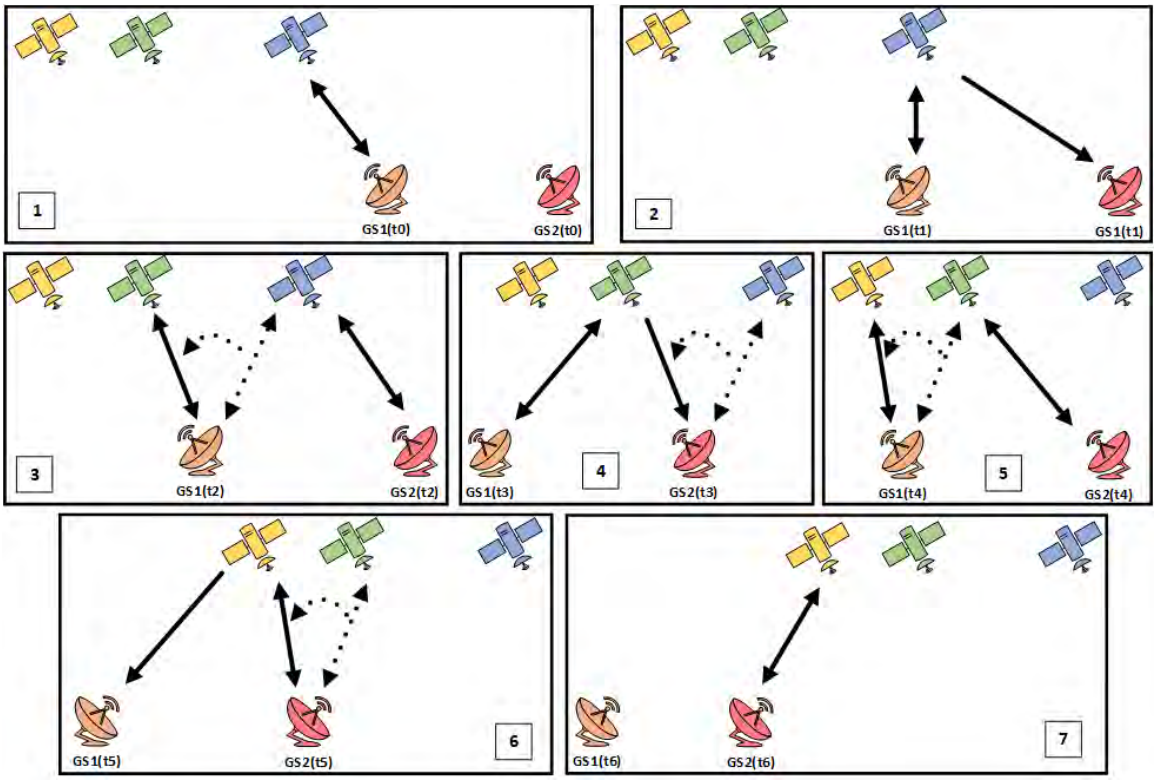


Figure 15. Three satellite/two ground station scenario.

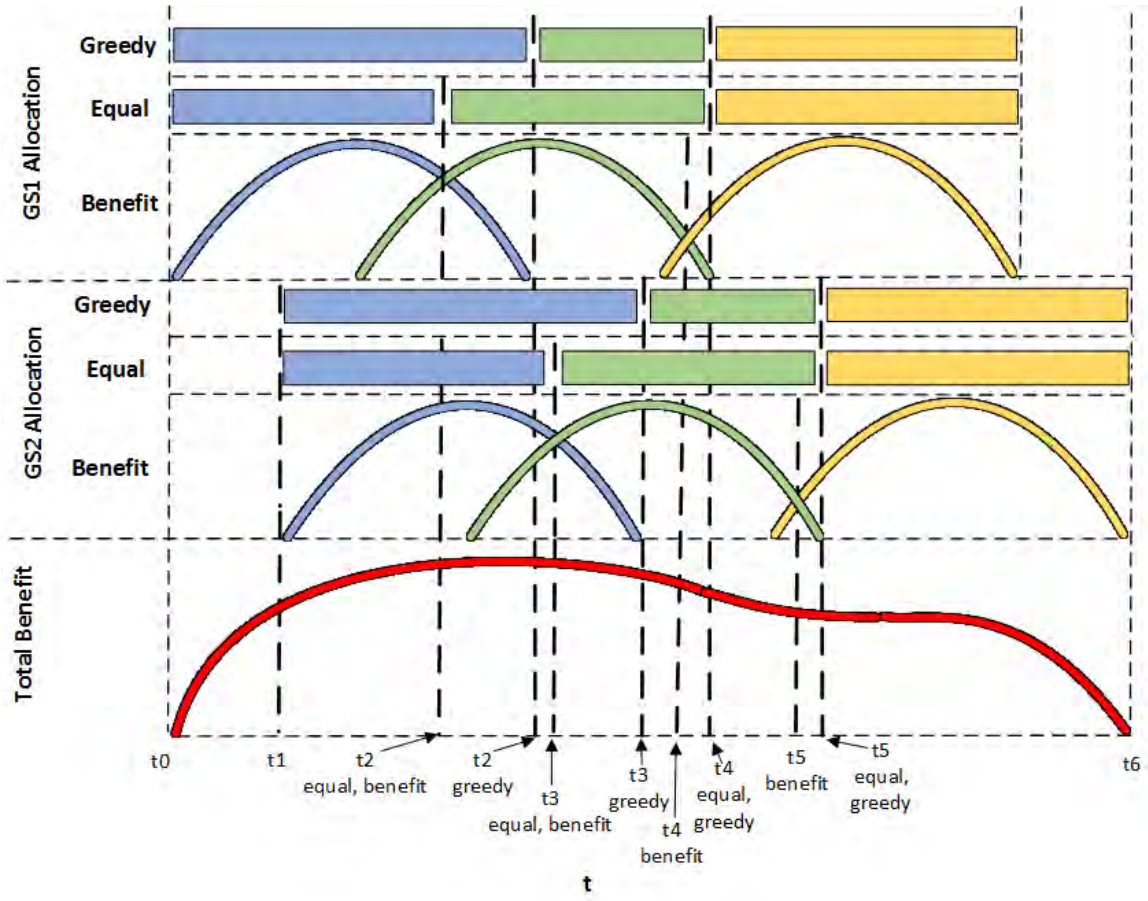


Figure 16. Resource allocation curves for three satellites, two ground stations.

The equal time distribution approach now spends double the total resource time tracking the end of the green satellite pass at t_4 and t_5 instead of transitioning to tracking the yellow satellite. Though a better use of resources than the greedy method, small inefficiencies in a strict equal time-distribution approach also compound with the addition of more stations.

The total benefit value curve of Figure 16 is the summation of individual benefit values for the three satellites at two ground stations. The stations provide the most service to their users when the blue and green satellites are simultaneously in view of both sites. Fundamentally, there are four benefit value curves (two for each station) being summed simultaneously. From a mission planning point of view, this window would be most critical for ensuring uptime. As the blue and green satellites depart and the yellow satellite

approaches, the total benefit is predictably reduced by about half as only two individual benefit value curves (yellow) are summed together.

An algorithm built to maximize the total benefit of these missions would presumably seek to follow the Total Benefit curve by whatever means possible. Real-world considerations such as antenna rotor speeds, precision, and position limits, as well as orbital mechanics, create scenarios whose complexity quickly outgrows manual or heuristic solutions. The approach used to model and generate initial solutions to these problems is described in the subsequent chapters.

E. SCHEDULING CHALLENGES

The simplified problem described in Section D involving multiple stations serving multiple satellites, will become increasingly common as more cost-constrained constellations are fielded. Typically, there is little overlap between ground stations servicing LEO satellites due to the cost of fielding high-performance stations. Many of the networks described in Section C purposefully maximize their geographic distribution to maintain persistent access to their constellations, inherently reducing overlapping contacts. However, as the number of satellites increases, so too will the number of small, low-cost ground stations with reduced capabilities. The same motivations that have made small satellites attractive apply to cost-constrained ground stations. Rather than invest several million dollars in a single high-performance station, developers may opt to invest a fraction of this budget into several reduced-performance stations that provide shortened contact times due to less available link margin. However, the total capacity of such a low-cost network consisting of several stations can provide the same, if not more, total access time than the single high-performance station at a fraction of the price. To accommodate this paradigm, there needs to be an increased reliance on automation for pass planning and scheduling, as well as for the C2 software that flies the satellites.

The development and operation of the MC3 network has enabled first-hand experience with problems in operating low-cost, limited-performance ground stations for an increasing population of small satellites built and operated by a diverse user base.

Anticipating this need, the MC3 network is being upgraded with the SmallSat Agile Transmit Receive Network (SATRN) software, which allows users to connect to the MC3 network using a virtual private network (VPN) and to schedule resources on a central server for their spacecraft contacts (refer to Figure 17). When the contact window opens, SATRN creates a connection from the user's terminal to the remote ground station and allows bidirectional data to be freely passed from the user to the spacecraft and back. This is accomplished by configuring and driving antenna positioners, software-defined radios (SDRs), and any additional required hardware as defined in a database entry corresponding to the particular spacecraft being serviced. The three primary components to the SATRN system are the Client (located at the user location or MOC), centralized Server, and Ground Site (located at the remote ground station). To implement the scheduling optimization algorithms developed as part of this research, all three SATRN components must be utilized.

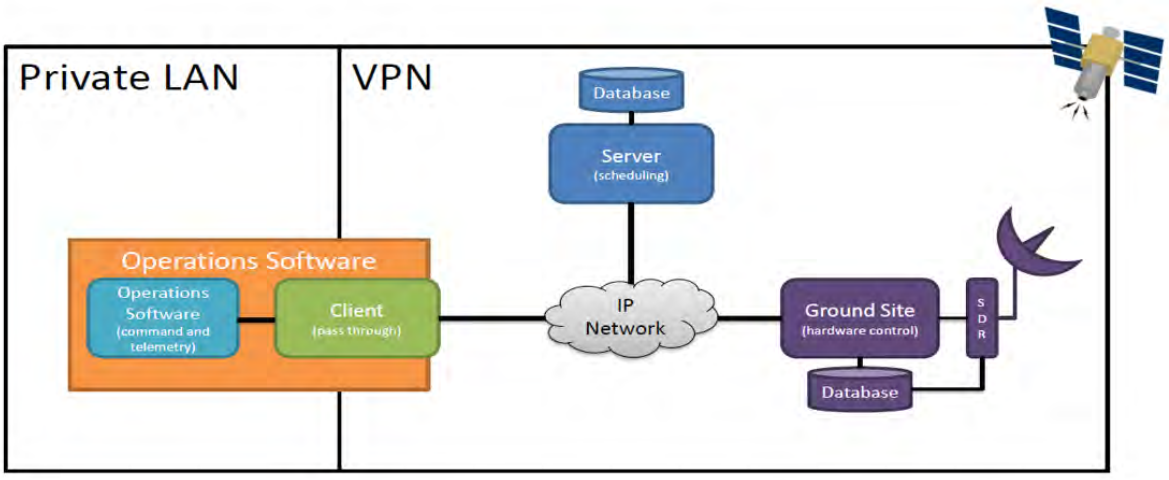


Figure 17. Simplified network diagram for SATRN connections on MC3.

Scheduling time on the MC3 network currently follows a traditional CONOP: users propagate a list of available contact times for ground stations compatible with their satellite(s) and select the opportunities that meet mission requirements. When concerning

LEO satellites, the contact windows are typically generated for horizon-to-horizon passes, resulting in contact durations that span several minutes.

In its initial releases, SATRN implemented a limited deconfliction algorithm, which allows system administrators to assign a priority value to a mission. If users select contact opportunities at a particular ground site, they may be pre-empted later by another higher priority user. As a result, the entire contact window is removed for the lower priority user, and they are forced to reschedule the contact(s). If two equally prioritized users are vying for overlapping windows, a first-come-first-served approach only allows one user at a time. The ground station resource is therefore allocated for the AOS-LOS window with other contact events only available after LOS. The CONOP most closely resembles the *exclusive* resource allocation method described in the previous section and is not globally optimal. The approach may be acceptable when operating several satellites on a distributed ground station network; however, the envisioned satellite population of a network such as MC3 numbers in the hundreds, or possibly, thousands. An automated optimization algorithm is therefore required to maximize usage and capacity of the network. Other ground station networks seeking to serve a growing community of small satellite users will be faced with a similar challenge.

Optimization of network capacity will be enabled by removing the typical horizon-to-horizon schedule window blocking constraint for contacting a satellite passing over a ground station. As further described in Chapter II, a small satellite's link margins typically preclude two-way communications at low elevation angles when slant-range distances are great, due to limited transmission power. A small satellite's downlink is typically constrained by finite onboard power capacity as well as radio licensing limitations. Consequently, true two-way links are possible for much shorter durations than are usually allocated. These durations are typically on the order of a few minutes as the spacecraft is directly over the ground station and slant range distances are at a minimum. Such operational details, if captured in a model and optimized, can greatly increase network capacity as the population of satellites serviced grows beyond the capabilities of first-order heuristics, such as mission prioritization.

F. DISSERTATION OBJECTIVES AND SCOPE

This dissertation is motivated by the ambitious goals of small satellite developers to field large quantities of satellites, numbering in the hundreds or thousands, and the associated operational constraints on ground-based networks. The primary objective of the dissertation is to apply techniques in optimal control to maximize the capacity and benefit of line-of-sight communications between satellites and their ground stations.

The objective is addressed by formulating a dynamic optimization and control model and associated cost/benefit function applicable to slewing ground-based antennas, enabling communications with a diverse population of satellites in various Earth orbits. The core assumption is that a communications antenna can only service one satellite at a time, as is the case with the majority of modern systems. Though ground stations may employ multiple antennas, each antenna is constrained to instantaneously communicate with a single satellite. The envisioned scenario is that hundreds or thousands of these diverse satellites are perpetually serviced by tens of geographically distributed stations. This is realistic, given that it is easier and cheaper to field low cost ground terminals than to expand satellite-to-satellite communications infrastructure like NASA's TDRS. It should be noted that several small satellite missions have recently leveraged (or will soon leverage) satellite-to-satellite communications through existing constellations such as Iridium, GlobalStar, OrbComm, and INMARSAT [45], [46]. However, these are mostly technology demonstrations and the feasibility of utilizing such an architecture is excluded from this research at this time to focus on the space-ground component.

For efficiency, ground operator activities must be abstracted away from anything that requires a "human-in-the-loop" directly driving the system; an intricate, time-consuming, and error-prone proposition when planning and deconflictioning tasks. Rather, operators should input parameters into a model for each satellite and station, accurately reflecting reality, and allowing the system to adjust its communications schedule automatically. This provides a "human-on-the-loop" architecture, where a small team of operators is responsible for guiding, and not directly driving, the complex ground segment tasks required for these constellations.

Guiding many satellites to communicate with a limited number of ground stations can be posed as a resource-constrained schedule optimization problem [24]. This research utilizes a dynamic optimization methodology, enabling the creation of a model of arbitrary complexity for the ground stations and satellites in orbit to maximize/minimize an associated objective function. By accurately modeling the system dynamics and defining operational boundary conditions, a globally optimal and feasible solution can be generated in a single try, satisfying the parameters of the objective function. Traditional methods employ several iterations to check for feasibility and require heuristics/simplifications to converge on a solution.

This dissertation describes research that contributes to the long-term goal of autonomously operating enormous constellations and adopts a proof-of-concept approach. Simulations to validate the algorithms presented in this research addressed up to ten satellites with a single ground station and up to three satellites with two ground stations. The algorithms were then field-tested with up to three orbiting CubeSats and two MC3 ground stations. By demonstrating these simple cases, the algorithms can later be scaled up to arbitrarily large constellations.

Contributions unique to this research are as follows:

- The formulation of an optimal control problem with equations that address space/ground communications [47]–[49]. This well-studied problem [50]–[53] has been solved with many diverse graph theoretic techniques, but not with a dynamic optimization formulation. The formulation differs from other implementations due to its use of Gaussian target models, benefit value functions (BVFs), and other parameters used in satellite mission operations. The formulation is applicable to an arbitrary quantity of satellites and ground stations.

- The software implementation of the formulated problem in a dynamic optimization solver to generate antenna slewing trajectories for targeting orbiting satellites from their respective ground stations. This novel approach

generates a feasible, optimal solution deterministically generated on the first try [48], [54].

- The experimental, real-world demonstration of generated antenna trajectory solutions using orbiting satellites and their supporting ground stations. This research performed demonstrations using up to three PROPCUBE CubeSats [55] and the MC3 ground station network [38]–[43].

G. DISSERTATION OUTLINE

The remainder of this dissertation is outlined as follows. Chapter II introduces a general modeling approach and the mathematical framework required for its application to space and ground systems. Chapter III presents the optimal control problem formulation and validation results for a simple scenario. Scenarios of increasing complexity are covered in Chapter IV to help validate the implementation of the problem formulation discussed in Chapter III. Chapter V applies the optimization model to real-world scenarios and shares results obtained from testing with real CubeSats operated by the MC3 network. Finally, Chapter VI presents conclusions as well as insights for future work.

II. GROUND-BASED SATELLITE TRACKING

This chapter begins with a literature review of the space-ground communications optimization problem and covers several relevant approaches taken to optimize mission operations. The broad field of optimizing space-ground communications has been well studied in recent years, as the population of spacecraft and their user base have increased [51], [56], [57]. Many techniques focus on graph-theoretic algorithms, though brute-force human-in-the-loop scheduling is also leveraged in some cases. A review of available literature suggests that optimal control methodologies have not yet been applied to this problem, and their effectiveness for solving this class of problem is unknown. Prior proof-of-concept research performed at NPS shows that it is possible to apply optimal control theory to the targeting of objects in a well-defined model [48], [49], [54].

The chapter continues with a description of the general modeling framework and mathematical definitions used for ground-based satellite tracking. These equations cover such aspects of the problem as orbital mechanics, reference frames, ground-based observations, and radio signal link budgets. Also covered is the prior work done to model targets by use of the Gaussian function and its application to capturing *benefit* as a quantifiable figure of merit that can be maximized. Chapter III then applies this modeling framework into creating and validating an optimal control problem formulation.

A. LITERATURE REVIEW

Satellites are often expensive resources, and maximizing their use is a priority for mission operators and managers. Typical approaches to usage optimization has involved event deconfliction and task scheduling [58], [59], techniques which address an *oversubscription scheduling problem.* Though some processes can be automated, they are largely overseen by human schedulers who arbitrate complex requests with various organizations to ensure that all conflicts are resolved with enough time for the users to prepare for their schedule slot (i.e., 24–48 hours). There are many quantitative factors to consider, such as the orbital mechanics that enable opportunities to communicate with a ground station network, link requirements, and efficiency of the communication for both

spacecraft and ground station. In addition, there exists a qualitative arbitration process that is generally difficult to quantify and model as it involves potential sensitivities such as customer priority, mission and security classification, experiment timeliness, funding, and many others.

An extreme example of human-in-the-loop task arbitration and deconfliction concerning mission operations occurs in human spaceflight, particularly in the context of International Space Station (ISS) operations [60]. Long-term missions pose significant challenges as hundreds of onboard experiments must be conducted and monitored by astronauts who must also avoid burnout, exercise regularly, eat, and rest. To coordinate the activities of just a few humans in orbit requires large teams on the ground to support multi-year, 24/7 operations [61]. Planning for a six-month expedition typically begins six months before launch and is divided into long-term and short-term timeframes. Long-term schedulers (operations planners, or OPSPLANs) are responsible for activities that are more than three weeks away and mostly deal with negotiating crew time and constraint management, including spacecraft docking/undocking and TDRS tasking. They also coordinate with mission stakeholders, multiple NASA centers, and foreign space agencies. They continually refine the entire schedule from before launch up to five days before task execution [62]. Real-time planning engineers (RPEs) sit on-console, making decisions that range from immediate up to five days away. The decisions being made on-console affect short-term plans, which are continually refined by RPEs and OPSPLANs.

Several tools have been developed to aid mission planners and operators with this heavily human-in-the-loop process [25]. One such tool is the On-board Short Term Plan Viewer (OSTPV) that offers a Gantt-chart-like interface for flight controllers and ISS crew to simultaneously reference. However, inputs into OSTPV may sometimes require human intervention, as interfaces between several schedule databases are not seamless. The inclusion of a new tool called the Operations Planning Timeline Integration System (OPTimIS) [63] provides some ability for dynamic replanning and increased efficiency for ISS planning efforts. As the complexity of human spaceflight missions increases, so too will the requirement to automate the schedule optimization process for human activities.

Though unpiloted spacecraft do not have the same complexities encountered in human spaceflight, they require significant effort and cost to maintain a persistent operational capability. The AFSCN operates more than 100 defense-oriented satellites in a coordinated effort that involves numerous civilian and military organizations. Sixteen antennas located at eight sites positioned around the globe [43] facilitate 24/7 space-ground communications. Customer organizations submit schedule requests based on a specific time period during which their satellite is in view of a ground station. Over 500 such task requests are submitted for a typical day, and it is impossible to accommodate all of them: approximately 20% of all requests conflict. For the tasks that cannot be automatically deconflicted, schedulers contact the requesting organizations and notify them of the conflict, along with potential alternatives. Outright rejection of a task is unacceptable and generally avoided [51], [64].

Ongoing research into AFSCN schedule optimization has involved mixed-integer programming (MIP) and insertion heuristics that schedule 91–95% of all task requests, and a genetic algorithm called *Genitor* that schedules approximately 96% of requests [51]. The research had also adapted greedy heuristics; however, as explained in [51], they are mostly applicable to a single station and require significant simplifications when applied to the entire network. The authors encountered problems with scaling larger, more complex scheduling tasks utilizing multiple resources on the network.

Other genetic algorithms have been developed to solve the satellite-scheduling problem. One such example sought to optimize operation scheduling of the Indian Space and Research Organization (ISRO) Telemetry Tracking and Command (ISTRAC) and its fleet of LEO satellites [65]. Other forms of genetic algorithms solve problems in resource-constrained scheduling [53], [56], which not only include the operational scheduling of satellite contacts, but also the outright tasking of Earth-observing satellites (EOS) [57], [66]. The similarities between EOS tasking and satellite mission operations planning are based on mapping tasks, such as observations, communications, downlinks, control maneuvers, etc., to resources, such as sensor satellites, relay satellites, ground stations, etc. [67].

Given the similarities between EOS and mission operations planning, we can expand the literature review to involve EOS applications. The goal for most EOS applications is to take the maximum number of high-priority Earth observations in a given time interval. EOS problems have often been generalized and formulated with common structures, such as a knapsack problem, a packing problem, a single-machine scheduling problem, or a network flow problem [59]. Other approaches commonly used are genetic algorithms and greedy algorithms based on prioritization, constraint programming, dynamic programming, and heuristics [68].

1. Graph-Theoretic Problem Formulation

In general, tasking an EOS for image collection planning and scheduling can be framed as a constrained hybrid dynamic optimization problem where the variables are discrete, continuous, dynamic, and constrained. Typical solution strategies involve forming and solving a graph problem. Thus, even a simplified version of this problem cannot be solved in nondeterministic polynomial time, *i.e.*, the problem is NP-hard. In other words, it falls into a class of problems that are too complex to be solved in a realistic amount of time. Though a full solution cannot be computed, the full EOS problem can be identified and has been characterized [69]. Most attempts to solve this NP-hard problem will approach it with a combination of graph theory and heuristics to create the simplifications necessary to converge on a solution. This leads to a degree of conservatism that is contrary to the requirements of large-scale systems.

The traditional means of solving this class of problems broadly apply across the various graph-theoretical algorithms described above, and are as follows [54]:

- Step 1: Reduce the task requisition cardinality by applying heuristics where possible. This produces a subgraph with reduced complexity.
- Step 2: Choose a simple closed-form solution to reduce the hybrid-dynamic problem of Step 1 to a non-dynamic but time-dependent graph problem.
- Step 3: The resulting problem is separated into separate problems of planning and scheduling. Each problem is then solved using heuristics and

graph-theoretic algorithms that generate a solution to the scheduling subproblem based on a given payoff function (e.g., profit).

- Step 4: A high-fidelity propagator simulates the scheduling sequence from Step 3. If the test fails, the entire process or parts of the process are repeated until a desirable solution is obtained for flight operations.

Such optimization tasks often take the form of orienteering problems (OP) [70], which are generalizations of the traveling salesman problem (TSP) [71]. The problems typically involve a particular starting location and require moving to an endpoint through multiple waypoints. Moving to each target along the way imparts a cost, and reaching a waypoint generates a profit or benefit. The problem space can be shown graphically using nodes interconnected with arcs. The directionality of the arcs is problem specific and is captured in a *directed graph*, or *digraph*, as shown in Figure 18.

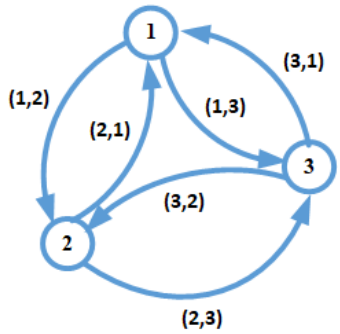


Figure 18. Example of a directed graph.

The overall cost of traveling between all nodes in the graph can be minimized (or profit maximized) through an *objective function*. This function is mathematically formulated to meet the goals or the objectives of the problem being solved. Given the constraints of the model (*i.e.,* arc directionality and associated costs); the objective function is made up of the sum of the individual benefits/costs from all of the nodes visited. Determining whether a node is visited depends on a decision variable Δ_{ij} . If the value of

each node is the same, the objective function is a summation of all decision variables to maximize all nodes visited:

$$\max_{\Delta_{ij}} \quad \sum_{j \in N} \sum_{i \in N} \Delta_{ij} \tag{1}$$

subject to revisit constraints and where N is the number of nodes [69]. Additional problem constraints and associated costs expand this formulation. The most challenging aspect of formulating this problem is translating mission requirements and any underlying physics into a model that fits this graph-theoretic template to utilize the associated techniques for solving the problem. Since the orienteering problem is, by definition, NP-hard [72], its solution techniques must include simplifications and heuristics for practical applications.

An alternative approach for solving the above scheduling and planning problem comes from the application of well-established techniques in optimal control, producing a different formulation of the same problem. Considering the system as a single integrated dynamic optimization problem eliminates many of the heuristics, simplification steps, and iterative loops typically required [54]. To accomplish this, the non-smooth graph problem described above is represented as smooth and time-continuous. The resulting solution has the advantage of static and dynamic constraint satisfaction at its very first iteration. The techniques used to solve OP, TSP, and EOS problems apply equally to those of ground station communications optimization. The following sections describe the fundamentals required to formulate the space-ground communications optimization problem.

2. Optimal Control Fundamentals

Physical technological processes are controllable once the underlying system dynamics have been characterized. This characterization process involves a mathematical model of the system to be controlled, as well as the resulting dynamics associated with the sensing and additional actuation in the controller. Optimal control seeks to not only induce control inputs to make a dynamic system reach its intended objective, but to do so optimally. The definition of optimality varies with the application, though each case seeks to minimize or maximize a particular process or characteristic. For example, a process may

require minimum time, minimum energy, minimum penalty, maximum profit, or maximum benefit, and control inputs are therefore selected to meet these objectives. Optimal control problems require characterization of the physical system to be controlled, a mathematically accurate description of the objective function, and any constraints in the system. Formulating the right problem becomes the most important facet of an optimal control problem [73].

Modeling the dynamics of the system utilizes the state-space form of

$$\dot{x}(t) = f(x(t), u(t)) \tag{2}$$

where x represents the state variables and u represents the control variables as a function of time. Boundary conditions, such as start and stop time, and starting and endpoint conditions are also part of the problem formulation and are represented by the *endpoint manifold e*. A function that seeks to minimize or maximize an objective is called the *Bolza cost functional J.* Also referred to as an *objective function*, or *cost function*, J is a function of functions (called a *functional*) composed of states, controls, and time [74]. The general form of J is

$$J[x(\cdot), \mathrm{u}(\cdot), t_f] = E(x(t_f)) + \int F(x(t), u(t))dt \tag{3}$$

where E is the endpoint (Mayer) cost at the final elapsed time and F is the running (Lagrange) cost of the system. Both $x(\cdot)$ and $u(\cdot)$ represent functionals (functions of functions), where $x(t), u(t)$ are the values of the function evaluated at the point t [73]. The full generic optimal control problem formulation is presented below:

$$\begin{cases} \text{Minimize} \quad J[x(\cdot), \mathrm{u}(\cdot), t_f] = E(x(t_f)) + \int F(x(t), u(t))dt \\ \text{subject to} \qquad \dot{x}(t) = f(x(t), \mathrm{u(t)}) \\ \qquad\qquad\qquad x(t_o) = x^0 \\ \qquad\qquad\qquad t_o = t^0 \\ \qquad\qquad\qquad t_f = t^f \\ \qquad\qquad\qquad e(x(t_f)) = 0 \end{cases} \tag{4}$$

Choosing which specific attributes fit into this generic formulation is the primary goal of the problem formulation process. Once a problem has been developed, several approaches can be used to solve it. This research effort primarily used the DIDO optimal control toolbox [73] due to institutional knowledge, though other solvers could have also sufficed. The specific methods and algorithms used within DIDO are not discussed in this dissertation, as they are not within the scope of the exposition.

DIDO uses pseudospectral optimal control theory to generate extremal solutions for properly formulated optimal control problems [75]. No knowledge of pseudospectral methods is necessary to use DIDO. The tool offers a reduced learning curve and powerful insight into the system beyond states and controls. The values for Hamiltonians, costates, and covectors are presented as user feedback to illustrate whether or not the solution is feasible as coded. Several examples of DIDO's use in the aerospace industry can be found in references [76]–[79]. After formulating the problem, it must then be evaluated for completeness. Theorem 2.1 (Pontryagin's Principle) in [73] states that:

"Given an optimal solution [to the problem], there exists an absolutely continuous covector function $\lambda(\cdot)$ and a covector υ that satisfy:

- the three Hamiltonian conditions:
    1. Hamiltonian minimization condition,
    2. Hamiltonian value condition,
    3. Hamiltonian evolution equation
- the adjoint equations, and
- the transversality condition."

Furthermore, [73] presents a useful pneumonic, HAMVET, to summarize the steps to develop the necessary conditions for optimality. The steps are as follows:

a)	Construct the **H**amiltonian	H
b)	Develop the **A**djoint equations	A
c)	**M**inimize the Hamiltonian	M
d)	Evaluate the Hamiltonian **V**alue condition	V
e)	Integrate the Hamiltonian **E**volution equation	E
f)	Formulate the **T**ransversality conditions	T

Chapter III presents problem formulations of increasing complexity and follows the HAMVET process to validate their use in an optimal control solution. Adherence to Pontryagin's Principle is a necessary but not sufficient condition for optimality. Additional validation and verification (V&V) allows the designer insight into problems that are too complex to be solved by hand or analytically.

3. Smooth Target Model Development

Developing models for optimal control not only requires the formulation of the system dynamics but also the path constraints and respective cost functions. The application this dissertation addresses necessitates a cost function that encompasses the targeting of satellites by ground-based antennas and associated weighing functions based on mission objectives. The optimizer used in this research requires smooth contours and continuous functions. Therefore, substantial emphasis has been placed on creating smoothly defined target fields that can be incorporated into the dynamic optimization problem formulation.

An example of previous work done using optimal control for performing targeting tasks has taken advantage of the p-norm to create simple smooth geometric shapes [80]. Different shapes are useful for dictating the path a solution may take from start to finish (represented as obstacles or targets). Multiple p-norms can be used to create multiple targets or more complex shapes if desired. Equation (5) shows that the general form of the center of the two-dimensional shape is defined by (x_c, y_c), the width is defined by a and b, and the shape itself is defined by the value of p.

$$h(x(t), \mathrm{y(t)}) = \left\| \left(\frac{x(t) - x_c}{a} \right)^p \right\| + \left\| \left(\frac{\mathrm{y}(t) - y_c}{b} \right)^p \right\| - \left\| c^p \right\| \tag{5}$$

Figure 19 illustrates three p-norms with varying values of p and a, b, c = 1 though other values are possible. The value of p can go to ∞, though p = 100 can be used for practical considerations.

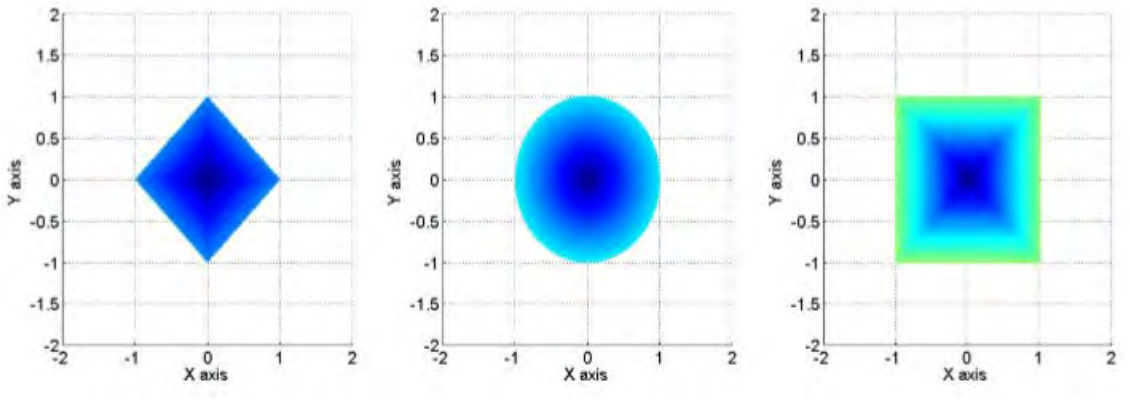


Figure 19. Unit p-norms for p = 1, 2, and 100—shapes are continuous and smooth. Source: [80].

Another useful technique for defining targets is the exponential (Gaussian) function, as explored in previous research [49]. The Gaussian center is defined by (x_n, y_n), and its width in each axis can be tuned with the parameters σ_x and σ_y. Equation (6) shows the general form of the function:

$$h(x(t), \mathrm{y(t)}) = e^{-\left(\frac{(x-x_n)^2}{2\sigma_x^2}+\frac{(\mathrm{y}-y_n)^2}{2\sigma_y^2}\right)} \tag{6}$$

The Gaussian produces a continuous, differentiable curve to describe a target. An added benefit to this function is that it approaches, but never reaches zero, allowing the optimizer to search for the peak of the curve, perpendicular to the contour, from anywhere in the problem space. An example Gaussian is shown in Figure 20.

The Gaussian functions can be scaled easily with a weighing factor to change the peak amplitude. This is beneficial when considering multiple targets of differing values. The weighing factor is referred to as a value function v and associates the value for the set of tasks to be performed [48]. Equation (7) represents a Gaussian scaled with the value function. The combined quantity is referred to as a benefit value function (BVF) and is a fundamental concept for the satellite tracking problem formulation.

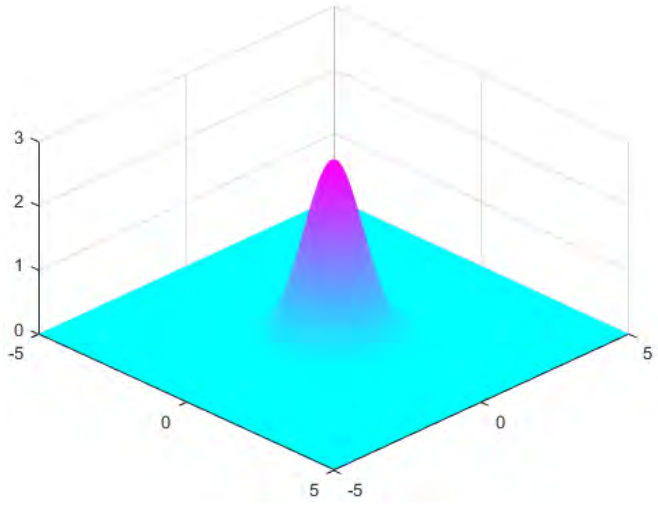


Figure 20. Example of Gaussian function centered at the origin. Source: [49].

$$BVF = ve^{-\left(\frac{(x-x_n)^2}{2\sigma_x{}^2}+\frac{(y-y_n)^2}{2\sigma_y{}^2}\right)} \tag{7}$$

The BVF provides a quantitative method for tracking qualitative attributes that comprise *benefit*. Due to Pontryagin's Principle and its resistance to the curse of dimensionality, the model used to create the BVF can be arbitrarily complex and can therefore be tuned to match realistic parameters and behaviors [49]. As a result, the Gaussian in Figure 20 is an adequate representation of a satellite target given its smooth contours and scalability. For example, altering its position over time can represent a moving satellite when observed from a ground station. Dynamically scaling the peak, as discussed in Chapter I, can correspond to factors such as radio link quality, mission priority, asset availability, onboard data quality, and many others. The addition of simultaneous moving targets creates a single smooth surface with multiple peaks from which a global maximum is computed. The cost function can be tuned to either seek or avoid these peaks across the target field.

The following section describes the mathematical fundamentals required to provide the necessary fidelity for tracking the Gaussian target peaks in the optimization problem.

The model utilizes concepts pertaining to orbital mechanics, link margin calculations, and classical dynamics.

B. MATHEMATICAL DEFINITIONS

The following section describes in detail the fundamentals of the mathematical and physical relationships that must be considered for solving the ground-based satellite tracking problem. A simplified two-body mathematical orbital analysis is first described with respect to a terrestrial ground station, such that an analytic solution can be generated for the initial optimal control problem formulation that this dissertation addresses. The complexities associated with analytic solutions will then require numerical approximations for implementation in real-world examples.

1. Classical Orbital Elements

Kepler's three laws describe the motions of two point masses under mutual gravitational attraction. When applied to satellite and planetary orbits, these laws are summarized as follows [81]:

Kepler's First Law: If two objects in space interact gravitationally, their paths will describe an orbit that is a conic section with the center of mass at one focus. If the bodies are permanently associated, the orbit shape is an ellipse; otherwise, it is a parabola.

Kepler's Second Law: If two objects in space interact gravitationally, a line joining one object to the other sweeps out equal areas in equal time.

Kepler's Third Law: If two objects revolve around each other due to mutual gravitational attraction, the square of the orbital period T is directly proportional to the cube of the semi-major axis a of the orbital ellipse. When using Newton's Law of gravitation, G is the gravitational constant, M is the mass of the larger object, and m is the mass of the smaller object, with $m << M$. The satellite's mass m is negligible with respect to the Earth's mass M, and can be ignored. Therefore, Kepler's Third Law can be written as follows:

$$\frac{T^2}{a^3} = \frac{4\pi^2}{GM} \tag{8}$$

Since GM is constant, it is assigned the value μ and the equation is written as

$$\frac{T^2}{a^3} = \frac{4\pi^2}{\mu} \tag{9}$$

Applying the three laws to solve a two-body motion problem allows the calculation of orbital position as a function of time. This is particularly useful for *propagation*, where the position of the orbiting satellite is predicted with respect to the Earth using the last known position and velocity measurements. The apparent simplicity of solving this problem is misleading: predicting a satellite's position exactly precludes an algebraic solution requiring numerical approximations of high precision. Additionally, many real-world disturbances must also be considered, primarily Earth-spheroid oblateness, atmospheric drag, gravitational pull from other celestial objects, solar wind pressure, and uneven distributions of mass within the Earth.

The simplest two-body orbit with an analytical, closed-form solution consists of the circle, which is a special case of the ellipse described in Kepler's First Law. The circular orbit, illustrated in Figure 21, has uniform angular velocity, allowing a constant relationship between the angular distance traveled and units of time. The equation for a circular orbit is as follows [82]:

$$\theta = \frac{\sqrt{\mu}}{a^{3/2}} t \tag{10}$$

where the semi-major axis a is always equal to the orbital radius r, μ is the planet's gravitational constant, t is time elapsed, and θ is the angle swept as a function of time, called t*rue anomaly*. The position vector for a circular orbit can therefore be written as the following:

$$\mathbf{r}_{2D} = \begin{bmatrix} r\cos\theta \\ r\sin\theta \\ 0 \end{bmatrix} \tag{11}$$

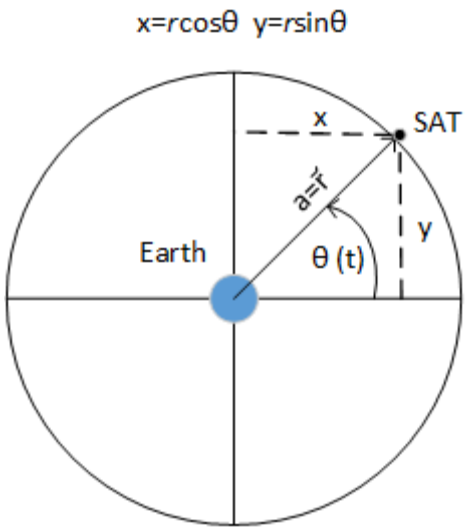


Figure 21. Circular orbit in two dimensions.

The more general case for a simple two-dimensional orbit involves the use of an ellipse, presented in Figure 22. The semi-major axis of the ellipse *a* can be defined by the relationship $2a = r_p + r_a$ where r_a is the radius of apogee and r_p is the radius of perigee. The distance from the center of the Earth to the satellite is radius *r*. The altitude of the satellite is therefore the radius of the Earth R_e subtracted from *r*. The *eccentricity*, *e*, of the orbital conic section defines the shape of the path. An ellipse (closed orbit) has $0 < e < 1$, with a circular orbit being a special case at $e = 0$. A hyperbolic orbit (escape trajectory) has $e > 1$, with a parabolic orbit being a special case at $e = 1$.

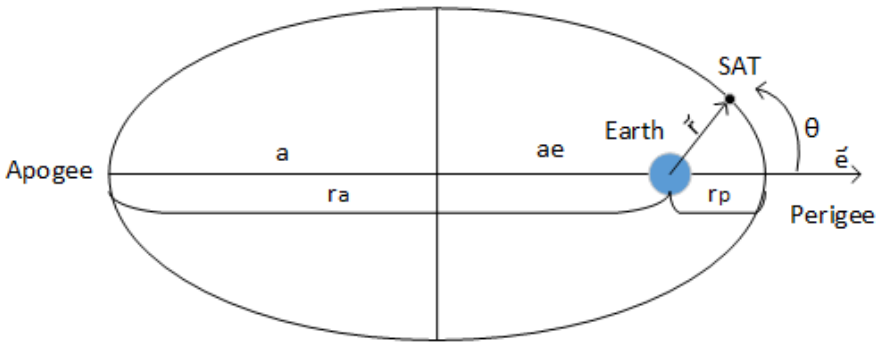


Figure 22. Elliptical orbit in two dimensions.

Obtaining satellite position as a function of time requires the application of the above definitions in three dimensions. This necessitates the use of additional orbital description parameters, known as *Classical Orbital Elements* (COEs) or *Keplerian elements* [81]. Though the focus of this work is on Earth-orbiting satellites, these definitions can apply to any two-body trajectories. Figure 23 shows a three-dimensional representation of an Earth orbit with the COEs.

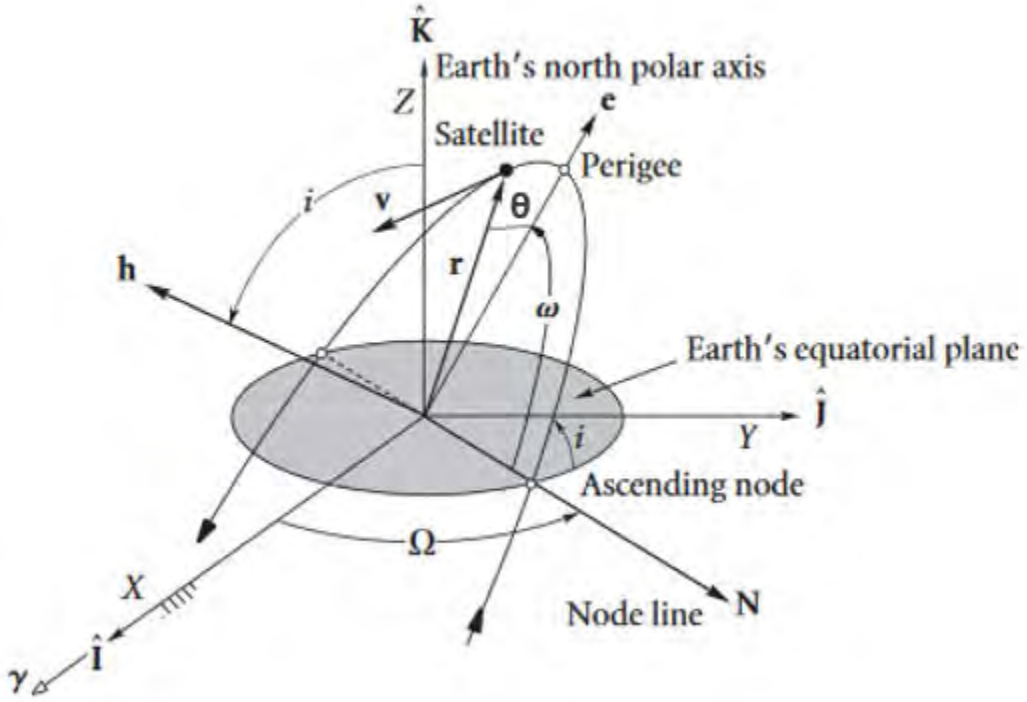


Figure 23. Three-dimensional representation of Earth orbit. Source: [83].

Figure 23 illustrates the required parameters for fully describing a satellite orbiting the Earth. The initial coordinate system is the Earth-Centered Inertial (ECI) frame IJK_{ECI} which remains fixed with the stars; specifically the $\hat{I}$ axis always points at the Point of Aries γ and the $\hat{K}$ axis is normal to Earth's equatorial plane. The Earth's rotation can be characterized by the Earth-Centered Earth-Fixed (ECEF) frame IJK_{ECEF}, which rotates about the $\hat{K}$ axis once every sidereal day (23.93 hours per day or 15.04 degrees per

second). At this constant rotation rate the time elapsed since $\hat{I}_{ECEF}$ crossed the $\hat{I}_{ECI}$ axis is measured as apparent sidereal time (AST), denoted by the displacement angle $\theta_{AST.}$ In Figure 23, ECI and ECEF frames are collocated, indicating 0 hours sidereal time or $\theta_{AST} = 0$.

The orbital plane in Figure 23 intersects Earth's equatorial plane at an angle equal to the *inclination*, *i*, of the orbit. Much like the orbit shown in Figure 21, the position of the satellite along the orbital track, or *true anomaly*, is defined as the angle θ, measured with respect to the eccentricity vector *e*, which intersects the orbital perigee and mass center of the Earth. The angle between the eccentricity vector and Earth's equatorial plane is the *argument of perigee*, ω.

Normal to the orbital plane is the angular momentum vector $\vec{h}$, intersecting the center of the Earth and the focus of the orbital ellipse. Lastly, the *right ascension of the ascending node* Ω rotates the orbital plane about the vertical axis referenced from the $\hat{I}_{ECEF}$ axis. These orbital elements allow for the computation of the satellite's instantaneous position and velocity in three-dimensional space by the vectors $\vec{r}$ and $\vec{v}$, respectively.

Converting a two dimensional orbit into three dimensions requires a transformation in the ECI frame. The transformation is accomplished with the following rotations [82]:

$$\begin{aligned} T_{ECI_{2D}->ECI_{3D}} &= \left[R_3(\omega)R_1(i)R_3(\Omega)\right]^T \\ \vec{r}_{ECI} &= T_{ECI_{2D}->ECI_{3D}}\vec{r}_{2D} \end{aligned} \tag{12}$$

and incorporating Equation (11) leads to the expression:

$$\mathbf{r}_{SAT_{ECI}} = \begin{bmatrix} \cos\Omega\cos\omega - \sin\Omega\sin\omega\cos i & -\cos\Omega\sin\omega - \sin\Omega\cos i\cos\omega & \sin\Omega\sin i \\ \sin\Omega\cos\omega + \cos\Omega\cos i\sin\omega & -\sin\Omega\sin\omega + \cos\Omega\cos i\cos\omega & -\cos\Omega\sin i \\ \sin i\sin\omega & \sin i\cos\omega & \cos i \end{bmatrix} \begin{bmatrix} r\cos\theta(t) \\ r\sin\theta(t) \\ 0 \end{bmatrix} \tag{13}$$

The analysis next requires a description of the satellite's position with respect to a rotating frame on the Earth's surface.

2. Satellite Observations from Earth's Surface

To define the satellite position with respect to a terrestrial ground station, the topocentric-horizon coordinate system, shown in Figure 24, is commonly used. Azimuth and elevation angles are used to describe the satellite's position in the sky, measured from local true north and the horizon, respectively. Converting from ECI frame into local azimuth and elevation angles requires several transformations, described below.

The three-dimensional orbit expressed in ECI must next be converted into the ECEF coordinate frame. This is accomplished by an R_3 rotation of the ECI frame by θ_{AST} degrees. The rotation can also be expressed as a function of time if using the Earth's axial rotation rate $\omega_{\oplus}$.

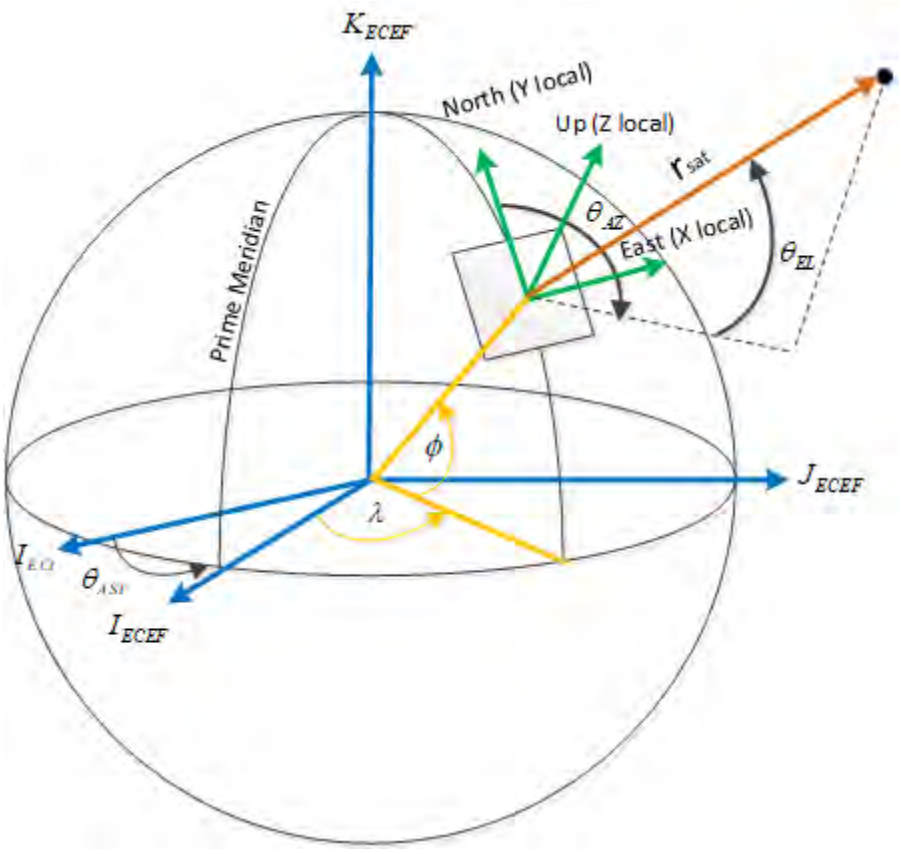


Figure 24. Topocentric-horizon coordinate system.

The conversion from ECI to ECEF can be expressed as follows:

$$\begin{aligned} T_{ECI->ECEF} &= R_3(\theta_{AST}) \\ T_{ECI->ECEF} &= R_3(\omega_{\oplus} t) \\ \vec{r}_{ECEF} &= T_{ECI->ECEF}\vec{r}_{ECI} \end{aligned} \tag{14}$$

The position of the ground station on the surface of the Earth must next be characterized. The angle λ is used for the station's longitude and φ for the latitude. This analysis assumes a spherical Earth, otherwise the geodetic latitude must be used for real-world applications due to Earth's oblate shape. The topocentric coordinate system, with origin at the ground station, has several options for orienting its reference frame. This analysis uses the East North Up (ENU) coordinate system, which points its local X-axis east, Y-axis to true north, and Z-axis zenith.

The conversion between ECEF and ENU requires two rotations, as shown in Figure 25 and Equation(15).

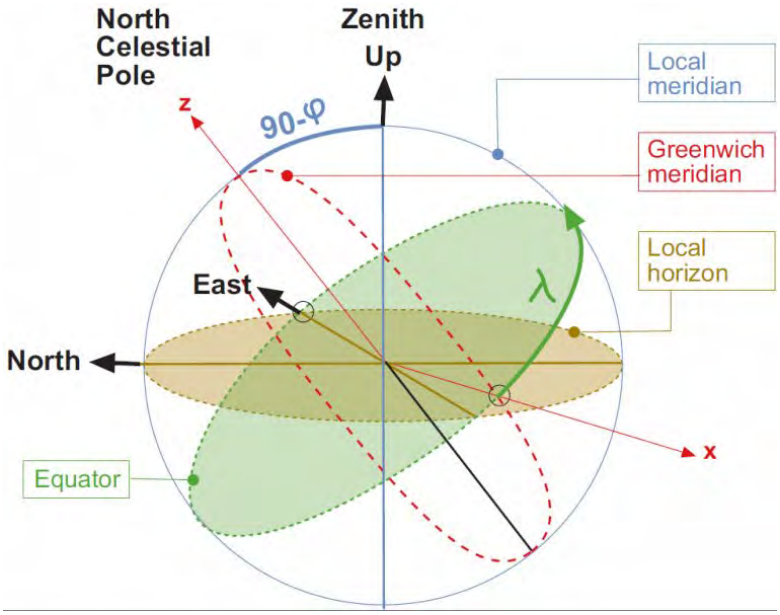


Image retrieved from European Space Agency (ESA)
http://www.navipedia.org/index.php/Transformations_between_ECEF_and_ENU_coordinates

Figure 25. ECEF to ENU transformation.

$$R_1\left(\frac{\pi}{2}-\varphi\right)R_3\left(\frac{\pi}{2}+\lambda\right)=\begin{bmatrix}-\sin\lambda & \cos\lambda & 0\\ -\cos\lambda\sin\phi & -\sin\lambda\sin\phi & \cos\phi\\ \cos\lambda\cos\phi & \sin\lambda\cos\phi & \sin\varphi\end{bmatrix}$$
$$\mathbf{r}_{r_{ENU}}=\begin{bmatrix}-\sin\lambda & \cos\lambda & 0\\ -\cos\lambda\sin\phi & -\sin\lambda\sin\phi & \cos\phi\\ \cos\lambda\cos\phi & \sin\lambda\cos\phi & \sin\varphi\end{bmatrix}\mathbf{r}_{r_{ECEF}} \tag{15}$$

Both satellite and ground station position vectors must be represented in the same ENU coordinate frame to produce position descriptions with respect to a ground station. The conversion requires a two-step transformation; the first is performed as a function of time using Earth's rotation rate and the second as a fixed relationship based on the latitude and longitude of the station. Starting with Equation (13), the satellite vector is transformed to ENU:

$$\mathbf{r}_{r_{SAT_{ECEF}}}=\begin{bmatrix}\cos\omega_{\oplus}t & \sin\omega_{\oplus}t & 0\\ -\sin\omega_{\oplus}t & \cos\omega_{\oplus}t & 0\\ 0 & 0 & 1\end{bmatrix}\mathbf{r}_{r_{SAT_{ECI}}}$$
$$\mathbf{r}_{r_{SAT_{ENU}}}=\begin{bmatrix}-\sin\lambda & \cos\lambda & 0\\ -\cos\lambda\sin\phi & -\sin\lambda\sin\phi & \cos\phi\\ \cos\lambda\cos\phi & \sin\lambda\cos\phi & \sin\varphi\end{bmatrix}\mathbf{r}_{r_{SAT_{ECEF}}} \tag{16}$$

Similarly, the ground station position description must be transformed to ENU from the ECI frame. Starting with a point located on the surface of the Earth at a distance of R_{EARTH} (6378 km for the spherical model) the rotations progress from the latitude and longitude to the ECEF and then ENU frames:

$$
\begin{aligned}
\vec{r}_{SITE_{ECI}} &= \begin{bmatrix} \cos\phi & 0 & -\sin\varphi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\lambda & \sin\lambda & 0 \\ -\sin\lambda & \cos\lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{\oplus} \\ 0 \\ 0 \end{bmatrix} \\
\vec{r}_{SITE_{ECEF}} &= \begin{bmatrix} \cos\omega_{\oplus}t & \sin\omega_{\oplus}t & 0 \\ -\sin\omega_{\oplus}t & \cos\omega_{\oplus}t & 0 \\ 0 & 0 & 1 \end{bmatrix} \vec{r}_{SITE_{ECI}} \\
\vec{r}_{SITE_{ENU}} &= \begin{bmatrix} -\sin\lambda & \cos\lambda & 0 \\ -\cos\lambda\sin\phi & -\sin\lambda\sin\phi & \cos\phi \\ \cos\lambda\cos\phi & \sin\lambda\cos\phi & \sin\varphi \end{bmatrix} \vec{r}_{SITE_{ECEF}}
\end{aligned}
\tag{17}
$$

When both satellite and ground station positions have been converted to ENU, the relative azimuth and elevation angles can be computed (Figure 24). The satellite range unit vector $\vec{\rho}$ is calculated by subtracting the site vector from the satellite vector, and is normalized by its magnitude. The range vector magnitude is also called the *slant range*, which is used extensively by satellite operators and in this research.

Finally, the local azimuth angle is calculated from the arctangent of the east and north components of the range vector. The elevation angle is computed from the arcsin of the zenith (up) vector. The set of equations describing the satellite position in the ENU frame is shown here:

$$
\begin{aligned}
slant \quad range &= \left\| \vec{r}_{SAT_{ENU}} - \vec{r}_{SITE_{ENU}} \right\| \\
\vec{\rho} &= \frac{\vec{r}_{SAT_{ENU}} - \vec{r}_{SITE_{ENU}}}{\left\| \vec{r}_{SAT_{ENU}} - \vec{r}_{SITE_{ENU}} \right\|} \\
\theta_{AZ} &= \arctan\left(\frac{\vec{\rho} \cdot \hat{e}}{\vec{\rho} \cdot \hat{n}} \right) \\
\theta_{EL} &= \arcsin(\vec{\rho} \cdot \hat{u})
\end{aligned}
\tag{18}
$$

Chapter II has so far presented satellite and ground station position descriptions to represent a simplified approximation model for validating the optimization algorithms developed for this research. Though the simplified model lacks the accuracy of modern-day propagators, and makes assumptions regarding time keeping, a spherical Earth, and many other simplifications, the terms generated in this process aid in the derivation and

validation of the optimal control problem formulation presented in subsequent sections. The accuracy of the simplified model is sufficient for this initial validation analysis, though a numerical propagator must be used for real-world implementations where increased accuracy improves system performance.

This dissertation uses the Systems Tool Kit (STK) 10 numerical propagator. STK is a physics-based software package that performs geometric time-dynamic position and attitude calculations for a wide array of assets in ground, sea, air, and space applications. To extract the desired spacecraft and ground station positions for time intervals of interest, both Two Line Elements (TLEs, described in the following section) and geodetic coordinates were input. The built-in Simplified General Perturbations Satellite Orbit Model 4 (SGP4, also described in the next section) propagator was then configured to export spacecraft ephemerides in ECEF and ENU coordinates [48], which were used in the targeting models. Later sections describe implementations of the numerical propagator for this research.

3. Two Line Elements

Classical orbital elements are commonly used in the scientific community; however, various incarnations exist involving small variations in the presentation of information. One such incarnation is the TLE, used by the United States military and distributed publically by the Joint Space Operations Command (JSpOC) [84]. These element sets, or *elsets*, contain the orbital elements necessary for estimating the position and velocity of an object when using a suitable prediction formula. TLEs also contain identifying information, such as satellite catalog numbers and corresponding launch date as well as terms that help better predict the effects of orbital perturbations such as atmospheric drag [85].

TLEs are typically generated by a radar measuring three-dimensional vectors $\vec{r}$ and $\vec{v}$ (Figure 23) referenced to the epoch time based on Universally Coordinated Time (UTC). Given a time, position, and velocity, the rest of the orbital elements can be calculated using Kepler's laws and the associated equations. The resulting elset is fed into a propagator to

predict the position of the satellite at other points in time. A common propagation model is SGP4 [86] that also factors in orbital perturbations, such as Earth-oblateness and atmospheric drag.

No accuracy information is provided with a TLE measurement. Position comparisons between onboard GPS and propagated TLE predictions performed in the scientific community typically generate an error in position knowledge of 1–2 km when using the TLE alone [87]. Furthermore, the perturbation values in a TLE are approximations of complex orbital dynamics, meaning that the further out a TLE is propagated, the greater the position error. The compounding error can be mitigated by more frequent position measurements as well as several other techniques including instantaneous Doppler shift measurements of the satellite's radio signals and onboard GPS measurements, which triangulate the satellite's position to an accuracy of several meters.

After accurately computing satellite and ground station positions as a function of time, several mathematical principles are leveraged to integrate these data into an optimal control problem formulation. Unit circles translated into Cartesian space, Gaussian distributions, and double-integrator dynamics are described next.

4. Target Representation as Polar Plot in Cartesian Space

Representing the satellite target as a Gaussian (Chapter II Section A.3) requires the definition of a Cartesian coordinate to center the peak. This dissertation defines the Gaussian center point location from the perspective of the ground station, with the station at the origin (0, 0) in the topocentric horizon frame described earlier. As ground stations can describe satellite positions as a function of local azimuth and elevation angles, a polar plot is a logical representation [88], [89]. Therefore, any azimuth and elevation pair can be transformed into a Cartesian grid coordinate. For simplicity, the grid is normalized from values of [-1, 1] in the x and y axes such that a unit circle can serve as the basis for the polar plot. What follows describes this mapping.

A unit circle has a radius of one and is centered at the origin (0, 0) in the Cartesian coordinate system. A point traveling along the unit circle can be described by the Cartesian

coordinates $x = cos(\theta)$, $y = sin(\theta)$ where θ is the angle between a line joining the point and the origin and the x-axis, incrementing counter-clockwise (note that θ for this example is different from θ corresponding to True Anomaly, described previously). A variation of the unit circle structure can be used when describing the position of a satellite with respect to a ground station using the azimuth and elevation angles described above. To accurately map these angles into a *(x, y)* coordinate in Cartesian space requires a transformation. Figure 26 shows the azimuth angle transformation.

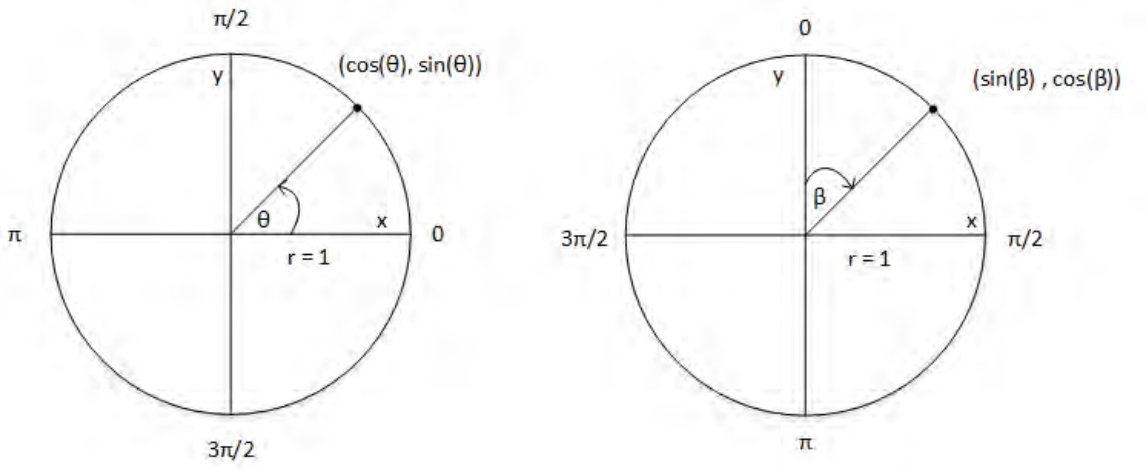


Figure 26. Cartesian space representations of traditional Unit Circle (left) and Compass circle with Azimuth angles (right).

The azimuth angle transformation requires a 90-degree rotation to move the zero-degree reference from the positive-*x* to the positive-*y* axis. Furthermore, azimuth angles increment clockwise, requiring the addition of a negative sign to the y-axis component.

The Cartesian space representation of a satellite's position with respect to a ground station also requires consideration of the elevation angle. The origin is the location of the ground station. Since elevation angles are measured with respect to local horizon and increment toward zenith, a satellite directly over the ground station has an elevation angle of 90 degrees while one at the horizon has an elevation of 0 degrees. Predictably, satellites below the horizon have negative elevation angles. To map an elevation angle to Cartesian

space, the radius of the unit circle is normalized as a function of the elevation angle. Figure 27 illustrates the inclusion of elevation angles in the modified unit circle.

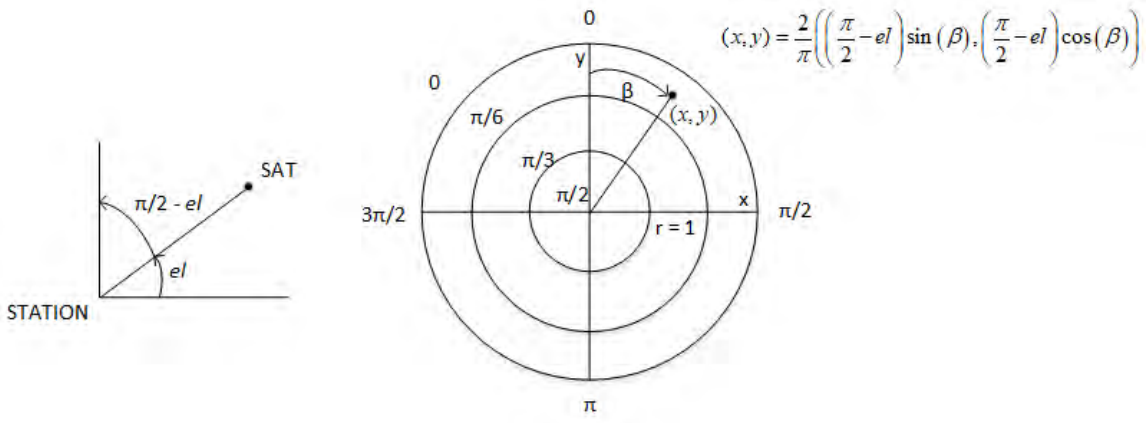


Figure 27. Elevation angles with respect to the horizon (left) and represented in Cartesian space (right).

The final expression converting azimuth and elevation angles into a normalized Cartesian grid is as follows:

$$(x,y)=\frac{2}{\pi}\left(\left(\frac{\pi}{2}-el\right)\sin(\beta),\left(\frac{\pi}{2}-el\right)\cos(\beta)\right) \tag{19}$$

This Cartesian representation is utilized to illustrate the position of a satellite with respect to a ground station. This enables modeling such as satellite signal strength as a function of range to the station and Gaussian functions representing the cost/benefit when targeting these satellites for communications or observations.

5. Link Strength Analysis

Remotely operated spacecraft typically employ the use of radio frequency (RF) communications for C2; telemetry, tracking and control (TT&C); and payload operations. A TT&C system is composed of transmitters and receivers located on the spacecraft and

ground station. The transmitter antenna radiates an electromagnetic wave into free space, and the receiver antenna collects the fraction of energy which falls on it.

Designing a wireless RF communications architecture requires characterization of numerous complex interactions, many of which are difficult to predict. Therefore, communications systems carry additional design margins to account for uncertainties such as pointing losses, atmospheric and ionospheric attenuation, and realistic inefficiencies in cables, connectors, and antennas [90]. The end goal is to generate a signal strong enough to overcome these losses such that the quantity of RF energy at the receiver is at or above the minimum sensitivity requirement for the radio. The difference between the minimum threshold and the actual signal strength is called *link margin*. Designers will create a positive link margin to ensure a closed link. The amount of margin depends on the mission criticality of the link.

The minimum required signal strength of the receiver is given in proportion to the surrounding noise of the environment, which is a function of factors such as other transmitters, system temperature, and even cosmic noise. The ratio of the received signal strength to the surrounding system noise is defined as the signal-to-noise ratio (SNR). The systematic analysis of a communications system to arrive at SNR estimates is called a *link budget* [90].

The small form factor of CubeSats forces limited battery capacity and reduced solar panel sizes when compared to traditional larger spacecraft. This creates a particular challenge for communications systems where onboard transmission power must be restricted to allow other systems and payloads to continue operating. Establishing two-way communications with a CubeSat from the ground is often limited by the space-to-ground link, given the low power transmission generated by the satellite and the high power transmitters employed on the ground. This research predominantly focuses on optimizing communications as a function of the space-to-ground link as it is often the limiting factor for CubeSat communications.

To derive an analytic link budget [91] we begin with the equivalent isotropic radiated power (EIRP) of an antenna, which is a function of transmission power (P_{TX}),

antenna gain (G_{TX}), and cable/connector losses (L_{OUT}). EIRP and transmitter power is typically quantified in dBW [92].

$$EIRP = P_{TX} + G_{TX} - L_{OUT} \tag{20}$$

The free space path loss (FSPL) is the loss in signal strength of an electromagnetic wave as a function of distance and frequency.

$$FSPL = 20\log_{10}\left(\frac{4\pi}{c}\right) + 20\log_{10}(frequency) + 20\log_{10}(range) \tag{21}$$

When using frequency units in MHz, range units in km, and speed of light constant c = 2.99 x 10^5 km/sec the FSPL equation can be written as:

$$FSPL = 32.45 + 20\log_{10}(frequency) + 20\log_{10}(range) \tag{22}$$

Carrier signal strength is a function of EIRP, receiver antenna gain (G_{RX}), FSPL, atmospheric losses (L_{ATM}), receiver input losses (L_{IN}), pointing losses ($L_{POINTING}$), and polarization losses ($L_{POLARIZATION}$).

$$C = EIRP + G_{RX} - FSPL - L_{ATM} - L_{IN} - L_{POINTING} - L_{POLARIZATION} \tag{23}$$

The carrier-to-noise received power ratio can be subsequently computed as a function of system noise temperature T_s and the Boltzmann constant k = 1.38064852 × 10^{-23} m^2 kg s^{-2} K^{-1}

$$C / N_0 = C - 10\log_{10}(T_s) - 10\log_{10}(k) \tag{24}$$

As a digital signal is modulated, its carrier energy is spread in the frequency domain in proportion to the data rate in bits per second (bps) as governed by the Shannon-Hartley theorem. A common metric for assessing this behavior is the energy per bit of the system E_b. The energy per bit is then compared to system noise to produce a signal-to-noise ratio (SNR):

$$E_b / N_o = C / N_0 - 10\log_{10}(rate) \tag{25}$$

The difference between actual and required SNR is the system's link margin *M* and is often the governing parameter in the performance of a communications link.

$$M = \mathrm{E}_b / N_o - \mathrm{E}_b / N_o _ required \tag{26}$$

The link margin for any communications system depends on the modulation, encoding scheme, data rate, receiver sensitivity, and many other factors, and is therefore system-specific. Likewise, depending on the fidelity of the analysis and complex real-world interactions, the minimum value of *M* often carries a substantial margin, a factor of two (3 dB) or more [90]. If the link margin is significantly above the minimum required value, engineers will often increase the data rate to take advantage of extra capacity in the system. Likewise, a poor link margin necessitates a decrease in data rates to increase the energy-per-bit in the system.

To develop the initial algorithms used in this research, many of the parameters described above can be kept constant. It may also be assumed that the spacecraft was not rotating, and therefore had fixed antenna pointing losses. Additionally, data rates can be kept constant to simplify the algorithm development. Placeholders in the model allow future applications to add increased fidelity after the proof-of-concept stage.

This research seeks to optimize the use of ground-based antennas for satellite communications. The modeling of directional antennas is a required component of the problem formulation, given that the optimization algorithms will direct the antenna to track satellites of greatest benefit to the mission.

Directional antennas used in satellite communications systems radiate or receive greater energy in a particular direction. The energy decreases as a function of angle from the center and is often characterized as a loss in dB. Two examples of directional antennas used in this research are parabolic dishes and Yagi antennas. Figure 28 illustrates gain as a function of angle for a Yagi antenna. Parabolic antennas have a similar, albeit narrower, main beam and less pronounced side/back lobes. The gain of the beam will typically exceed that of a similarly sized Yagi antenna. Therefore, the dish is best suited for communications requiring higher power and better precision than the Yagi.

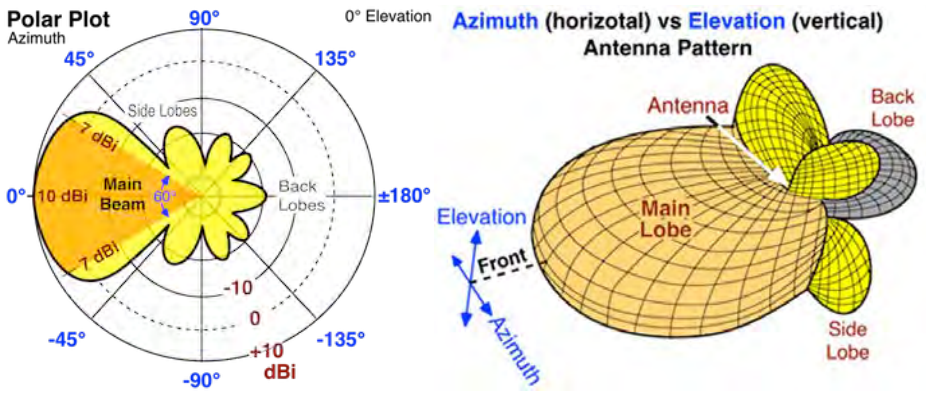


Figure 28. Typical polar diagram for a Yagi antenna. Source: [93].

The beamwidth for the antenna shown in Figure 28 is defined as the angle within which the edges of the main lobe are at half the power (3 dB) of the peak effective radiated power. To maximize signal strength, the antenna must be pointed within this tolerance when transmitting or receiving. Considerations governing operating within this beamwidth include antenna rotator position accuracy and satellite position knowledge.

This research can be applied to any style of antenna, if its parameters are integrated into the optimization model. The experimental results were obtained using Yagi antennas that are used in the MC3 system. From a practical standpoint, receiving signals with the Yagi antennas is possible within ±5 degrees off-center in each axis. In contrast, a typical dish antenna used by the MC3 system operates within ±0.5 degrees or better. When evaluating the pointing accuracy of the algorithms in Chapters IV and V, these values are used as benchmarks for pointing accuracy.

6. Kinematics and Dynamics

Ground-based sensors and tracking systems used for satellite communications and operations vary greatly depending on the application. The commonality between these systems is that there are at least two degrees of freedom required to position the ground-based sensor such that the satellite enters its field of view. Figure 29 shows three examples of satellite communications apertures with different positioners employed for tracking.



Left: A 3-meter parabolic dish at NPS employs an azimuth/elevation rotor system. Center: A 3-meter dish at AFIT uses a modern X/Y rotator architecture to minimize keyhole effects. Right: A 13-meter dish at UAF uses an azimuth/elevation positioning system with an additional 7° tilting capability for keyhole mitigation.

Figure 29. Three different satellite communications apertures.

Regardless of the physical means of pointing the aperture, the satellite communications problem requires that pointing to a local azimuth and elevation be performed as shown in Figure 24. The models developed in this dissertation assume the use of an azimuth and elevation positioner system which involves two large, geared motors mounted orthogonally and whose azimuth axis of rotation is aligned with zenith.

A good approximation for a geared motor moving an antenna is the double integrator model commonly used in dynamic analyses. The basis for the double integrator is that the motor provides a torque τ into the system causing an angular acceleration about the axis of rotation. The acceleration can be twice integrated as a function of time to produce velocity and position values. Mathematically, the double integrator model for the azimuth and elevation positioners is as follows:

$$\begin{aligned} \tau_{AZ_{SITE}} &= J_{AZ_{SITE}} \ddot{\theta}_{AZ_{SITE}} \\ \tau_{EL_{SITE}} &= J_{EL_{SITE}} \ddot{\theta}_{EL_{SITE}} \end{aligned} \tag{27}$$

and rearranged as:

$$\ddot{\theta}_{AZ_{SITE}} = \frac{\tau_{AZ_{SITE}}}{J_{AZ_{SITE}}}$$
$$\ddot{\theta}_{EL_{SITE}} = \frac{\tau_{EL_{SITE}}}{J_{EL_{SITE}}} \tag{28}$$

where J is the rotational moment of inertia of the system and θ is the angular position of the system. Since the motor torques are the control variables, the dynamic state-space formulation becomes:

$$\dot{\theta}_{AZ_{SITE}} = \omega_{AZ_{SITE}}$$
$$\dot{\theta}_{EL_{SITE}} = \omega_{EL_{SITE}}$$
$$\dot{\omega}_{AZ_{SITE}} = \frac{\tau_{AZ_{SITE}}}{J_{AZ_{SITE}}} \tag{29}$$
$$\dot{\omega}_{EL_{SITE}} = \frac{\tau_{EL_{SITE}}}{J_{EL_{SITE}}}$$

where ω is the angular velocity of each rotational degree of freedom. The angles map directly to the traditional topocentric reference frame described earlier in this chapter. The resulting system states are as follows:

$$x(t) = \begin{bmatrix} \theta_{AZ_{SITE}} & \theta_{EL_{SITE}} & \omega_{AZ_{SITE}} & \omega_{EL_{SITE}} \end{bmatrix} \tag{30}$$

The system's controls are summarized as a function of motor torques only:

$$u(t) = \begin{bmatrix} \tau_{AZ} & \tau_{EL} \end{bmatrix} \tag{31}$$

Both controls and states are necessary elements of the problem formulation described in Chapter III. Each degree of freedom requires an allowable range of values within which to operate. These correspond to the physical limits of the systems being modeled to provide solutions to real-world scenarios.

7. Target Model Development

To illustrate the use of the value function and Gaussian in the satellite tracking problem we introduce a typical scenario for a satellite passing a ground station. Figure 30 shows a simulated example of a satellite in a representative orbit passing an MC3 ground station located at NPS in Monterey, CA. The duration of the encounter lasts less than ten minutes. The satellite approaches the station from the south-west, passes to the south, and recedes from view in the north-east. With the locations of both satellite and ground station modeled in STK, the ECEF positions can be extracted to compute the position relative to the station and corresponding slant range. These data sets are comprised of azimuth, elevation, and range (AER) values.



Figure 30. Satellite flyby of the NPS ground station at three time points: acquisition of signal (top left), point of closest approach (top right), and loss of signal (bottom).

This research defines AOS and LOS thresholds at 10° above the horizon. In this example, the slant-range distance at AOS/LOS between spacecraft and ground station varies from approximately 1,200 km to 460 km at the time of closest approach (TCA). These distances are fed into the free space path loss calculation defined in the section II.B.5, using a frequency of 914 MHz (the downlink frequency of the PROPCUBE satellites, described in Chapter V). The link margin can then be computed for each time step in the encounter. Figure 31 shows the computed link margin for the above scenario based on these equations.

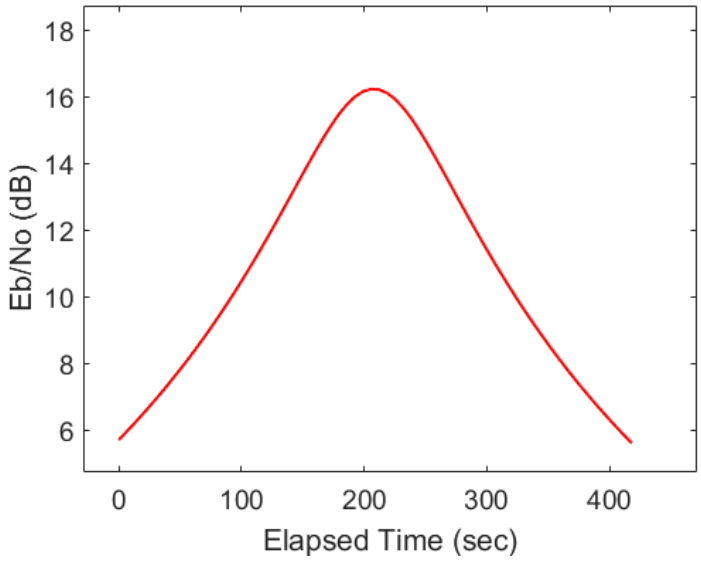
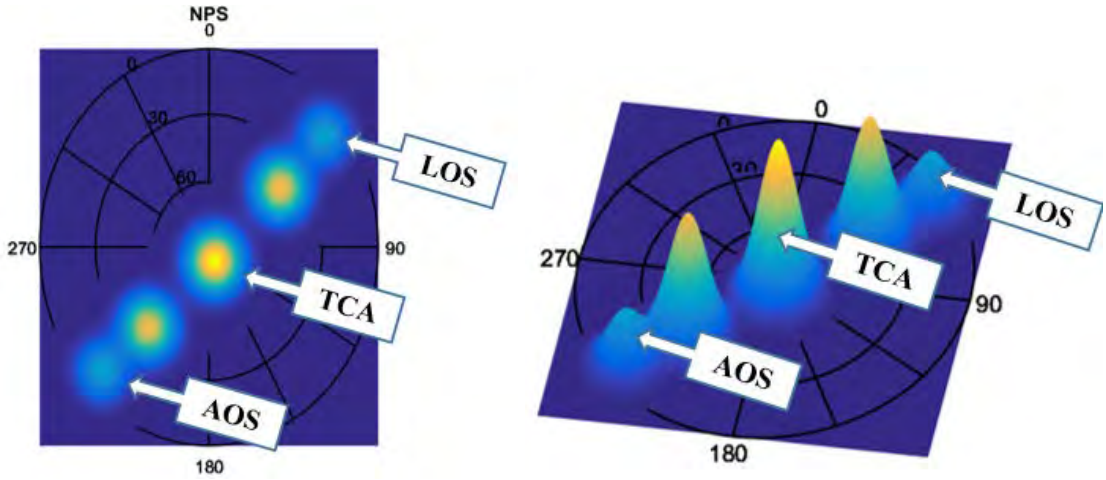


Figure 31. Idealized link margin calculation for the CubeSat as it passes the NPS ground station (assumes no spacecraft rotation/antenna pointing losses, and no environmental noise).

Using AER data extracted for this scenario, the target Gaussian can be plotted as a function of azimuth and elevation presented in Figure 32. The computed link margin value is used as the scaling function *v* from Equation (7) to change the amplitude of the Gaussian. This is beneficial for an optimization problem that acquires moving targets. As the satellite approaches a station, its signal strength is maximized at TCA and consequently the target is of greatest value for attempting a data downlink. The scaling function can involve higher

fidelity physical models to better represent reality. It can also include a prioritization multiplier that includes qualitative attributes such as customer priority, mission urgency, etc.



Gaussian peak value is mapped to calculated space-to-ground link margin and is maximized at the point of closest approach. Overhead (left) and oblique views shown

Figure 32. Azimuth/elevation polar plot representation of ground station and satellite Gaussian.

The solvers used for the optimization problems presented in the following chapter require smooth surfaces in the search space from which global maxima and minima can be computed to find an optimal trajectory. Leveraging the Gaussian targeting method was made possible by adapting the mathematical framework presented in this chapter. Both analytic (low accuracy) and numerical (high accuracy) propagations can be used to compute the Gaussians' location with respect to the ground stations and including the link budget calculations offers additional model fidelity. However, despite growing in complexity, these models are simplified from reality, acting as a proof-of-concept.

C. CHAPTER SUMMARY

This chapter covered the fundamental principles necessary for modeling the space-ground communications scenario. Previous work was presented and contrasted against the modeling approaches used for this research effort. The well-studied physical attributes discussed in this chapter were combined into a single representation of the satellite/ground station targeting model which enables a unique problem formulation that can leverage optimal control theory. The specifics of the problem formulation are developed and validated in the following chapter.

III. PROBLEM FORMULATION AND VALIDATION

This chapter presents the full problem formulation utilizing optimal control fundamentals to solve the multiple satellite/multiple ground station tracking problem. To accomplish this, the physical models discussed in Chapter II are applied to the cases illustrated in Chapter I. Specifically, the problem formulation is generalized such that an arbitrary number of satellites and ground stations can be considered. This chapter subsequently steps through the validation and verification process of the simplest case involving one satellite and one ground station. Validating the formulation and algorithm implementation for the simplest case enables the analysis of more complex scenarios in Chapter IV and integration into real systems shown in Chapter V.

A. INITIAL PROBLEM FORMULATION

To get an understanding of the mathematical behavior associated with the satellite-ground station targeting problem, the initial formulation was created and analyzed using optimal control theory. The equations shown in this section were then compared to the results generated by the software implementation. These initial results served to demonstrate the initial capability of modeling and implementing the satellite tracking problem within a dynamic optimization framework.

1. Benefit Value Function

This example's cost function V, also called a benefit value function (BVF), consists of a Gaussian scaled by factor v enumerated for each satellite when in view of each ground station in the planning horizon [47]. Equation (32) describes the BVF and is directly applied from Equation (7):

$$V\left(t;\theta_{AZ_{SITE}},\theta_{EL_{SITE}},\omega_{AZ_{SITE}},\omega_{EL_{SITE}},M,\theta_{AZ_{SAT}},\theta_{EL_{SAT}}\right)=$$
$$=\sum_{i=0}^{n_{SAT}}\sum_{j=0}^{n_{SITE}}\left(v_{ij}e^{-\left(\frac{(x_{iSAT}-x_{jSITE})^2}{2\sigma_x{}^2}+\frac{(y_{iSAT}-y_{jSITE})^2}{2\sigma_y{}^2}\right)}\right) \tag{32}$$

Benefit value function V is dependent on several parameters. First, V consists of antenna rotor position azimuth and elevation angles $\theta_{AZ_{SITE}}$ and $\theta_{EL_{SITE}}$ translated into Cartesian coordinates (X_{SITE}, Y_{SITE}) as defined in Equation (19). The satellite target azimuth and elevation angles $\theta_{AZ_{SAT}}$ and $\theta_{EL_{SAT}}$ are derived from positions transformed from the ECI frame (Equation (10)), to ECEF (Equation (14)), to ENU, and finally to AER using Equation (18). The angles change as a function of time based on the orbital path of the satellite and the rotation of the Earth. Similarly, the satellite azimuth and elevation angles are translated into Cartesian coordinates (X_{SAT}, Y_{SAT}), also using Equation (19). Appendix A shows the entire lengthy expression corresponding to X_{SAT} and Y_{SAT} for this analytic case. A direct comparison between (X_{SAT}, Y_{SAT}) and the antenna position (X_{SITE}, Y_{SITE}) can now be made in the BVF since both are in an equivalent reference frame. The closer the two values, the higher the Gaussian's magnitude. The highest magnitude is therefore when the antenna position matches the satellite position. The value of σ_x and σ_y defines the Gaussian's target area and the values are user-adjustable, determining the area of influence each satellite's Gaussian exerts in the scenario. Larger values correspond to a bigger footprint so an operator may opt to give higher priority satellites a large Gaussian footprint, for example. The dynamic optimization solution would then select such a target sooner than one with a narrower footprint. The satellite range term is used to compute the idealized link margin M derived in Equation (26) creating the v term. This term is used for scaling the Gaussian in the problem formulations.

2. Problem Formulation

Formulating the initial problem first requires identification of the states and controls. The states correspond to the angular velocity and position of the antenna rotors from the double integrator model presented in Chapter II.B.6. The positions and velocities are a function of the control torques τ for each axis. The state-space representation is shown in Equation (33) and after introducing the BVF from Equation (32), the full analytic problem formulation becomes:

$$x = \begin{bmatrix} \theta_{AZ_{SITE}} & \theta_{EL_{SITE}} & \omega_{AZ_{SITE}} & \omega_{EL_{SITE}} \end{bmatrix} \qquad u = \begin{bmatrix} \tau_{AZ_{SITE}} & \tau_{EL_{SITE}} \end{bmatrix} \tag{33}$$

$$\begin{cases} \text{Minimize} \quad J[x(\cdot), \mathrm{u}(\cdot), t_f] = -\int_{t_0}^{t_f} \left(\sum_{i=0}^{n_{SAT}} \sum_{j=0}^{n_{SITE}} \left(v_{ij} e^{-\left(\frac{(x_{iSAT} - x_{jSITE})^2}{2\sigma_x{}^2} + \frac{(y_{iSAT} - y_{jSITE})^2}{2\sigma_y{}^2} \right)} \right) \right) dt \\ \text{subject to} \quad \dot{\theta}_{AZ_{SITE}} = \omega_{AZ_{SITE}} \\ \dot{\theta}_{EL_{SITE}} = \omega_{EL_{SITE}} \\ \dot{\omega}_{AZ_{SITE}} = \frac{\tau_{AZ_{SITE}}}{J_{AZ_{SITE}}} \\ \dot{\omega}_{EL_{SITE}} = \frac{\tau_{EL_{SITE}}}{J_{EL_{SITE}}} \\ t_0 = t^0 \\ t_f = t^f \\ v = M \\ \sigma_x, \sigma_y = \sigma^{x_{SAT}}, \sigma^{y_{SAT}} \\ \theta_{AZ\min_{SITE}} \le \theta_{AZ_{SITE}} \le \theta_{AZ\max_{SITE}} \\ \theta_{EL\min_{SITE}} \le \theta_{EL_{SITE}} \le \theta_{EL\max_{SITE}} \\ \omega_{AZ\min_{SITE}} \le \omega_{AZ_{SITE}} \le \omega_{AZ\max_{SITE}} \\ \omega_{EL\min_{SITE}} \le \omega_{EL_{SITE}} \le \omega_{EL\max_{SITE}} \\ \tau_{AZ\min_{SITE}} \le \tau_{AZ_{SITE}} \le \tau_{AZ\max_{SITE}} \\ \tau_{EL\min_{SITE}} \le \tau_{EL_{SITE}} \le \tau_{EL\max_{SITE}} \\ \left(x_{SAT}, y_{SAT} \right) = \frac{2}{\pi} \left(\left(\frac{\pi}{2} - \theta_{EL_{SAT}} \right) \sin\left(\theta_{AZ_{SAT}} \right), \left(\frac{\pi}{2} - \theta_{EL_{SAT}} \right) \cos\left(\theta_{AZ_{SAT}} \right) \right) \\ \left(x_{SITE}, y_{SITE} \right) = \frac{2}{\pi} \left(\left(\frac{\pi}{2} - \theta_{EL_{SITE}} \right) \sin\left(\theta_{AZ_{SITE}} \right), \left(\frac{\pi}{2} - \theta_{EL_{SITE}} \right) \cos\left(\theta_{AZ_{SITE}} \right) \right) \end{cases} \tag{34}$$

The cost functional equation described in Equation (3) is restated in Equation (34) to only include a running cost and no endpoint cost. The running cost is the BVF from Equation (32) summed for each ground station and satellite pairing for each timestep of the

scenario. To adhere to Pontryagin's Principle, the cost function must be minimized, subject to various constraints for converging on an optimal solution. Since the problem formulation involves maximizing the benefit (e.g., the maximum area under the curve of each Gaussian), the problem formulation requires minimizing the negative of this value. Therefore, the cost function of Equation (34) includes a negative sign applied once all of the individual BVFs for each satellite and ground station pairing have been summed.

With the cost function defined, the problem constraints are then introduced. A second-order differential equation relationship is established between antenna position angles, angular velocity, and angular acceleration. The angular acceleration terms stem from the torque controls in this example, and thus have a direct time-based effect on the antenna position. No drag/friction in the rotor gearbox was modeled for these demonstrations, though systems of increased complexity may benefit from the additional fidelity.

Next, the scenario start time is defined, followed by the term(s) used to scale the individual Gaussians. The v term is mapped to the link margin M, calculated in Equation (26), for each satellite/ground station pair. This scaling term adjusts the height of the Gaussian and can be as complex as the link budget equations will allow, though the simplified cases presented in this dissertation only assume a changing free space path loss with all other variables kept constant. The σ_x and σ_y terms adjust the footprint size of the Gaussian in each axis and are user-selectable.

The next six terms are *path constraints* bounding angular positions, velocities, and torques within an allowable range. The final two constraints explicitly describe the translation from local azimuth and elevation angles to Cartesian grid points. This facilitates computation of target Gaussians as shown in Equation (19). These conversions occur for both satellite targets and ground station antenna angles.

a. Hamiltonian and KKT Conditions

Following the process described in Chapter II.A, the Hamiltonian is first constructed using the general form:

$$H(\lambda,x,u,t) = F(x,u,t) + \lambda^T f(x,u,t) \tag{35}$$

and applying the problem states, controls, and constraints defined in Equations (33) and (34) to produce:

$$H = -ve^{-\left(\frac{(x_{iSAT}-x_{jSITE})^2}{2\sigma_x{}^2}+\frac{(y_{iSAT}-y_{jSITE})^2}{2\sigma_y{}^2}\right)} + \lambda_{\theta_{AZ_{SITE}}}\omega_{AZ_{SITE}} + \lambda_{\theta_{EL_{SITE}}}\omega_{EL_{SITE}}\ldots$$
$$+\lambda_{\omega_{AZ_{SITE}}}\frac{\tau_{AZ_{SITE}}}{J_{AZ_{SITE}}} + \lambda_{\omega_{EL_{SITE}}}\frac{\tau_{EL_{SITE}}}{J_{EL_{SITE}}} \tag{36}$$

The path constraints presented in the problem formulation require the usage of additional multipliers in the Hamiltonian. These *path covectors* form the *Lagrangian of the Hamiltonian*, the general form of which is stated as:

$$\overline{H}(\mu,\lambda,x,u,t) = H(\lambda,x,u,t) + \mu^T h(x,u,t) \tag{37}$$

Substituting values specific to the problem yields the following:

$$\overline{H} = -ve^{-\left(\frac{(x_{iSAT}-x_{jSITE})^2}{2\sigma_x{}^2}+\frac{(y_{iSAT}-y_{jSITE})^2}{2\sigma_y{}^2}\right)} + \lambda_{\theta_{AZ_{SITE}}}\omega_{AZ_{SITE}} + \lambda_{\theta_{EL_{SITE}}}\omega_{EL_{SITE}}\ldots$$
$$+\lambda_{\omega_{AZ_{SITE}}}\frac{\tau_{AZ_{SITE}}}{J_{AZ_{SITE}}} + \lambda_{\omega_{EL_{SITE}}}\frac{\tau_{EL_{SITE}}}{J_{EL_{SITE}}}\ldots$$
$$+\mu_{\theta_{AZ_{SITE}}}\theta_{AZ_{SITE}} + \mu_{\theta_{EL_{SITE}}}\theta_{EL_{SITE}} + \mu_{\omega_{AZ_{SITE}}}\omega_{AZ_{SITE}} + \mu_{\omega_{EL_{SITE}}}\omega_{EL_{SITE}}\ldots$$
$$+\mu_{\tau_{AZ_{SITE}}}\tau_{AZ_{SITE}} + \mu_{\tau_{EL_{SITE}}}\tau_{EL_{SITE}} \tag{38}$$

The theoretical behaviors of the problem formulation emerge by analyzing the Karush-Kuhn-Tucker (KKT) [79] conditions, which implement the stationary condition:

$$\frac{\partial \overline{H}}{\partial u} = \frac{\partial H}{\partial u} + \left(\frac{\partial h}{\partial u}\right)^T \mu = 0$$
$$\frac{\partial \overline{H}}{\partial x} = \frac{\partial H}{\partial x} + \left(\frac{\partial h}{\partial x}\right)^T \mu = 0 \tag{39}$$

and the complimentary condition:

$$\mu_{u,i} \begin{cases} \leq 0 \\ = 0 \\ \geq 0 \\ unrestricted \end{cases} \quad if \quad \begin{matrix} h_i(u) = h_i^L \\ h_i^L < h_i(u) < h_i^U \\ h_i(u) = h_i^U \\ h_i^L = h_i^U \end{matrix} \tag{40}$$

Applying problem-specific parameters, the following relationship is found:

$$\frac{\partial \overline{H}}{\partial \tau_{AZ_{SITE}}} = \frac{\lambda_{\theta_{AZ_{SITE}}}}{J_{AZ_{SITE}}} + \mu_{\tau_{AZ_{SITE}}} = 0$$
$$\frac{\partial \overline{H}}{\partial \tau_{EL_{SITE}}} = \frac{\lambda_{\theta_{EL_{SITE}}}}{J_{EL_{SITE}}} + \mu_{\tau_{EL_{SITE}}} = 0 \tag{41}$$

which can be rewritten as:

$$\mu_{\tau_{AZ_{SITE}}} = -\frac{\lambda_{\theta_{AZ_{SITE}}}}{J_{AZ_{SITE}}}$$
$$\mu_{\tau_{EL_{SITE}}} = -\frac{\lambda_{\theta_{EL_{SITE}}}}{J_{EL_{SITE}}} \tag{42}$$

The KKT conditions of Equation (42) are later confirmed in the software implementation V&V analysis presented in Section C of this chapter.

b. Adjoint Equations

The adjoint equations can be useful for determining the behavior of the costates associated with the problem formulation, its states, and cost function. The general form of the costate equation is:

$$-\dot{\lambda}(t) = \frac{\partial \bar{H}}{\partial x} \tag{43}$$

Applying this equation to the states for this problem formulation generates the following complex relationships, starting with the degrees of freedom corresponding to the azimuth and elevation:

$$-\dot{\lambda}_{\theta_{AZ_{SITE}}}(t) = \frac{\partial \overline{H}}{\partial \theta_{AZ_{SITE}}} =$$

$$= \begin{Bmatrix} \frac{4}{\pi\sigma_x^{\,2}}\left[-x_{SAT}\left(\frac{\pi}{2}-\theta_{EL_{SITE}}\right)\cos\left(\theta_{AZ_{SITE}}\right)+\left(\left(\frac{\pi}{2}-\theta_{EL_{SITE}}\right)\sin\left(\theta_{AZ_{SITE}}\right)\right)\left(\left(\frac{\pi}{2}-\theta_{EL_{SITE}}\right)\cos\left(\theta_{AZ_{SITE}}\right)\right)\right]+\ldots \\ \frac{4}{\pi\sigma_y^{\,2}}\left[\mathrm{y}_{SAT}\left(\left(\frac{\pi}{2}-\theta_{EL_{SITE}}\right)\sin\left(\theta_{AZ_{SITE}}\right)\right)-\left(\left(\frac{\pi}{2}-\theta_{EL_{SITE}}\right)\cos\left(\theta_{AZ_{SITE}}\right)\right)\left(\left(\frac{\pi}{2}-\theta_{EL_{SITE}}\right)\sin\left(\theta_{AZ_{SITE}}\right)\right)\right] \end{Bmatrix} * \ldots$$

$$ve \begin{Bmatrix} \frac{1}{2\sigma_x^{\,2}}\left[x_{SAT}{}^2-\frac{4}{\pi}x_{SAT}\left(\frac{\pi}{2}-\theta_{EL_{SITE}}\right)\sin\left(\theta_{AZ_{SITE}}\right)+\left(\frac{2}{\pi}\left(\frac{\pi}{2}-\theta_{EL_{SITE}}\right)\sin\left(\theta_{AZ_{SITE}}\right)\right)^2\right]+\ldots \\ \frac{1}{2\sigma_y^{\,2}}\left[\mathrm{y}_{SAT}{}^2-\frac{4}{\pi}\mathrm{y}_{SAT}\left(\left(\frac{\pi}{2}-\theta_{EL_{SITE}}\right)\cos\left(\theta_{AZ_{SITE}}\right)\right)+\left(\frac{2}{\pi}\left(\frac{\pi}{2}-\theta_{EL_{SITE}}\right)\cos\left(\theta_{AZ_{SITE}}\right)\right)^2\right] \end{Bmatrix} + \mu_{\theta_{AZ_{SITE}}}$$

$$-\dot{\lambda}_{\theta_{EL_{SITE}}}(t) =$$

$$= \begin{Bmatrix} \frac{2\sin\left(\theta_{AZ_{SITE}}\right)}{\sigma_x^{\,2}\pi^2}\left(x_{SAT}\pi-\pi\sin\left(\theta_{AZ_{SITE}}\right)+2\theta_{EL_{SITE}}\sin\left(\theta_{AZ_{SITE}}\right)\right)+\ldots \\ \frac{2\cos\left(\theta_{AZ_{SITE}}\right)}{\sigma_y^{\,2}\pi^2}\left(y_{SAT}\pi-\pi\cos\left(\theta_{AZ_{SITE}}\right)+2\theta_{EL_{SITE}}\cos\left(\theta_{AZ_{SITE}}\right)\right) \end{Bmatrix} * \ldots$$

$$ve \begin{Bmatrix} \frac{1}{2\sigma_x^{\,2}}\left[x_{SAT}{}^2-\frac{4}{\pi}x_{SAT}\left(\frac{\pi}{2}-\theta_{EL_{SITE}}\right)\sin\left(\theta_{AZ_{SITE}}\right)+\left(\frac{2}{\pi}\left(\frac{\pi}{2}-\theta_{EL_{SITE}}\right)\sin\left(\theta_{AZ_{SITE}}\right)\right)^2\right]\ldots \\ +\frac{1}{2\sigma_y^{\,2}}\left[\mathrm{y}_{SAT}{}^2-\frac{4}{\pi}\mathrm{y}_{SAT}\left(\left(\frac{\pi}{2}-\theta_{EL_{SITE}}\right)\cos\left(\theta_{AZ_{SITE}}\right)\right)+\left(\frac{2}{\pi}\left(\frac{\pi}{2}-\theta_{EL_{SITE}}\right)\cos\left(\theta_{AZ_{SITE}}\right)\right)^2\right] \end{Bmatrix} + \ldots$$

$$\mu_{\theta_{EL}SITE} \tag{44}$$

The remainder of the adjoint equations are as follows:

$$-\dot{\lambda}_{\omega_{AZ_{SITE}}}(t) = \lambda_{\theta_{AZ_{SITE}}} + \mu_{\omega_{AZ_{SITE}}}$$
$$-\dot{\lambda}_{\omega_{EL_{SITE}}}(t) = \lambda_{\theta_{EL_{SITE}}} + \mu_{\omega_{EL_{SITE}}} \tag{45}$$

The adjoint equations in (44) are clearly complex and difficult to analyze manually due to the use of the Gaussian BVF in the cost function. The remaining equations in (45) show a more predictable relationship with the ω costate derivatives, depending on their corresponding μ path constraint covectors and θ costate values.

c. Hamiltonian Evolution Equation

Evaluating the *Hamiltonian Evolution Equation* determines whether or not there is a time-dependent behavior in the Hamiltonian. If no time-dependency exists, its derivative is zero and the minimized Hamiltonian remains constant. However, this problem formulation contains a time dependency in the satellite position terms and corresponding link budget calculation. X_{SAT}, Y_{SAT}, and v vary as a function of time since the satellite target changes with every time step of the problem. The Hamiltonian Evolution Equation is as follows:

$$\dot{H} = \frac{dH}{dt} = \frac{\partial \overline{H}}{\partial t} = \frac{\partial \begin{pmatrix} -v(t)e^{-\left(\frac{(x_{SAT}(t)-x_{SITE})^2}{2\sigma_x^2}+\frac{(y_{SAT}(t)-y_{SITE})^2}{2\sigma_y^2}\right)} + \lambda_{\theta_{AZ}SITE}\omega_{AZ_{SITE}} + \lambda_{\theta_{EL}SITE}\omega_{EL_{SITE}}\ldots \\ +\lambda_{\omega_{AZ_{SITE}}}\frac{\tau_{AZ_{SITE}}}{J_{AZ_{SITE}}} + \lambda_{\omega_{EL_{SITE}}}\frac{\tau_{EL_{SITE}}}{J_{EL_{SITE}}} + \mu_{\theta_{AZ_{SITE}}}\theta_{AZ_{SITE}} + \mu_{\theta_{EL_{SITE}}}\theta_{EL_{SITE}}\ldots \\ +\mu_{\omega_{AZ_{SITE}}}\omega_{AZ_{SITE}} + \mu_{\omega_{EL_{SITE}}}\omega_{EL_{SITE}} + \mu_{\tau_{AZ_{SITE}}}\tau_{AZ_{SITE}} + \mu_{\tau_{EL_{SITE}}}\tau_{EL_{SITE}} \end{pmatrix}}{\partial t} =$$
$$= -M(t)\left(\frac{(x_{SAT}(t)-x_{SITE})}{\sigma_x^{\,2}} + \frac{(y_{SAT}(t)-y_{SITE})}{\sigma_y^{\,2}}\right)e^{-\left(\frac{(x_{SAT}(t)-x_{SITE})^2}{2\sigma_x^2}+\frac{(y_{SAT}(t)-y_{SITE})^2}{2\sigma_y^2}\right)}\ldots$$
$$-M'(t)e^{-\left(\frac{(x_{SAT}(t)-x_{SITE})^2}{2\sigma_x^2}+\frac{(y_{SAT}(t)-y_{SITE})^2}{2\sigma_y^2}\right)} \tag{46}$$

The computation of variables X_{SAT}, Y_{SAT}, and M terms in Equation (46) have an explicit time-dependence (see Appendices A and B). Therefore, the partial time derivative of the Hamiltonian is non-zero. Consequently, this behavior should be expressed when

analyzing results from the software implementation. The fully derived values for X_{SAT} and Y_{SAT} are shown in Appendix A, and those corresponding to link margin *M* in Appendix B.

d. Transversality

The transversality conditions apply to the endpoint function of the problem. The general form is given by the following:

$$\bar{E} = E + \nu^T e = 0 \tag{47}$$

The cost function does not contain an endpoint (Mayer) cost, which means that $E = 0$. The only defined endpoint is the time constraint in which the problem has a fixed time horizon. This leads to an endpoint Lagrangian of:

$$\bar{\bar{E}} = \nu_{t_0}(t_0 - 0) + \nu_{t_f}(t_f - T) \tag{48}$$

Further analyzing the initial and final transversality conditions leads to the general form:

$$\begin{aligned} \lambda(t_0) &= -\frac{\partial \bar{\bar{E}}}{\partial x_0} = 0 \\ \lambda(t_f) &= \frac{\partial \bar{\bar{E}}}{\partial x_f} = 0 \end{aligned} \tag{49}$$

The transversality conditions are zero because there are no state variables in the endpoint Lagrangian. The following relationships therefore exist for this problem:

$$
\begin{aligned}
\lambda_{\theta_{AZ}SITE}(t_0) &= -\frac{\partial \overline{E}}{\partial \theta_{AZ_{SITE\ 0}}} = 0 \\
\lambda_{\theta_{AZ}SITE}(t_f) &= \frac{\partial \overline{E}}{\partial \theta_{AZ_{SITE\ f}}} = 0 \\
\lambda_{\theta_{EL}SITE}(t_0) &= -\frac{\partial \overline{E}}{\partial \theta_{EL_{SITE\ 0}}} = 0 \\
\lambda_{\theta_{EL}SITE}(t_f) &= \frac{\partial \overline{E}}{\partial \theta_{EL_{SITE\ f}}} = 0 \\
\lambda_{\omega_{AZ}SITE}(t_0) &= -\frac{\partial \overline{E}}{\partial \omega_{AZ_{SITE\ 0}}} = 0 \\
\lambda_{\omega_{AZ}SITE}(t_f) &= \frac{\partial \overline{E}}{\partial \omega_{AZ_{SITE\ f}}} = 0 \\
\lambda_{\omega_{EL}SITE}(t_0) &= -\frac{\partial \overline{E}}{\partial \omega_{EL_{SITE\ 0}}} = 0 \\
\lambda_{\omega_{EL}SITE}(t_f) &= \frac{\partial \overline{E}}{\partial \omega_{EL_{SITE\ f}}} = 0
\end{aligned}
\tag{50}
$$

Lastly, using Equation (48), the two Hamiltonian value conditions become:

$$
\begin{aligned}
H\left[@t_0\right] &= \frac{\partial \overline{E}}{\partial t_0} = \nu_{t_0} \\
H\left[@t_f\right] &= -\frac{\partial \overline{E}}{\partial t_f} = -\nu_{t_f}
\end{aligned}
\tag{51}
$$

These transversality conditions comprise the endpoint boundary conditions for the analytic propagator problem. The software implementation of this problem should generate costates and Hamiltonian boundary values that correspond to the above equations and will be confirmed in the V&V section of this chapter.

B. PROBLEM FORMULATION WITH TORQUE SMOOTHING

A more complex problem formulation involves enhancing the initial formulation with an additional torque term for each degree of freedom in the cost function, shown in Equation (52). These terms act as a penalty for extraneous control inputs since their sign is opposite to that of the Gaussian BVF. When minimizing the cost function, the torque signs remain positive whereas the BVF is negative. Therefore, the greater the torque, the greater the penalty in the cost function.

$$J[x(\cdot),\mathrm{u}(\cdot),t_f] = -\int_{t_0}^{t_f}\left(\sum_{i=0}^{n_{SAT}}\sum_{j=0}^{n_{SITE}}\left(v_{ij}e^{-\left(\frac{(x_{iSAT}-x_{jSITE})^2}{2\sigma_x{}^2}+\frac{(y_{iSAT}-y_{jSITE})^2}{2\sigma_y{}^2}\right)}\right)+\tau_{AZ_{SITE}}(t)^2+\tau_{EL_{SITE}}(t)^2\right)dt \tag{52}$$

The inclusion of the torque terms is beneficial, as the previous two examples only required that the Gaussian targets were tracked and made no stipulations as to how many control inputs were necessary to achieve the desired conditions if the states and controls remained within bounds. A valid solution could therefore include excessively oscillatory control inputs that still satisfied position and velocity constraints, but had little relevance to real-world systems due to jitter, pointing losses, and mechanical wear. This behavior will be explored in greater detail in the following sections; however, there is a mathematical reasoning to this behavior shown in the KKT conditions.

The states and controls for this problem remain equivalent to Equation (33), as shown in Equation (53):

$$x=\begin{bmatrix}\theta_{AZ_{SITE}} & \theta_{EL_{SITE}} & \omega_{AZ_{SITE}} & \omega_{EL_{SITE}}\end{bmatrix} \qquad u=\begin{bmatrix}\tau_{AZ_{SITE}} & \tau_{EL_{SITE}}\end{bmatrix} \tag{53}$$

The full problem formulation combines the familiar elements of the previous double integrator example with the modified cost function to produce the following:

$$
\begin{cases}
\text{Minimize} \quad J[x(\cdot),\mathrm{u}(\cdot),t_f] = -\int_{t_0}^{t_f} \left(\sum_{i=0}^{n_{SAT}} \sum_{j=0}^{n_{SITE}} \left(v_{ij} e^{-\left(\frac{(x_{iSAT}-x_{jSITE})^2}{2\sigma_x^2} + \frac{(y_{iSAT}-y_{jSITE})^2}{2\sigma_y^2} \right)} \right) + \tau_{AZ_{SITE}}(t)^2 + \tau_{EL_{SITE}}(t)^2 \right) dt \\
\text{subject to} \\
\qquad \dot{\theta}_{AZ_{SITE}} = \omega_{AZ_{SITE}} \\
\qquad \dot{\theta}_{EL_{SITE}} = \omega_{EL_{SITE}} \\
\qquad \dot{\omega}_{AZ_{SITE}} = \dfrac{\tau_{AZ_{SITE}}}{J_{AZ_{SITE}}} \\
\qquad \dot{\omega}_{EL_{SITE}} = \dfrac{\tau_{EL_{SITE}}}{J_{EL_{SITE}}} \\
\qquad t_0 = 0 \\
\qquad v = M_{SAT->SITE} \\
\qquad \sigma_x, \sigma_y = \sigma^{xSAT}, \sigma^{ySAT} \\
\qquad \theta_{AZ\min_{SITE}} \le \theta_{AZ_{SITE}} \le \theta_{AZ\max_{SITE}} \\
\qquad \theta_{EL\min_{SITE}} \le \theta_{EL_{SITE}} \le \theta_{EL\max_{SITE}} \\
\qquad \omega_{AZ\min_{SITE}} \le \omega_{AZ_{SITE}} \le \omega_{AZ\max_{SITE}} \\
\qquad \omega_{EL\min_{SITE}} \le \omega_{EL_{SITE}} \le \omega_{EL\max_{SITE}} \\
\qquad \tau_{AZ\min_{SITE}} \le \tau_{AZ_{SITE}} \le \tau_{AZ\max_{SITE}} \\
\qquad \tau_{EL\min_{SITE}} \le \tau_{EL_{SITE}} \le \tau_{EL\max_{SITE}} \\
\qquad (x_{SAT}, y_{SAT}) = \dfrac{2}{\pi}\left(\left(\dfrac{\pi}{2} - \theta_{EL_{SAT}} \right) \sin\left(\theta_{AZ_{SAT}} \right), \left(\dfrac{\pi}{2} - \theta_{EL_{SAT}} \right) \cos\left(\theta_{AZ_{SAT}} \right) \right) \\
\qquad (x_{SITE}, y_{SITE}) = \dfrac{2}{\pi}\left(\left(\dfrac{\pi}{2} - \theta_{EL_{SITE}} \right) \sin\left(\theta_{AZ_{SITE}} \right), \left(\dfrac{\pi}{2} - \theta_{EL_{SITE}} \right) \cos\left(\theta_{AZ_{SITE}} \right) \right)
\end{cases}
\tag{54}
$$

The torque penalties in the cost function seek to minimize extraneous control input and produce smoother trajectories. The mathematical rationale is found in the KKT condition evaluation. The Hamiltonian Lagrangian is adapted from Equation (38), and two torque terms are added as follows:

$$\overline{H} = -ve^{-\left(\frac{(x_{SAT}-x_{SITE})^2}{2\sigma_x{}^2}+\frac{(y_{SAT}-y_{SITE})^2}{2\sigma_y{}^2}\right)} + \lambda_{\theta_{AZ}SITE}\omega_{AZ_{SITE}} + \lambda_{\theta_{EL}SITE}\omega_{EL_{SITE}} \ldots$$
$$+\lambda_{\omega_{AZ}SITE}\frac{\tau_{AZ_{SITE}}}{J_{AZ_{SITE}}} + \lambda_{\omega_{EL}SITE}\frac{\tau_{EL_{SITE}}}{J_{EL_{SITE}}} + \mu_{\theta_{AZ}SITE}\theta_{AZ_{SITE}} + \mu_{\theta_{EL}SITE}\theta_{EL_{SITE}} \ldots$$
$$+\mu_{\omega_{AZ}SITE}\omega_{AZ_{SITE}} + \mu_{\omega_{EL}SITE}\omega_{EL_{SITE}} + \mu_{\tau_{AZ}SITE}\tau_{AZ_{SITE}} + \mu_{\tau_{EL}SITE}\tau_{EL_{SITE}} \ldots$$
$$+\tau_{AZ_{SITE}}(t)^2 + \tau_{EL_{SITE}}(t)^2 \tag{55}$$

Using Equation (39) the partial derivatives of the Hamiltonian create:

$$\begin{aligned}
\frac{\partial\overline{H}}{\partial\tau_{AZ_{SITE}}} &= \frac{\lambda_{\omega_{AZ}SITE}}{J_{AZ_{SITE}}} + \mu_{\tau_{AZ}SITE} + 2\tau_{AZ_{SITE}} = 0 \\
\frac{\partial\overline{H}}{\partial\tau_{EL_{SITE}}} &= \frac{\lambda_{\omega_{EL}SITE}}{J_{EL_{SITE}}} + \mu_{\tau_{EL}SITE} + 2\tau_{EL_{SITE}} = 0 \\
\frac{\partial\overline{H}}{\partial\omega_{AZ_{SITE}}} &= \lambda_{\theta_{AZ}SITE} + \mu_{\omega_{AZ}SITE} = 0 \\
\frac{\partial\overline{H}}{\partial\omega_{EL_{SITE}}} &= \lambda_{\theta_{EL}SITE} + \mu_{\omega_{EL}SITE} = 0 \\
\frac{\partial\overline{H}}{\partial\theta_{AZ_{SITE}}} &= \frac{\partial F}{\partial\theta_{AZ_{SITE}}} + \mu_{\theta_{AZ}SITE} = 0 \\
\frac{\partial\overline{H}}{\partial\theta_{EL_{SITE}}} &= \frac{\partial F}{\partial\theta_{EL_{SITE}}} + \mu_{\theta_{EL}SITE} = 0
\end{aligned} \tag{56}$$

where for simplicity the Gaussian term in the Hamiltonian is replaced by a function F, such that:

$$F(\theta_{AZ_{SITE}} \quad \theta_{EL_{SITE}} \quad \theta_{AZ_{SAT}} \quad \theta_{EL_{SAT}}) = -ve^{-\left(\frac{(x_{SAT}-x_{SITE})^2}{2\sigma_x{}^2}+\frac{(y_{SAT}-y_{SITE})^2}{2\sigma_y{}^2}\right)} \tag{57}$$

The key difference with the implementation of Equations (41) and (56) is found in the torque control expressions. The two equations concerning $\tau_{AZ_{SITE}}$ and $\tau_{EL_{SITE}}$ control inputs in Equation (41) do not have any torques in the expression. This means that the costate λ and path covector μ can adhere to the KKT conditions of (40) for any arbitrary value of τ. As long as the torque values did not exceed bounds specified in the path constraints, they were free to assume any value necessary to satisfy the KKT conditions. This most can result in a *bang-bang* control scheme, since it is most efficient to use the most torque available in the system if there is no penalty.

The relations derived in Equation (56) take a different approach because there are only a finite set of τ values that can simultaneously satisfy the equality with λ and μ covectors. Since the goal is to minimize the requisite control inputs to satisfy the constraints, some fraction of the full torque is then used. The primary challenge of this implementation then becomes one of scaling the torque penalty to prevent the system from avoiding the targets altogether. If the control input penalty were disproportionately weighted over the target Gaussians, there would be no advantage to tracking the satellites.

C. ONE SATELLITE, ONE GROUND STATION

The following section introduces the simplest tracking case for V&V analysis: the one satellite, one ground station problem, shown in Figure 33. This example allows direct inspection of solution outputs to verify whether the software implementation of the problems described above produces a valid result. This initial example involves two tests. The first directly compares the simplified analytic propagator with that of ephemerides produced by STK. This allows insight into some behaviors of the propagator-based solution that may otherwise be lost in the complexity of using externally generated ephemerides. The second test, building on the stability demonstrated in the first propagator-based trial, demonstrates the additional use of torque-smoothing terms to limit controller-induced oscillations.

The scenario established to test this case involves a spacecraft in LEO and a single ground station (GS1). Though meant to be a simplified case, these simulations leveraged

the same software that real-world applications would also use. The key differences between the case described in this chapter and more complex cases shown in Chapter IV are the quantities and locations of stations and satellites.

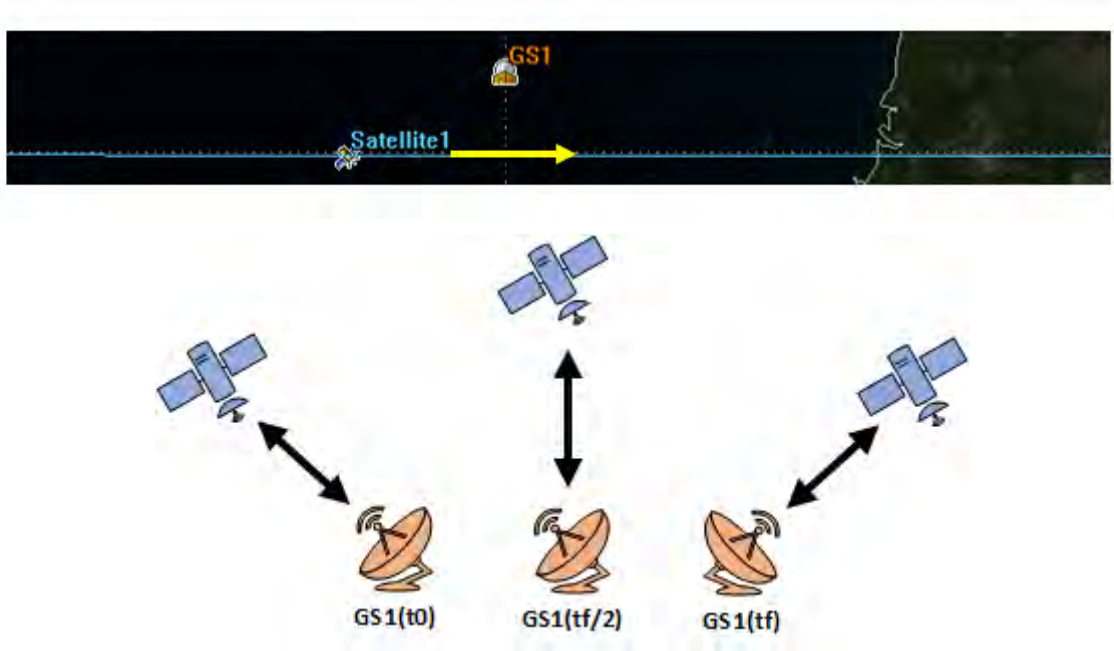


Figure 33. One satellite, one ground station simulation.

The trial runs were performed with a satellite in realistic, though idealized orbits. The orbit chosen for these test cases was circular at 400 km altitude along the equator at 0° inclination. To minimize the effect of Earth's rotation the GS1 facility was placed at sea level altitude, located at 2°N, 0°E to generate predictable passes that were not directly overhead. Perfectly overhead contacts are less frequent in the real world and could require specific consideration due to the potentially excessive rotor velocities required to maintain contact with the satellite at the point of closest approach (unless alternative rotor configurations are used such as X/Y, an additional tilt axis, etc., which constitutes future research).

Figure 34 shows the simulation output when tracking a single satellite. The scenario duration is approximately five minutes and corresponds to AOS and LOS elevation angles

of 10°. The benefit plot of Figure 34 shows the idealized link margin (blue) derived from Equation (26) and is a function of the slant range distance in Figure 34. As the distance to the spacecraft decreases, the signal strength increases and the potential for a successful contact increases. Put differently, the benefit to the satellite's mission is proportional to the signal strength when data downlinks are the primary objective. The satellite target is represented as a Gaussian BVF and the computed link margin is used to scale the BVF, derived in Equation (32). In the one-satellite case, the link margin and benefit curves scale linearly with respect to each other (by approximately a factor of five in this case). The scaling is a result of continuously tracking a single target, creating a constant Gaussian term from Equation (32). The only variability therefore comes from the link equation. Figure 35 shows the animated results as generated by the software implementation as a simple progression.

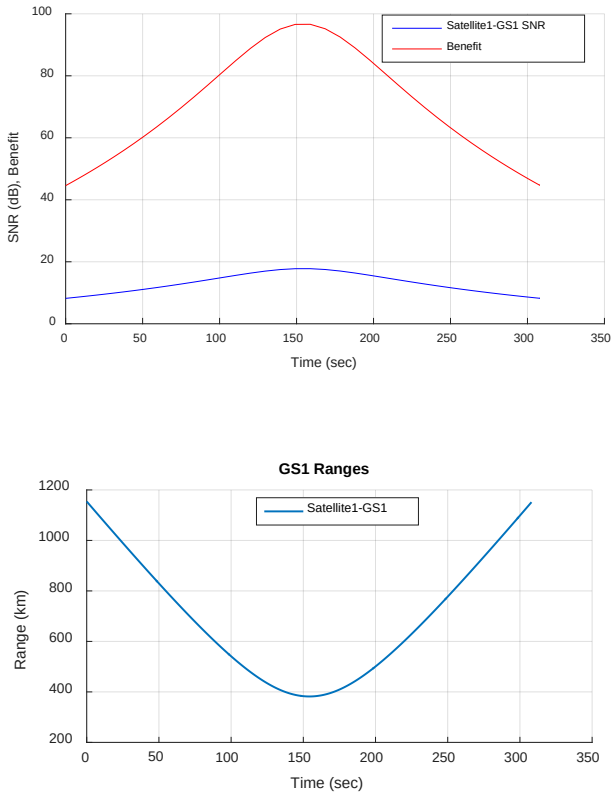
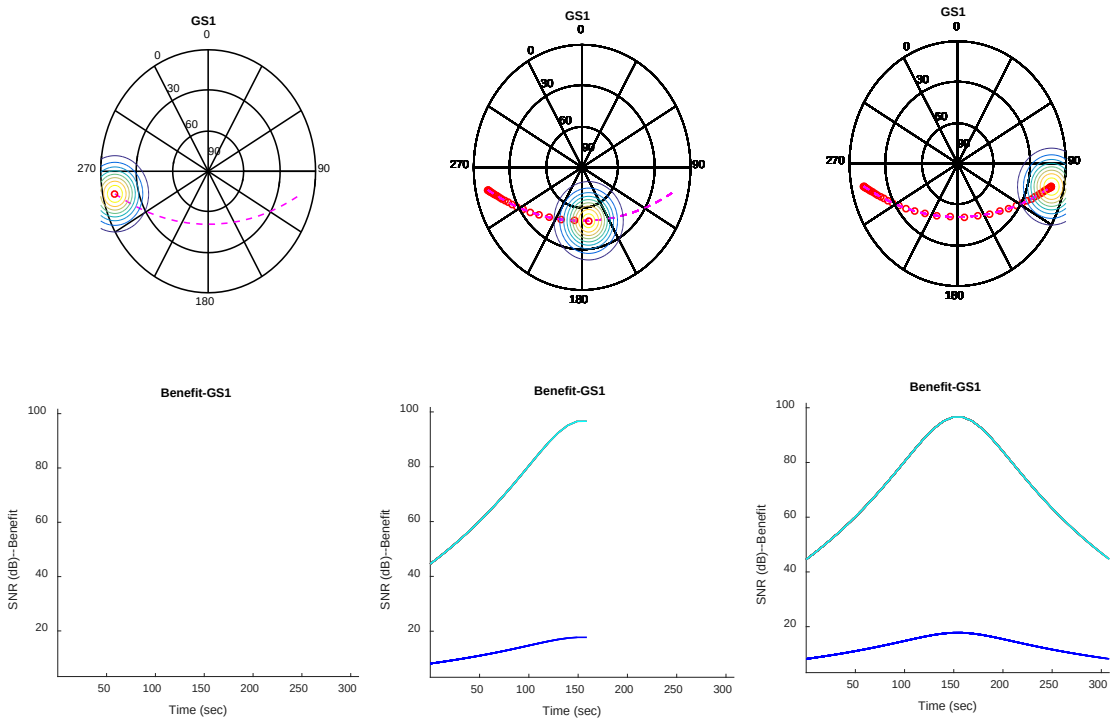


Figure 34. Benefit plots (top) and slant range distance to ground station (bottom).



Satellite target Gaussian denoted as contour plot. Scenario start at left and progresses to the right. Antenna position shown as red circle. Computed benefit from objective function Equation (32) shown in cyan, idealized link margin of Equation (26) in blue.

Figure 35. Animated results from simulation, with progression from left to right.

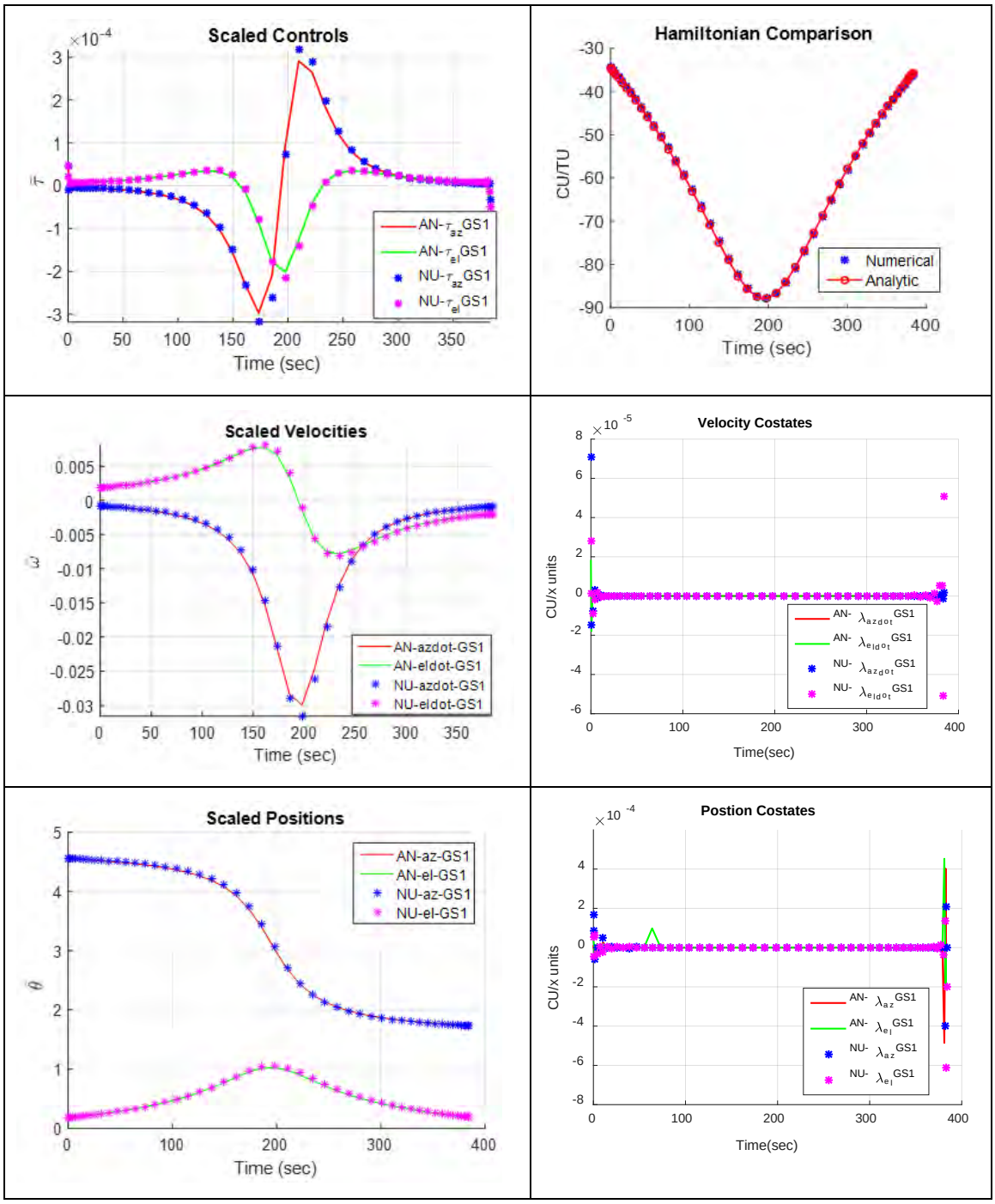
Figure 35 illustrates a two-dimensional polar plot corresponding to azimuth and elevation angles (described in Chapter II) with respect to the GS1 ground station. The satellite target Gaussian is shown as a contour plot. The center of the Gaussian represents the greatest value, and thus the brightest color. Its overall intensity is scaled with the idealized link margin curve. The antenna position is denoted as the red circle and is a function of the control torques, velocities, and positions computed by the software. Using the objective function defined in Equation (32), the greatest value gained at any point in time is when the distance of the center of the satellite's Gaussian and antenna position are minimized. Since there is only a single target to track, the antenna's red circle follows the Gaussian for the entire duration of the track, maximizing the benefit value curve. For reference, this simulation used $\sigma_x, \sigma_y = 0.15$ to size the Gaussian footprint and a factor of two scaling for the link margin. Scaling details will be discussed later in this section.

Two tests were conducted for the initial V&V analysis. The first test directly compares the analytic and numerical models, demonstrating that the optimization routines are agnostic regarding the origin of the satellite and ground station position data. The data can equally come from a custom-developed two-body propagator with known kinematics, or a commercial solution (STK) with its realism settings reduced to match those of the custom propagator (spherical Earth and circular orbit). Next, the double integrator with torque smoothing is shown to improve the performance of the numerical model. The double integrator with torque smoothing and numerical propagator will be the basis of more complex simulations in Chapter IV and real-world applications in Chapter V.

1. Analytic vs. Numerical Orbit Propagation

The problem formulation, shown in Equations (33) and (34), was implemented with two separate data sources. By choosing the same orbit, ground station location, and start/stop times, a direct comparison could be made between the two propagator inputs. Analyzing outputs of the simulation, illustrated in 0, allows for evaluation of parameters directly and indirectly associated with the optimal control problem.

Identical scaling factors were used for both cases and in each simulation. The position and velocity values were scaled 1:1 with real-world units. The torque controls were scaled by a factor of 10^{-5}. The link margin Gaussian was scaled by a factor of two, and Gaussian sizes were assigned values of $\sigma_x, \sigma_y = 0.15$, creating the footprint shown in Figure 35. These factors were found to generate accurate solutions with short computation times, shown in 0. The associated position costate values computed by the solver are very small, on the order of 10^{-5}, due to the torque control scaling. Ideally, the state and costate terms would be scaled to within an order of magnitude, indicating additional scaling considerations may be necessary. In general, scaling and balancing is important for increasingly complex problems. DIDO (or similar optimization packages) often require scaled designer units to ease computational burdens and decrease processing time [73], [94], [95].



Hamiltonian shown as function of cost units (CU)/ time units (TU). Costates presented as a function of CU/ state (x) units.

Figure 36. Optimal control solutions from analytic (AN) and numerical (NU) trials for one satellite and GS1 station with identical orbits and start/stop times.

The solution's Hamiltonian outputs are inversely proportional to the benefit plots shown in Figure 34 and Figure 35. The objective functions in the problem formulation shown in Equation (34) takes the BVF developed in Equation (32) and explicitly multiplies by -1, as covered earlier in the chapter. Maximizing the benefit can be equivalently stated as minimizing the cost. 0 shows a Hamiltonian that reaches its minimum at the point of closest approach for the satellite, indicating the potential for maximum data transfer. The time-varying Hamiltonian is indicative of a changing running cost, as developed in the Hamiltonian Evolution Equation (46). Additionally, DIDO's *dual* outputs show a relationship conforming to Equation (51). Both analytic and numerical solutions generate identical Hamiltonian curves, further confirming interchangeability between the two propagation methods.

The next V&V check involves using the KKT conditions from Equation (42). The KKT multiplier equalities in these equations were plotted and compared, as shown in Figure 37, and are directly related to the torque control inputs. The torque values were constrained to the vendor-specified values of 6 N-m and 14 N-m for the azimuth and elevation axes respectively, or $6\text{x}10^{-4}$ and $14\text{x}10^{-4}$ when scaled by 10^{-5}. Since the torque values in 0 did not reach their limits, μ is expected to be zero. Given the equalities established in Equation (42), λ must also be zero. This corresponds to the KKT outputs plotted in Figure 37 where both μ and λ terms are on the order of 10^{-7}, or effectively zero. The pattern continues when evaluating the initial and terminal transversality conditions from Equation (49). The position costates in 0 at the beginning and end of the scenario are also zero in practical terms.

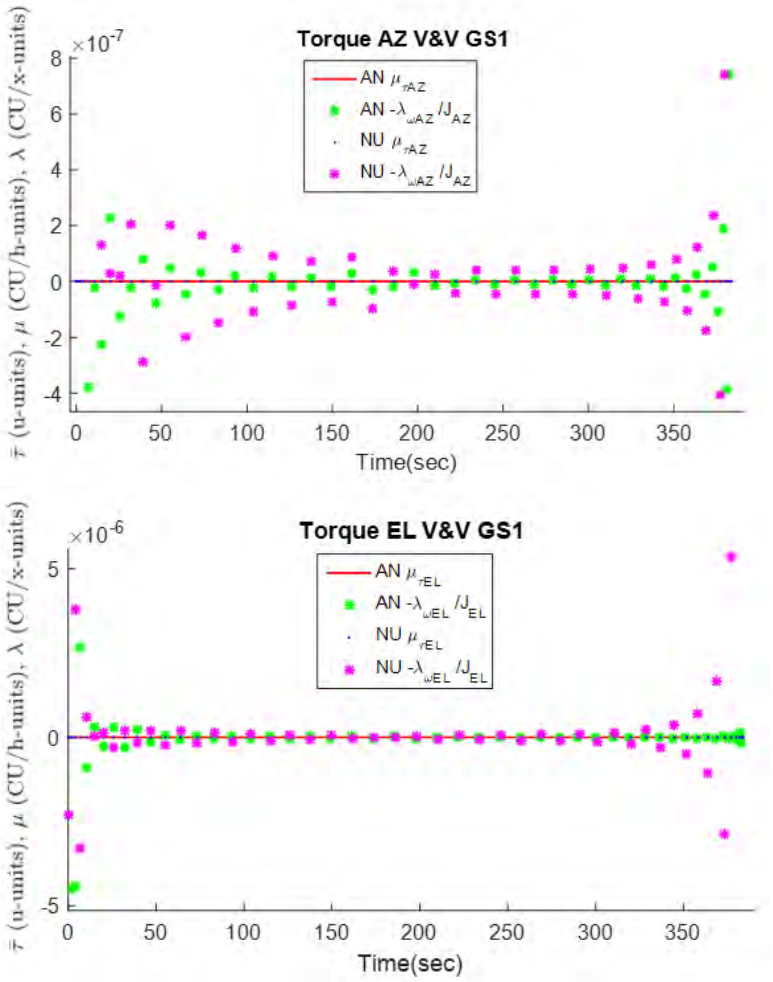


Figure 37. Torque control V&V plots for analytic and numerical double integrator models.

2. Numerical Propagator Double Integrator with Torque Smoothing

The final V&V case for the simplest single satellite/single ground station model involves the torque smoothing variation of the initial problem formulation. Figure 38 shows the outputs from the simulation. All scaling parameters remained unchanged from the previous example, and the torque penalty was implemented without any additional scaling in the cost function.

The positions and velocities match expected values as the antenna tracked the target. As with the previous example, the costates remained very small at near zero levels.

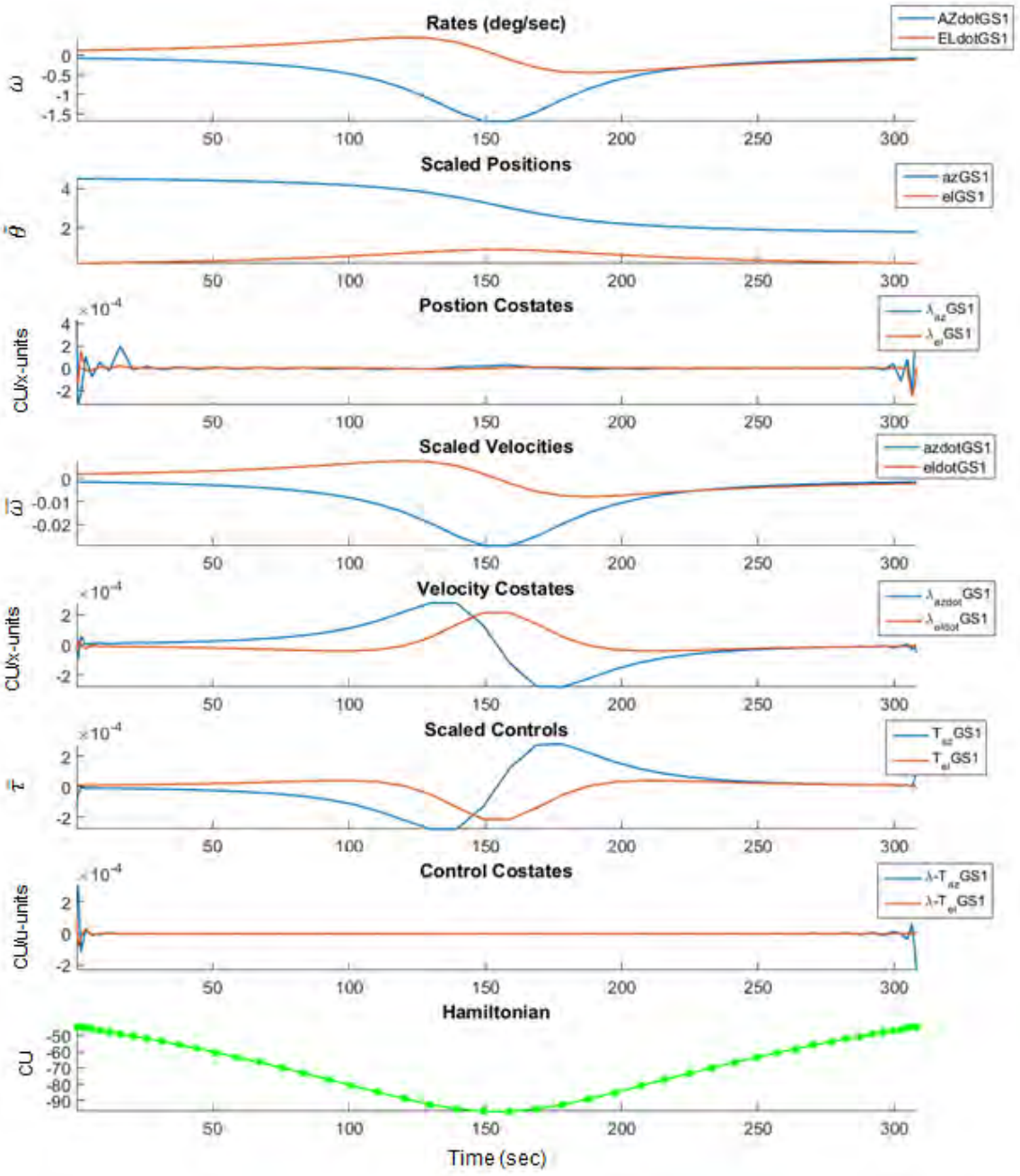


Figure 38. Outputs from double integrator with torque smoothing.

Likewise, the torque inputs remained below their maximum constraints. The full benefit of the torque smoothing addition in the cost function is not readily visible in the simplest case as there is only one target available for tracking though additional comparisons are made in Chapter IV.

The Hamiltonian behavior is nearly identical to the previous two cases. The addition of two torque terms has little effect on the Hamiltonian due to their small values. The Gaussian BVF remains the dominant expression. Its time-varying behavior is expected from the Hamiltonian Evolution Equations derived in the previous examples.

The KKT conditions change slightly due to the additional torque terms in Equation (56) but remain near zero, practically speaking. Figure 39 shows the scaled torque and associated KKT conditions.

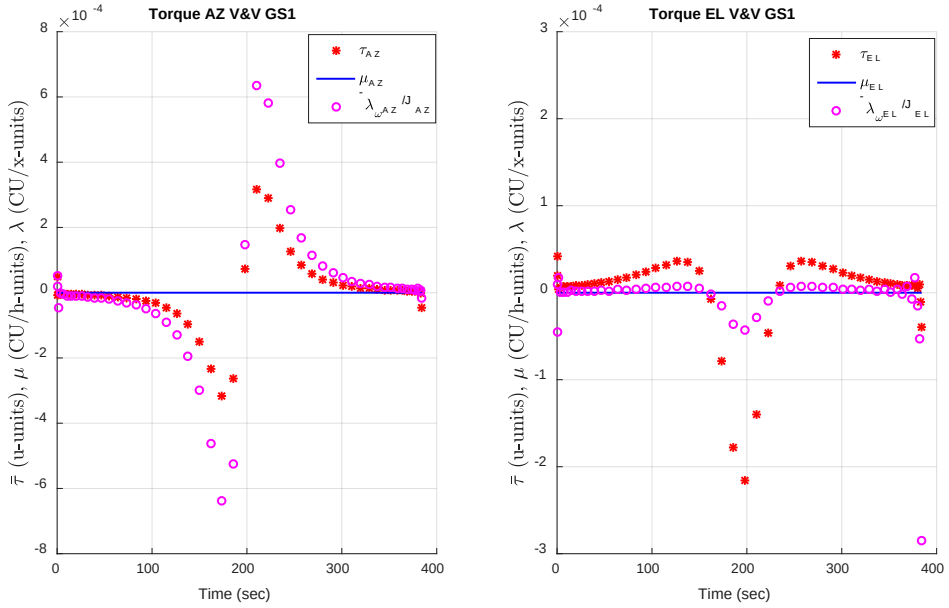


Figure 39. Torque control KKT verification.

Figure 40 plots the velocity values when compared to its associated KKT covectors. We see the same behavior as in the previous cases since the additional torques do not alter the velocity expressions in Equation (56). Lastly, the transversality conditions are identical

to the initial double integrator case due to Equation (51). Both initial and final λ values are zero as observed in Figure 38.

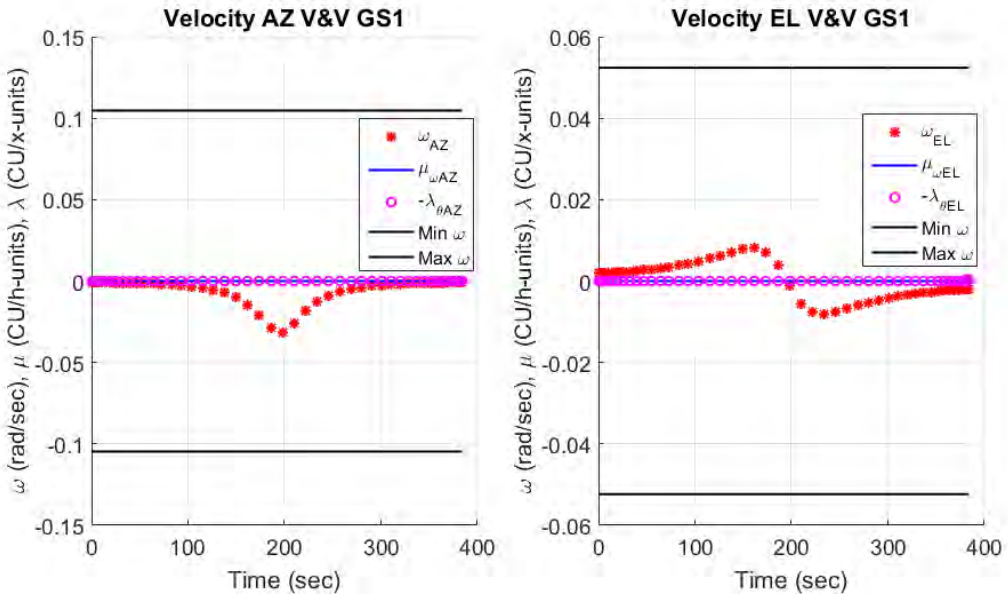


Figure 40. Velocity state KKT verification for double integrator with smoothing.

D. CHAPTER SUMMARY

This chapter presented a systematic derivation and V&V of the optimal control problem formulation addressing satellite/ ground station targeting. The models leveraged physical behaviors outlined in Chapter II and applied optimal control fundamentals to arrive at three solutions of increasing complexity. The solutions were implemented in MATLAB and solved with the DIDO optimal control toolbox. The outputs provided by the computation were compared to theoretical values, which converged to provide necessary conditions for adhering to Pontryagin's Principle. The following chapter applies the software implementation to scenarios of increasing satellite and ground station quantities.

IV. MULTI-SATELLITE SCENARIOS

This chapter expands on the implementation of the problem formulation shown in Chapter III by demonstrating the generation of valid solutions to the simple problems of increasing complexity described in Chapter I.D. This will then enable evaluation of more complex real-world problems that may not have intuitive solutions in Chapter V.

A. TWO SATELLITES, ONE GROUND STATION

Complexity of the simulation runs increases incrementally from the one satellite case covered in Chapter III by adding a second satellite and bounding AOS/LOS times to maintain a 10° elevation threshold for line-of-sight communications. Figure 41 illustrates the scenario.



Figure 41. Two satellite, one ground station scenario. Top-down ground track view (left) and reference from Chapter I (right). Satellites move from left to right.

The BVF weights were kept equal for both satellites to demonstrate a predictable transition as the station tracked the two objects. Problem scaling remained identical to the single satellite case shown in Chapter III. With these scaling constants, the addition of a second satellite still produced a valid solution and increased computation time by approximately one minute. Figure 42 is a top-down polar plot view showing the two targets transitioning past the station.

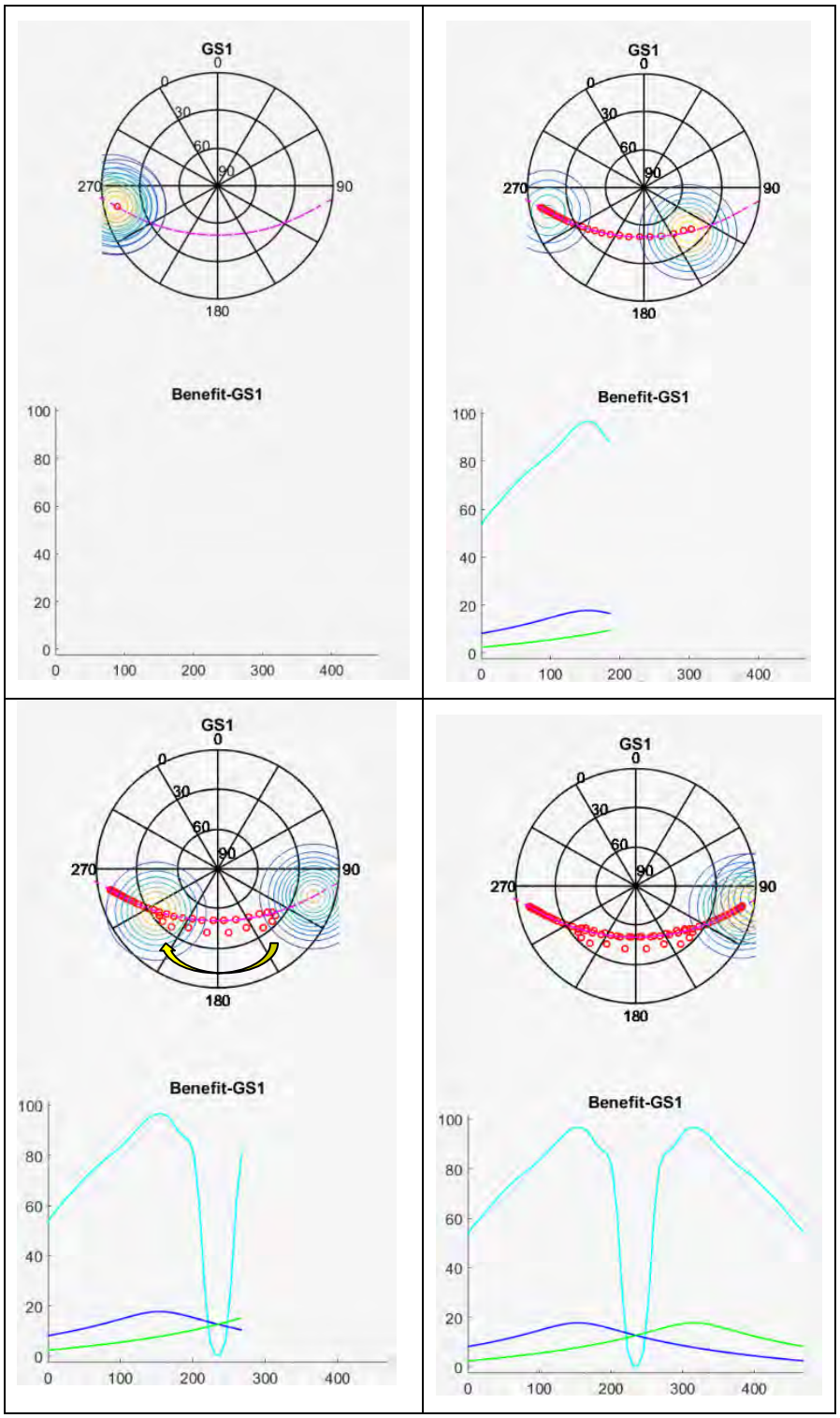


Figure 42. Two-satellite animation progression starting at top left. Satellite1 link margin in blue (dB), Satellite2 link margin in green (dB), combined benefit in cyan (CU).

In Figure 42, the satellites are shown as equally weighted Gaussian contour plots with the red circle indicating the instantaneous position of the antenna. The total benefit (cyan) is computed as the sum of individual Gaussian contributions when compared to the commanded position of the antenna. The blue and green curves show the idealized link margin used for scaling the peak of the Gaussian for each satellite. Figure 43 shows a detailed view of the benefit value curves and the idealized reference case from Chapter I.

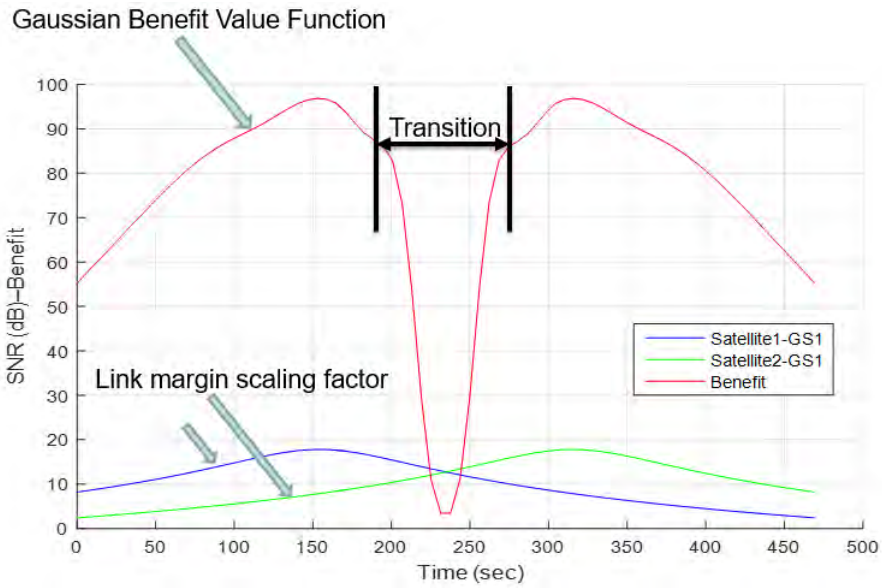


Figure 43. Benefit value curves for two satellite/one ground station scenario.

The BVF accrued value as the antenna tracked the first satellite up until such a point where it became more beneficial overall to break off and transition to the second. The temporary drop in accumulated benefit was necessary to continue accruing additional benefit in the second half of the scenario. The results match the idealized case for equal benefit discussed in Chapter I and provide a valuable benchmark for the feasibility of utilizing optimal control fundamentals when communicating with multiple space-based assets by means of ground-based antennas. Next, the two satellite/one ground station solution contrasts the performance between two methods (double integrator and double integrator with smoothing).

1. Double Integrator

Solving the two satellite/single ground station problem with the initial formulation described in Chapter III generates a valid solution with the transition point initiating just before the scenario mid-point. Figure 44 shows the resulting position and velocity trajectories.

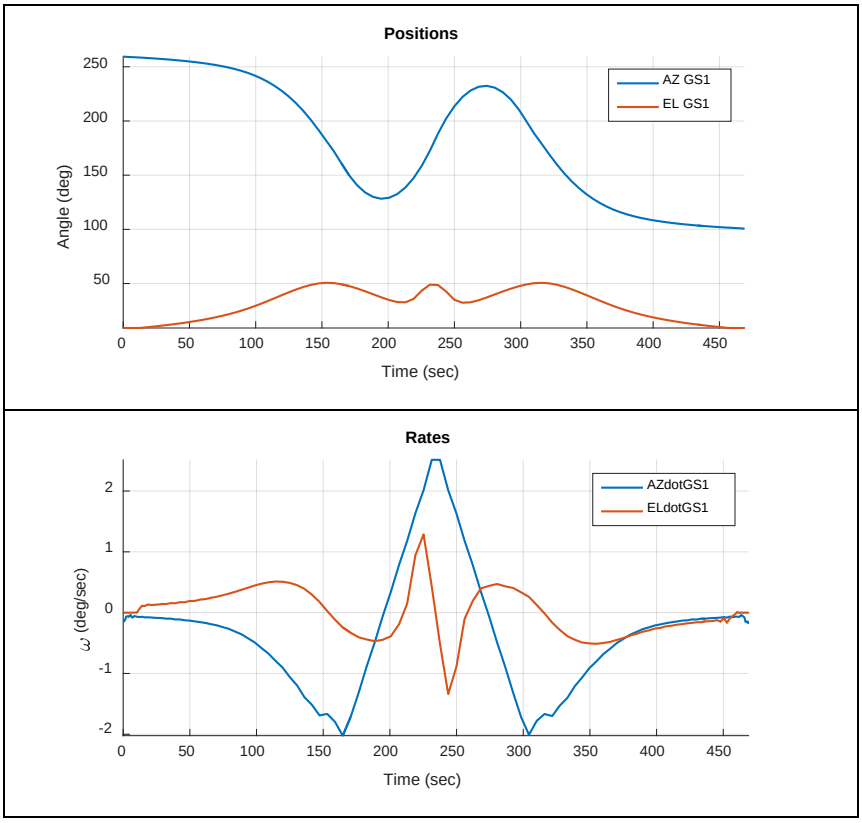


Figure 44. Angular positions and velocities for double integrator problem.

This model generates positions and velocities that show smooth trajectories with a predictable transition between the two targets. The transition occurred automatically to maximize the scenario benefit. This solution may be sufficient if the real-world antenna system controller accepted inputs of velocity or angle trajectories (as many do). If explicitly controlling the rotor torques applied to each degree of freedom, the control trajectory should minimize unnecessary control inputs to prevent jitter that can lead to pointing

inaccuracies and mechanical wear. The double integrator model in this example generates control torques that are oscillatory, as shown in Figure 45.

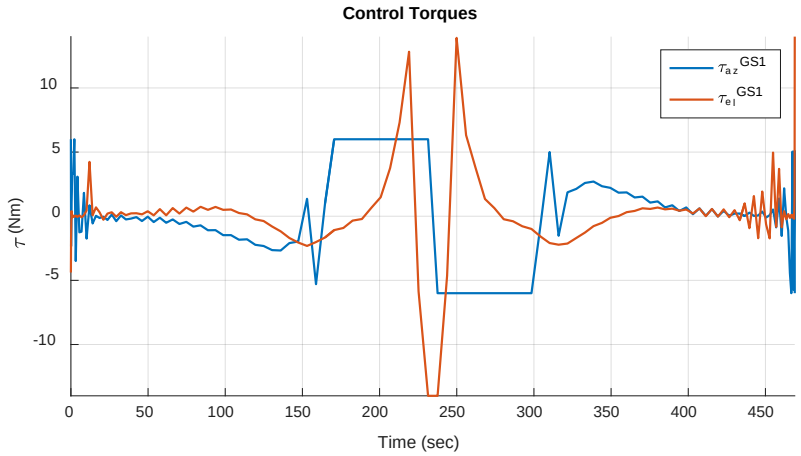


Figure 45. Double integrator torque controls, two satellites/one ground station.

The overall torque values in Figure 45 reach their vendor-specified maximum values of 6 and 14 N-m for azimuth and elevation, respectively, during the transition between targets. Since there was no penalty for additional control inputs, the torques were free to assume any value to satisfy the problem formulation within the constraints.

The transition between satellites generates an interesting behavior. Since there is little benefit available to collect between target Gaussians, the Hamiltonian temporarily increases in value. To compensate for this increase and maintain conformance to the KKT conditions, the costates for velocity and position also experience a sudden increase in amplitude, potentially orders of magnitude greater than in the rest of the scenario, illustrated in Figure 46.

The temporary increase in costate value can lead to scaling challenges when generating a numerical solution in DIDO. Optimization packages such as DIDO often require designer units to bring states and controls to within approximately the same orders of magnitude for facilitating computation. References [73], [94], and [105] cover

techniques for properly choosing designer scaling units. This example utilized torque scaled by a factor of 10^{-5}, link margin scaled by two, and Gaussian footprint sizes defined by $\sigma_x, \sigma_y = 0.25$. In general, being able to utilize the optimization techniques developed in this dissertation for large quantities of targets requires consideration of these scaling issues to enable accurate solutions with reduced processing time.

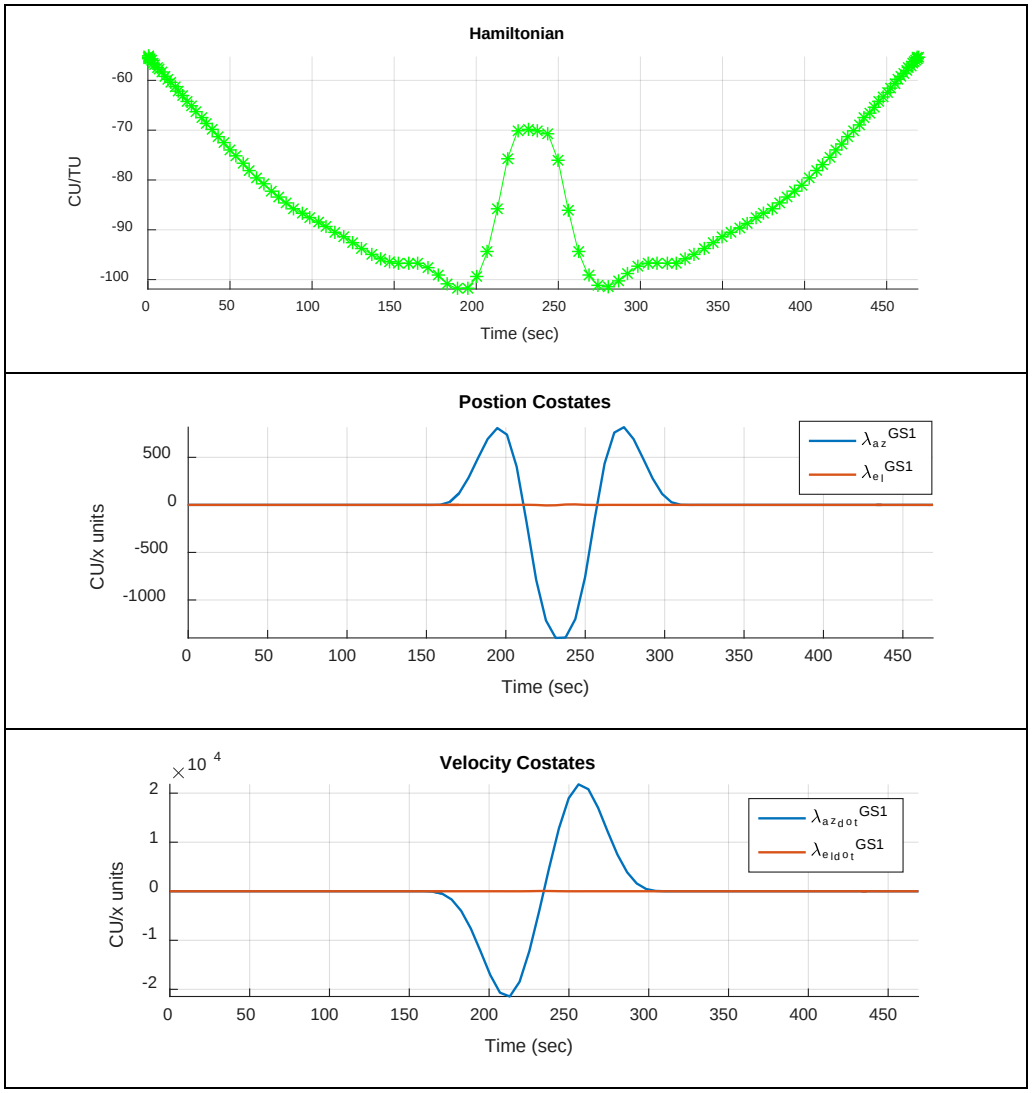


Figure 46. Hamiltonian and costates for double integrator solution.

2. Double Integrator with Torque Smoothing

The addition of torque smoothing terms in the cost function of Equation (54) reduces oscillations in the control and velocity solutions by penalizing extraneous torque inputs. This produces smoother trajectories with less potential for pointing losses and mechanical wear in the antenna system. Figure 47 contrasts the torque, velocity, and position trajectories for the two satellite/one ground station problem with and without this feature.

a.)

b.)

c.)

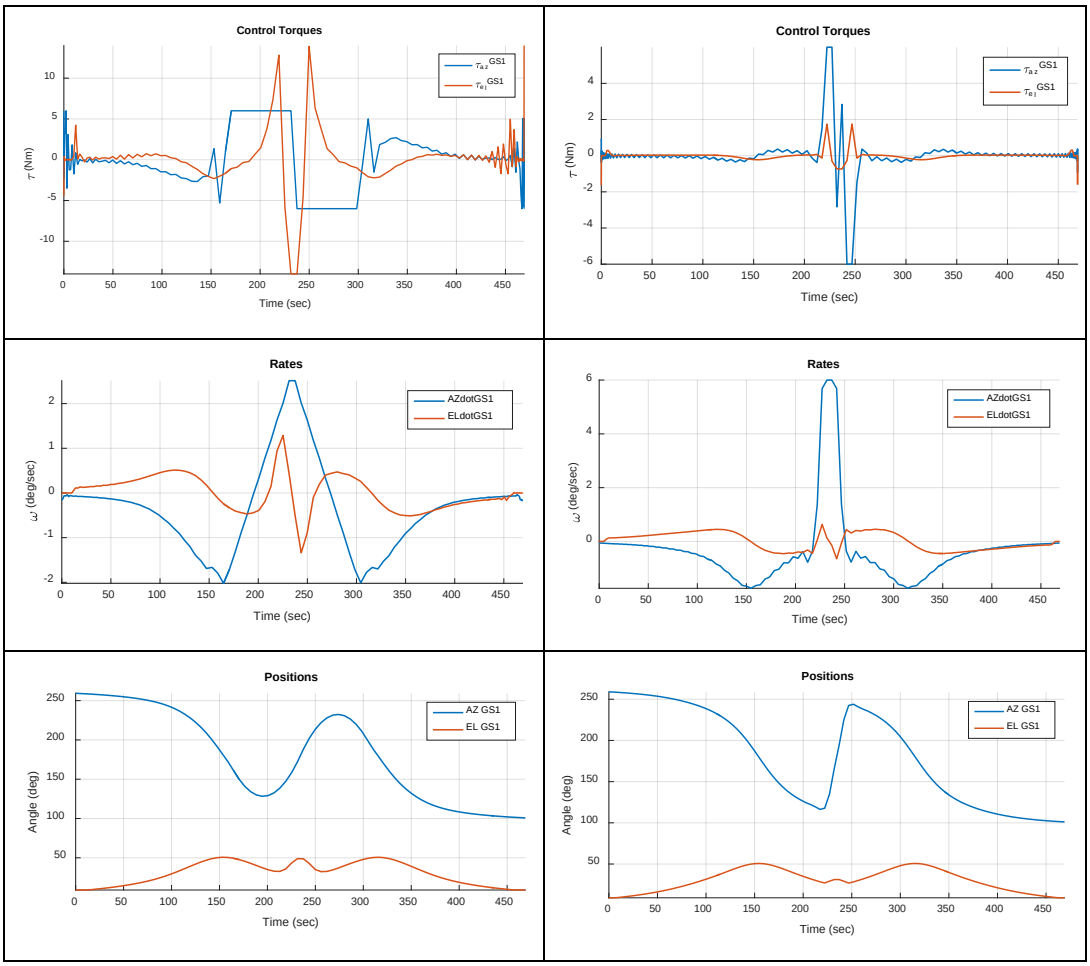


Figure 47. Double integrator without (left) and with torque smoothing (right), for two satellites/one ground station.

The trajectory oscillations in the controls and states diminished when comparing the torque smoothing solution with that of the standard double integrator, as shown in Figure 47. The torque smoothing solution limits the total cumulative torque expended, resulting in a shorter, higher amplitude control input at the transition. This creates a more abrupt transition between satellites and, as will be shown in the following section, increases accuracy by smoothing the trajectory during each tracking phase.

The smoothing effect was achieved by not only including the torque terms in the cost function, but also scaling them such that the torque penalty did not influence the Hamiltonian by too much or too little. The dominant term is typically the azimuthal degree of freedom given the larger ranges of motion; therefore, it was penalized more than the elevation axis. Much how scaling designer variables is an iterative process, so too is the torque penalty scaling method. Using an adaptation of the general torque smoothing cost function with scaled control penalties, as shown in Equation (58), obtained the best results.

$$J[x(\cdot),\mathrm{u}(\cdot),t_f] = -\int_{t_0}^{t_f} \left(\sum_{i=0}^{n_{SAT}} \sum_{j=0}^{n_{SITE}} \left(v_{ij} e^{-\left(\frac{(x_{iSAT}-x_{jSITE})^2}{2\sigma_x{}^2} + \frac{(y_{iSAT}-y_{jSITE})^2}{2\sigma_y{}^2} \right)} \right) \cdots + 0.1 * \tau_{AZ_{SITE}}(t)^2 + 0.01 * \tau_{EL_{SITE}}(t)^2 \right) dt \qquad (58)$$

Implementing this variant required rescaling some designer variables for use in the optimization routine. Torques for this application were rescaled from 10^{-5} to 10^{-4}, and the Gaussian footprint sizes were reduced from $\sigma_x, \sigma_y = 0.25$ to $\sigma_x, \sigma_y = 0.15$. Exploring more examples, the scaling terms converged on a range of acceptable values with minimized processing times.

3. Real-World Application

The purpose of the simulations in this chapter is to serve as a benchmark for developing more complex real-world scenarios. The outputs of these solutions should be applicable to real systems, and their accuracy should be sufficient to perform operations for real satellites. To serve this need, another useful evaluation involves the comparison of the generated antenna position outputs with those of the satellites as propagated by STK.

The propagated positons serve as the "ideal," and the solution outputs should fall within an acceptable tolerance for pointing accuracy as a function of time.

The torque-smoothed solution generated for the two satellite/one ground station problem utilized 150 nodes for a 470-second scenario. The MC3 ground stations that are involved in the real-world demonstration portion of this dissertation require one position update per second (as do many other systems); therefore, it is necessary to interpolate the 150 nodes into 470 discrete positions. MATLAB's *interp1* function effectively performed the interpolation. The function uses a lookup table approach by applying a cubic spline interpolation between node points of the antenna position trajectories. Figure 48 shows a plot of azimuth and elevation positions for the satellites and outputs of the DIDO optimization package.

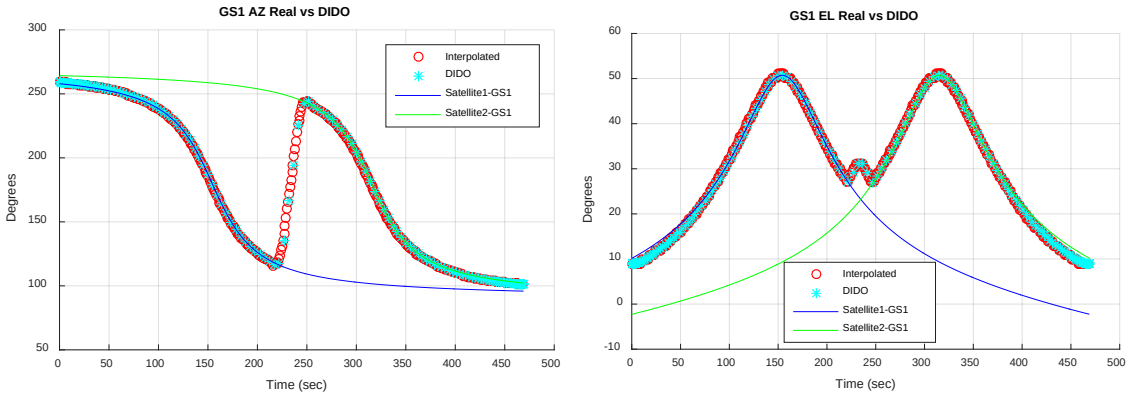


Figure 48. Satellite positions vs. DIDO solution and corresponding interpolated outputs for azimuth (left) and elevation (right) axes.

Predictably, the antenna tracks one satellites' azimuth/elevation positions until the transition point, after which it acquires the second. The solution includes a small increase in elevation during the transition phase to follow the gradient of the Gaussian and accrue additional value as it is acquiring the satellite. The benefit gained by taking the most expedient route perpendicular to the Gaussian contour outweighs the penalty for the extra deviation. To further evaluate the accuracy of the solution, the instantaneous difference was calculated between the solution and STK-propagated satellite positions, shown in

Figure 49. The evaluation includes solutions both with and without torque smoothing. The use of torque smoothing creates tighter pointing accuracy in this model, remaining within one degree of the propagated satellite position when the targets are overhead. This result further justifies the use of torque smoothing in the problem formulation.

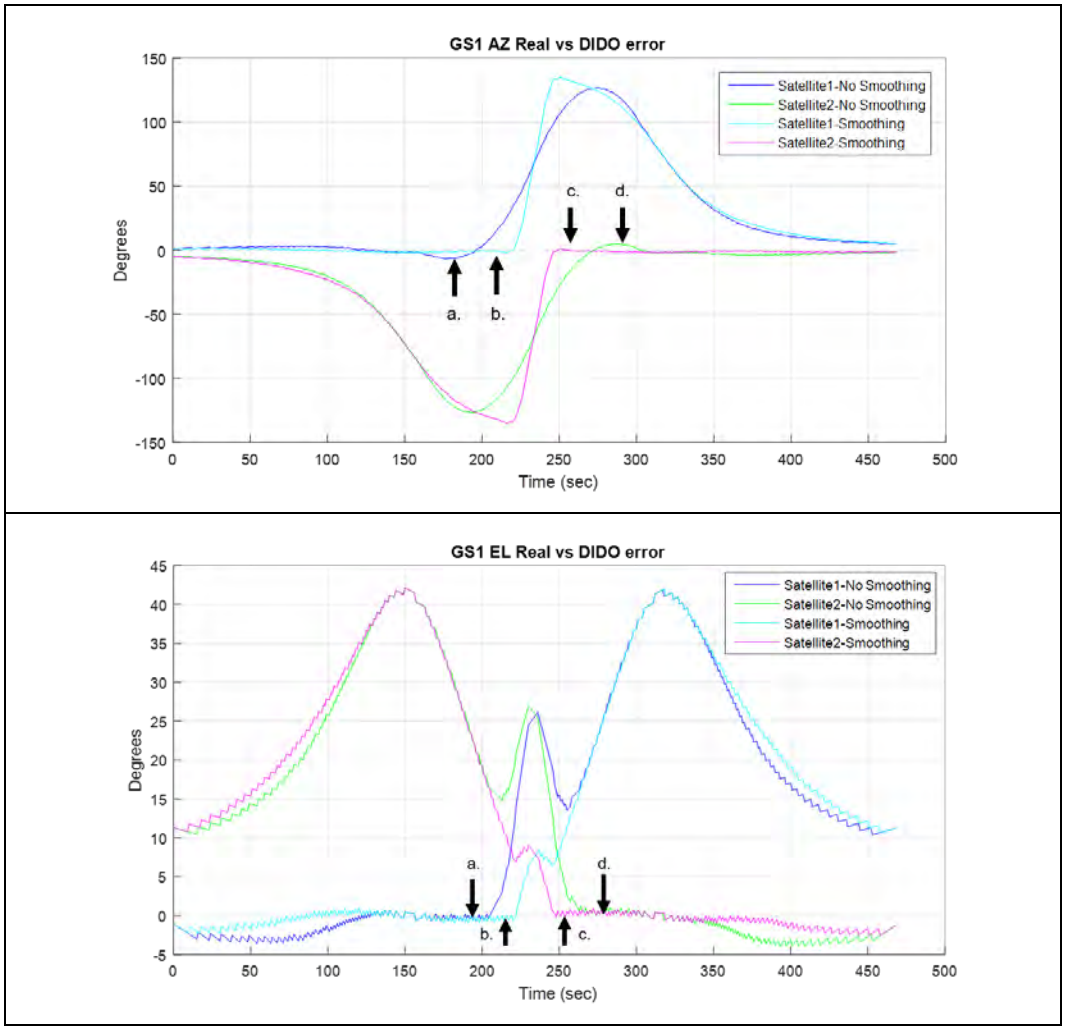


Figure 49. Antenna pointing accuracy from propagated satellite positions and solution outputs for azimuth (top) and elevation (bottom) degrees of freedom with and without torque smoothing.

Referring to Figure 49, the solution without smoothing breaks away from Satellite1 prematurely (denoted as *a.*) and intercepts Satellite2 after the trajectory of the smoothing

solution (denoted as *d.*). The smoothing solution constrains the transition to occur only when necessary, leaving Satellite1 at *b.* and intercepting Satellite2 at *d.*

Small deviations occur when both satellites are near the horizon in overlapping positions with respect to the ground station. The cost function performs a summation of each Gaussian (from Equation (54)), meaning that the position with most benefit for antenna pointing may be in between two peaks generating a higher summed benefit than any individual peak centered on a satellite. Two possibilities for mitigating this effect are to create smaller, tighter Gaussians that do not overlap or to incorporate additional constraints to effectively "turn off" the benefit contribution from a satellite not being actively tracked. The latter requires the use of a binary variable constraint, which remains as a future research topic that can leverage existing techniques [48].

4. Numerical Integration V&V

Having achieved smooth trajectories that are applicable to real-world systems, the generated control torques were integrated twice as ordinary differential equations (ODEs) using MATLAB's ODE45 solver to yield corresponding velocities and positions. These trajectories were then compared to the outputs produced by the optimization software to check the validity of the result, as shown in Figure 50.

The ODE45 solver leverages a Runge-Kutta method to integrate an ordinary differential equation with initial value. This is MATLAB's default solver, and its outputs should closely match the simulation outputs for a valid solution. Some discrepancies in the integration can exist, especially in cases where the function quickly changes direction. The satellite-tracking application can include such rapid changes in direction during the transition phase between satellites, as can be seen in Figure 50. The most pronounced drift in integration accuracy is evident in Figure 50 where elevation rates in row *a.* switch rapidly during transition, inducing integration errors at the end of the elevation position trajectory in row *b.* Despite small errors, the ODE45 integration exercise provides additional confidence in the validity of outputs generated by the software implementation. Additional information regarding numerical integration techniques for ODEs is highlighted here [96].

a.)

b.)

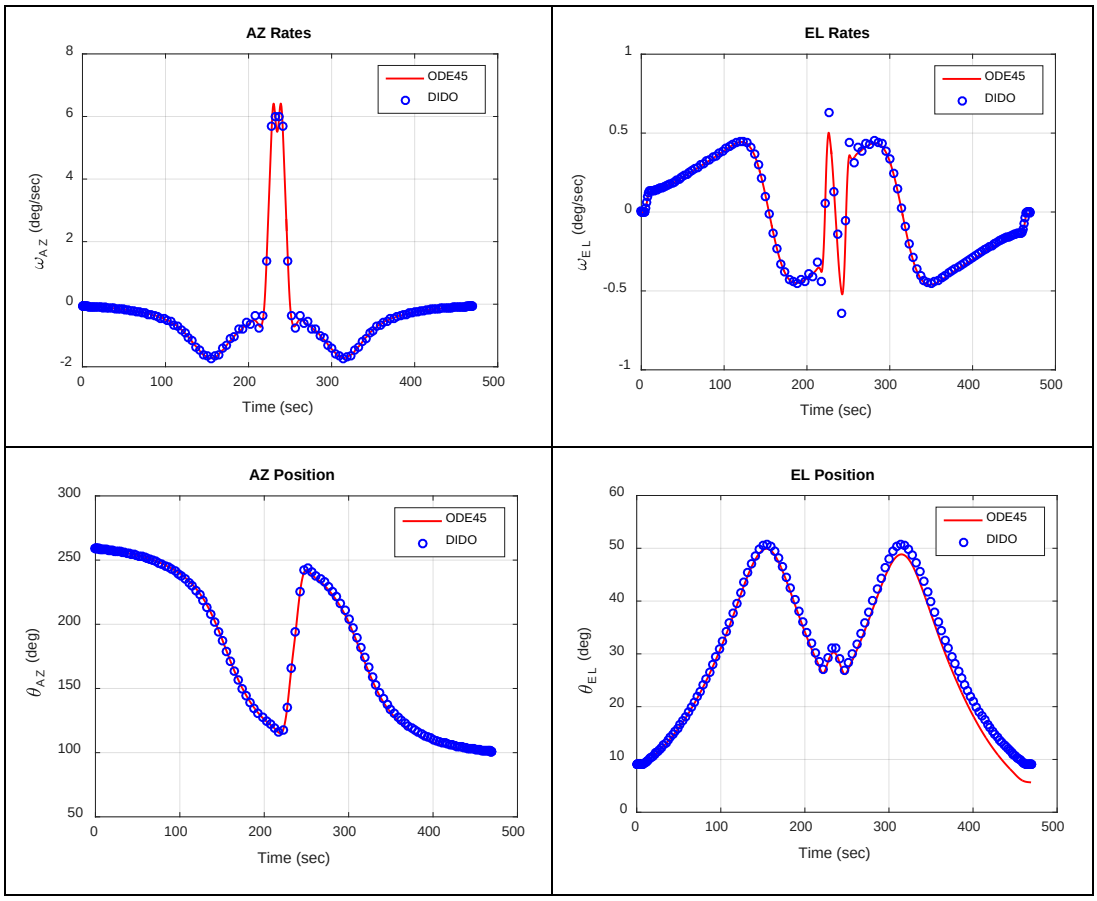


Figure 50. Velocity and position trajectory integration for azimuth (left column) and elevation (right column).

B. THREE SATELLITES, ONE GROUND STATION

Expanding on the above scenario, a third satellite was introduced with a larger offset to demonstrate the algorithm's indifference to satellite position and spacing in a predictable manner. Figure 51 shows the ground track.

The positions of Satellite1 and Satellite2 were left unchanged with respect to the previous example. The spacing between Satellite2 and Satellite3 is double that of Satellite1 and Satellite2 (true anomaly spacing is 20° and 10°, respectively). The circular orbits remained at 400 km altitude and 0° inclination.

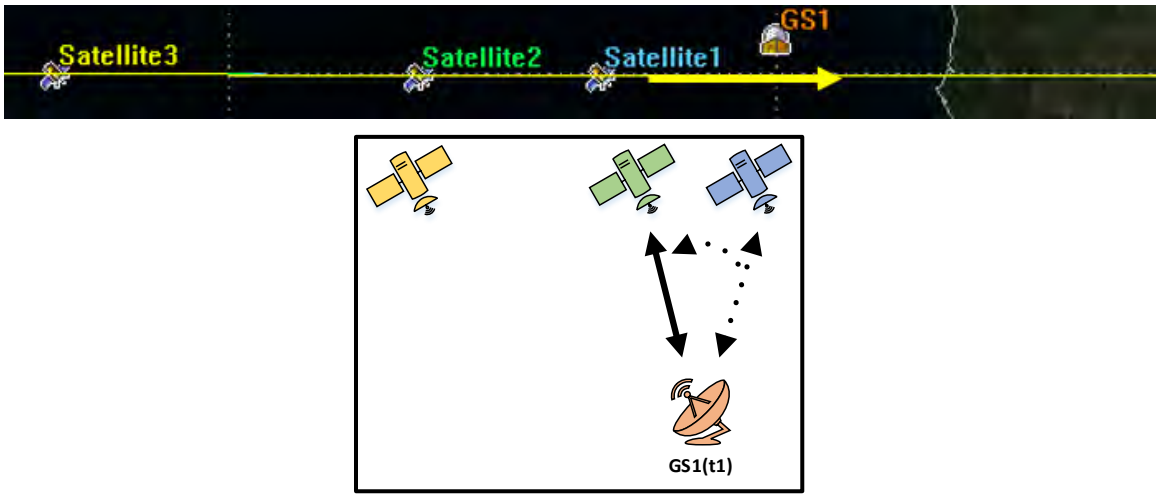


Figure 51. Three satellite/one ground station scenario.

Figure 52 shows the three-satellite animation as a progression with corresponding benefit functions. The antenna is directed to track Satellite1 until there is an opportune transition point for Satellite2. This transition happens at the same time as for the two-satellite scenario discussed in Section A due to the satellite positions remaining identical for the two test cases. Once GS1 acquires Satellite3 it tracks that satellite until completion of the scenario. Due to the additional spacing between Satellite2 and Satellite3, the total contact duration for Satellite3 is longer than for those of Satellite1 and Satellite2.

The scenario's benefit plots in Figure 53 show two transitions at the intersections of the individual link margin curves. The algorithm is indifferent to the spacing of the satellites and looks to maximize the cumulative time spent tracking the Gaussian targets. The drops in benefit are necessary in the transition between targets and are minimized given the rotation rates, position constraints, and available torques of the system. Figure 53 also shows another perspective for the multi-satellite solution for each joint angle of the ground station rotor.

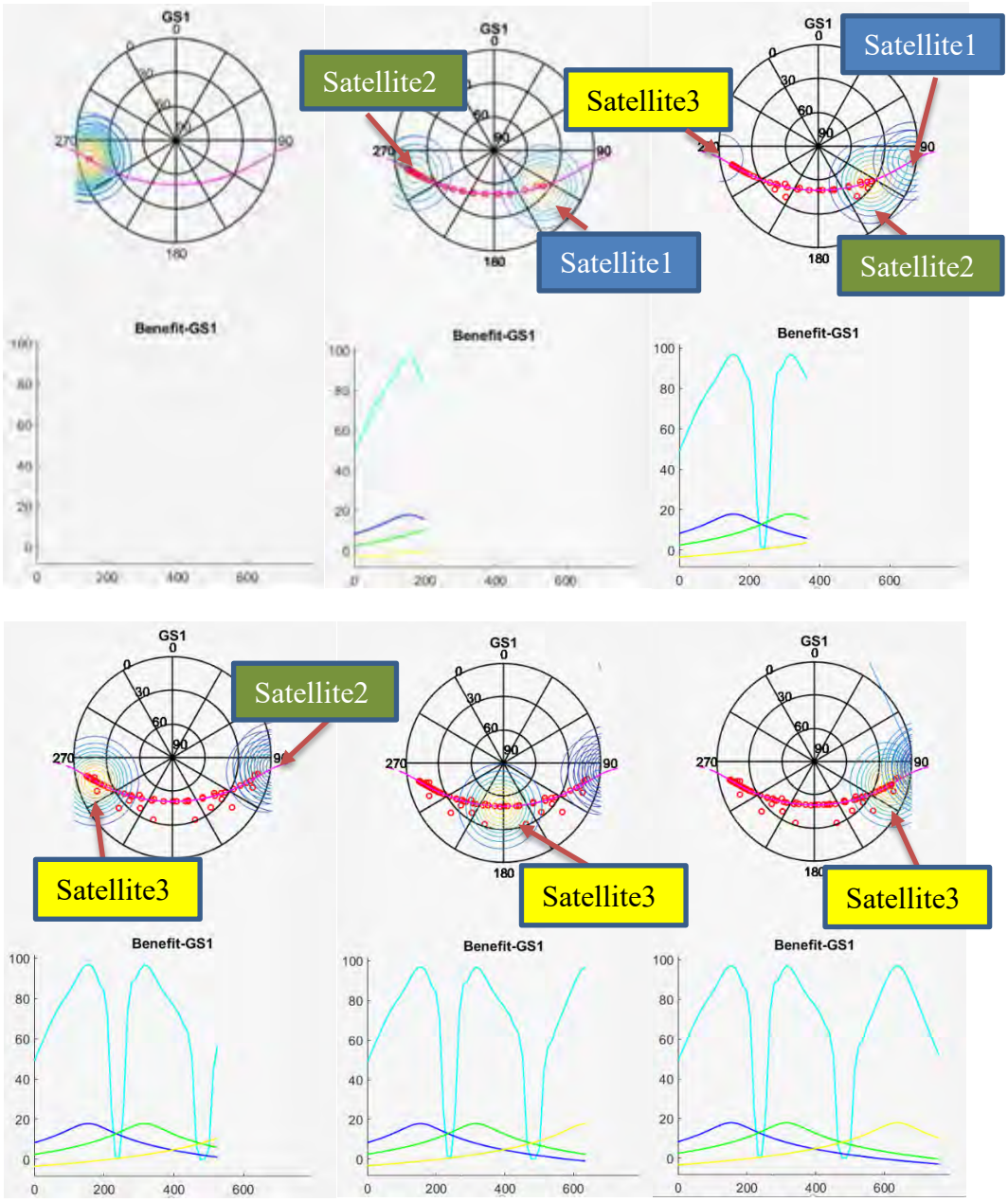
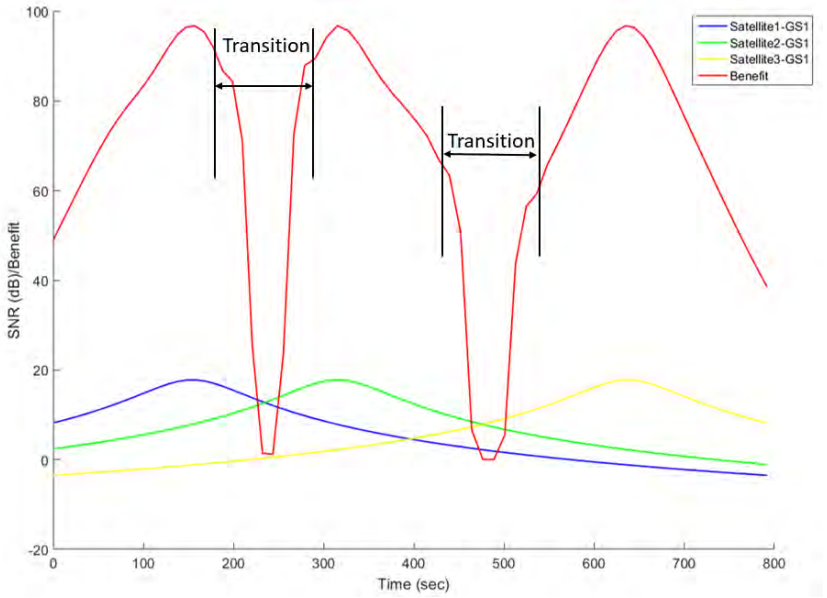
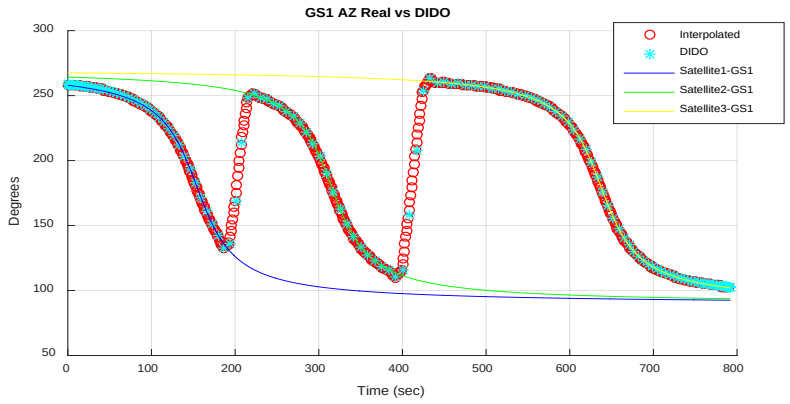


Figure 52. Three satellite animation plots, progressing from top left to bottom right.

a.)



b.)



c.)

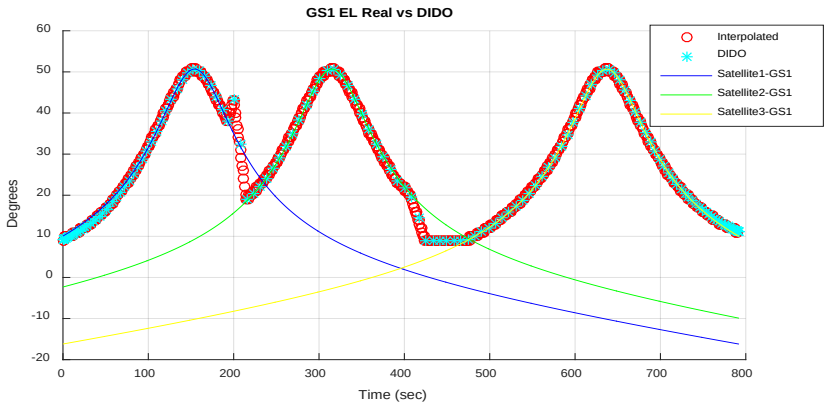


Figure 53. Benefit plot (*a.)*, azimuth (*b.*), and elevation (*c.*) plots and corresponding solution outputs.

Each target in Figure 53 has a local azimuth and elevation pairing with respect to the horizon at the ground station. The solution's rotor positions track each of these curves for segments corresponding to the points of closest approach. The azimuth and elevation trajectories smoothly transition from one target to another. Two small elevation excursions near the transition phases are byproducts of an objective function that does not penalize additional movements enough. The excursions correspond to the trajectory favoring movement along the perpendicular gradient of the Gaussian curve during transition. Since the azimuth trajectory is the dominant term during the transition, the elevation remains momentarily unconstrained and experiences a larger dither before resuming its track. Rescaling the torque penalties in the objective function further reduces these small excursions, if desired.

This nearly 13-minute simulation required 150 nodes to compute a solution with resolution sufficient for interpolation. The solution node points were then interpolated to produce the position pairings shown in Figure 53 at one-second intervals. This process allowed for integration into the MC3 system, as the real-world antennas require frequent position updates to maintain acceptable pointing accuracy when tracking the overhead satellites.

The final figure of merit for the three satellite, one ground station case involves the evaluation of the generated controls, states, costates, and Hamiltonian, presented in Figure 54.

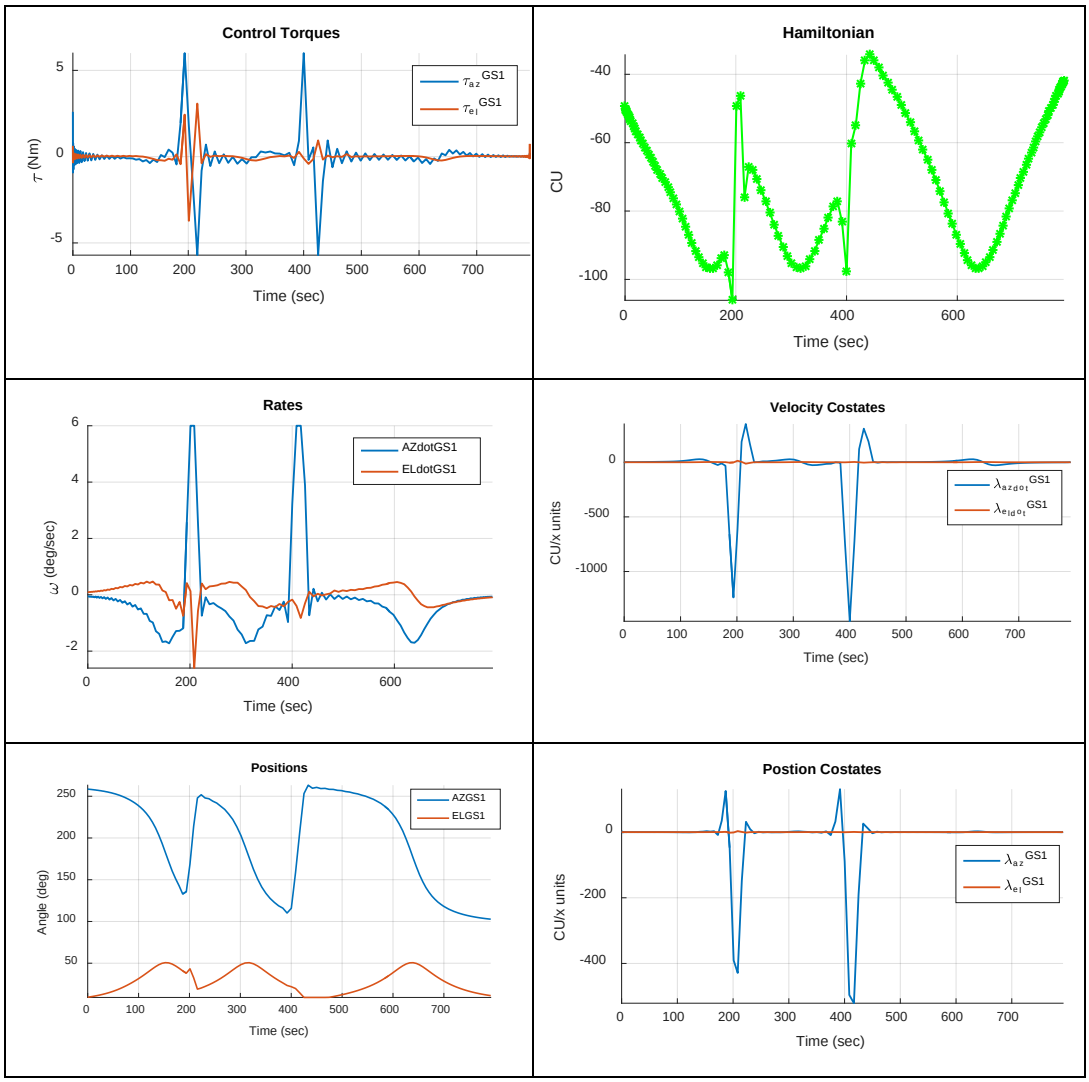


Figure 54. Controls, states, costates, and Hamiltonian for three-satellite problem.

The main evaluation for Figure 54 is to check whether controls and states are approximately within an order of magnitude with their corresponding costates. This qualitative rule-of-thumb helps assess the scaling and balancing used for a numerical solution. When compared to the simplest one-satellite case in Chapter III.C, Figure 54 shows that the states and corresponding costates (particularly angular velocity) are beginning to diverge. This trend continues with the addition of more satellites. Handling

cases of many hundreds of satellites will require the use of dynamic scaling to generate feasible solutions with reasonable processing times.

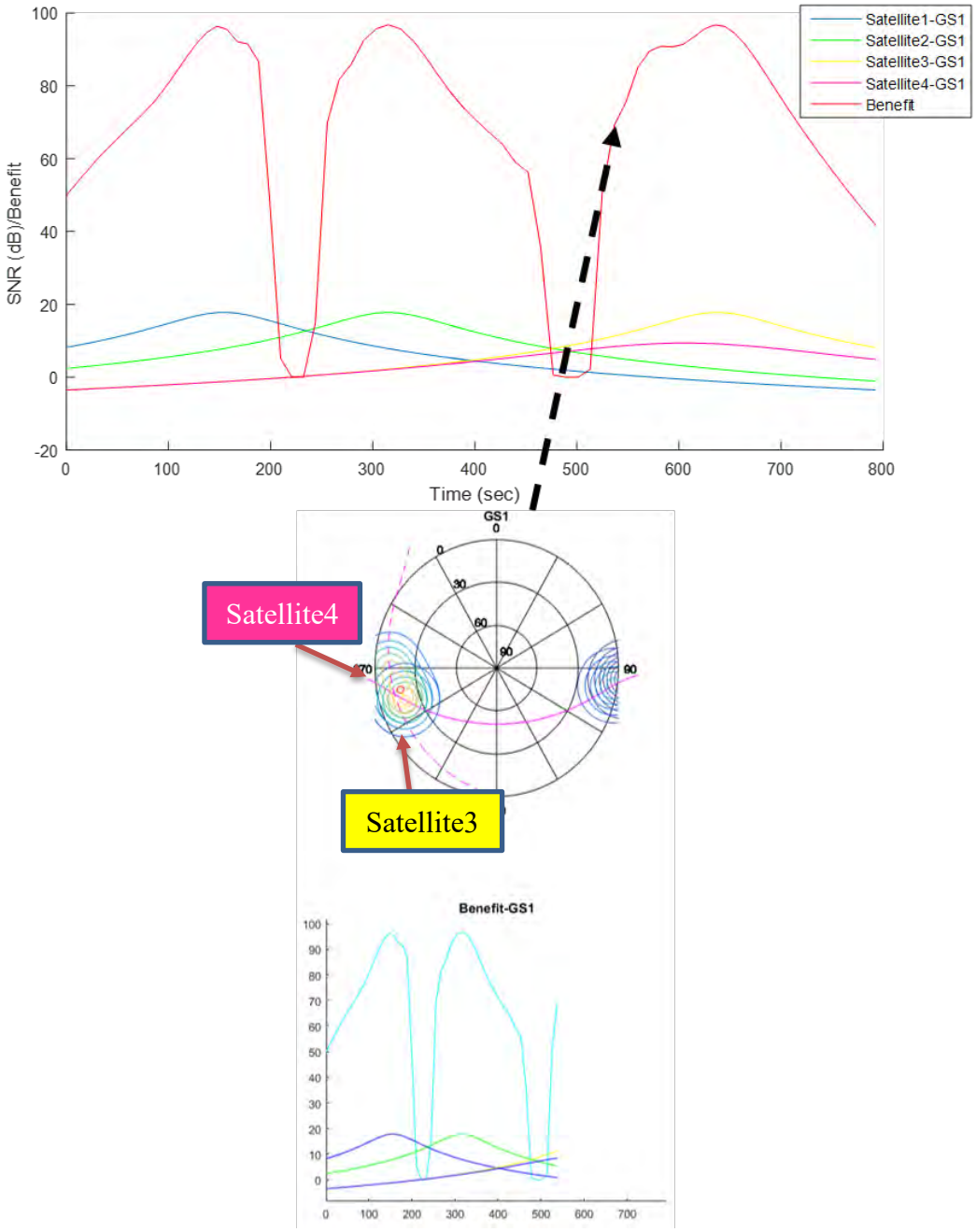
C. FOUR SATELLITES, ONE GROUND STATION

A fourth satellite was added to the three-satellite scenario to test how well the algorithm ignores targets that do not provide an increase in benefit. Satellite4, shown in Figure 55, was positioned in a different orbit such that it would approach closely enough to GS1 for consideration but should not supersede the benefit of tracking of Satellite3.



Figure 55. Four satellite, one ground station case. Satellite4 approaches GS1 (top) and crosses the Satellite3 ground track but remains further away, even at closest approach (bottom); satellite3 therefore remains GS1's target.

All scaling factors remained the same as the previous simulation. The resulting tracking solution, shown in Figure 56, did not differ significantly from that of the three-satellite case.



Red circle denotes instantaneous antenna position.

Figure 56. Four satellite/one ground station solution benefit curves and overlapping Gaussian snapshot (inset).

One small difference exists at the horizon for GS1 when Satellite3 and Satellite4 are near each other. Since the two Gaussians briefly overlapped, the solution temporarily placed the antenna between the two satellites to collect the most benefit. This created an extra ridge in the benefit plot (at approximately 570 seconds) when tracking Satellite3. In practice, this occurrence would have little impact to the communications link as both satellites were on the horizon at long distances from GS1, consequently with poor link margins. The tracking exclusively focuses on Satellite3 shortly after the overlapping event.

D. TWO SATELLITES, TWO GROUND STATIONS

A second ground station, GS2, is now introduced with the satellite positions remaining unchanged (Figure 57). This allows for a direct comparison to the single ground station results discussed above.

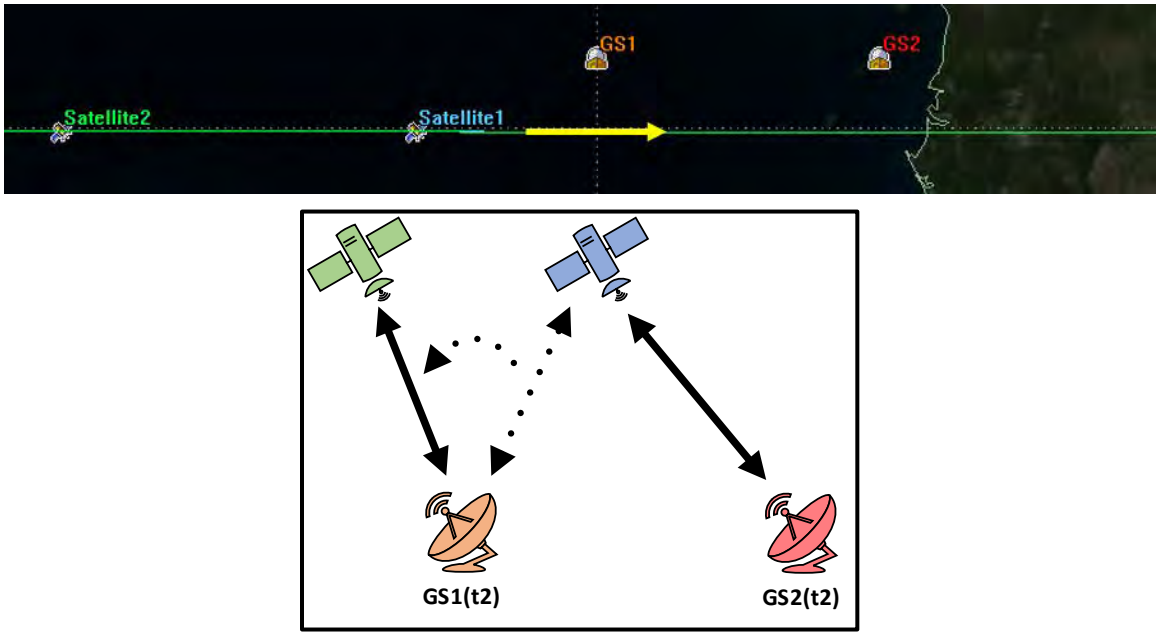


Figure 57. Two satellite, two ground station scenario.

GS2 was placed in a 2°N, 8°E position to compliment GS1 located at 2°N, 0°E. These experiments sought to increase the complexity of the ground segment in a predictable manner to further test the validity of the algorithms. The scenario began at the

time at which the elevation angle from GS1 was 10° to Satellite1. The scenario endpoint corresponded to the time at which the GS2 elevation angle for Satellite2 dropped below 10°. Figure 58 illustrates the scenario progression.

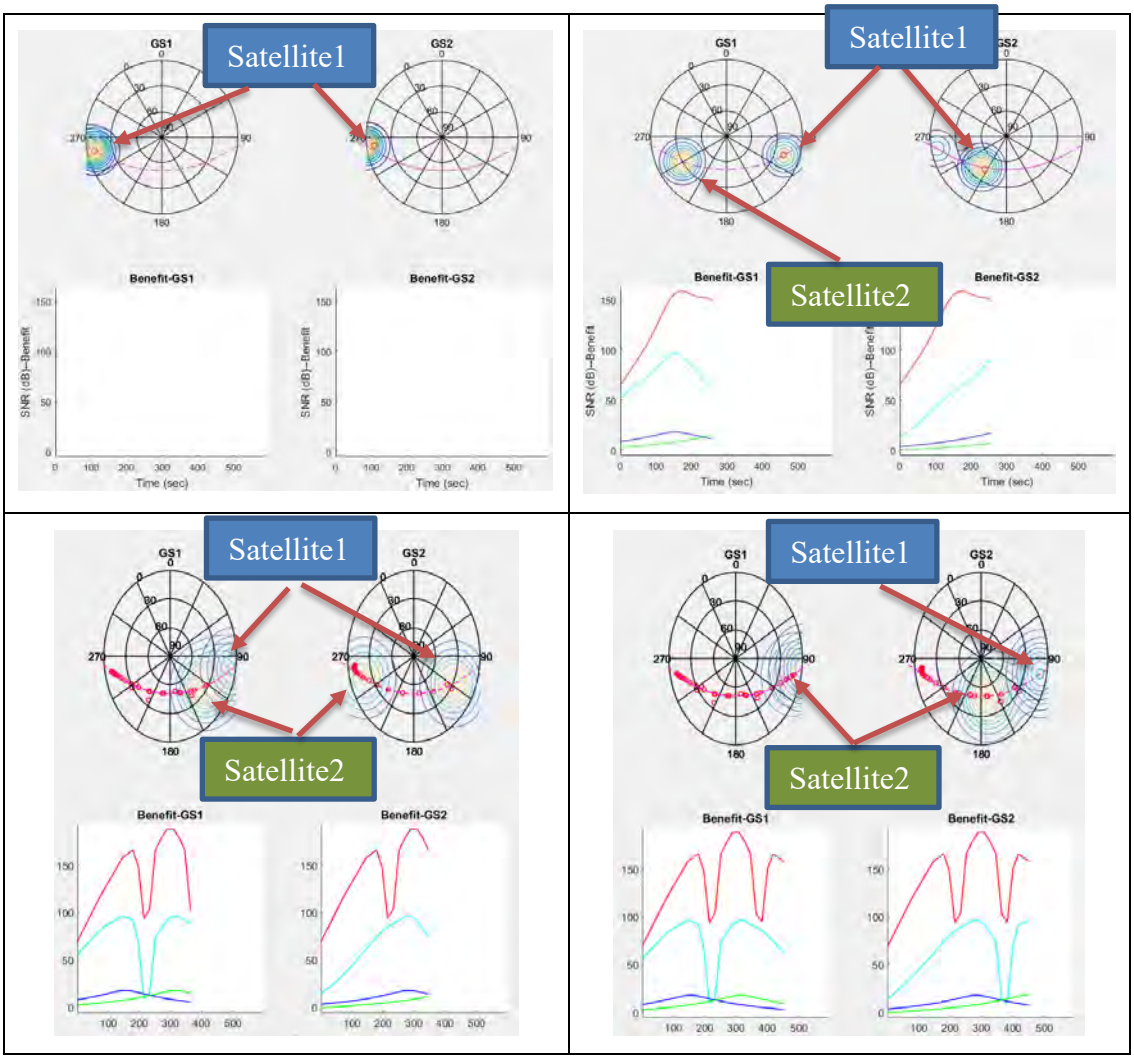


Figure 58. Four panel progression of two-satellite scenario.

The scenario starts with Satellite1 in view of GS1 and with the ground antenna tracking the satellite. Both targets are below the 10° minimum elevation angle at GS2, which keeps the antenna pointed at its elevation limit in the general direction of the targets. GS1 proceeds to follow Satellite1 until it reaches its transition point, at which point GS1

begins to track Satellite2. GS2 continues to track Satellite1 until its individual benefit contribution necessitates a transition to Satellite2. The Gaussian benefit contributions (Figure 58, cyan) for each site correspond to the local idealized link margins (blue and green). The total mission benefit is shown in red and corresponds to the sum of the two cyan curves. Figure 59 provides a closer look at the total summed benefit curve as compared to the idealized link margins.

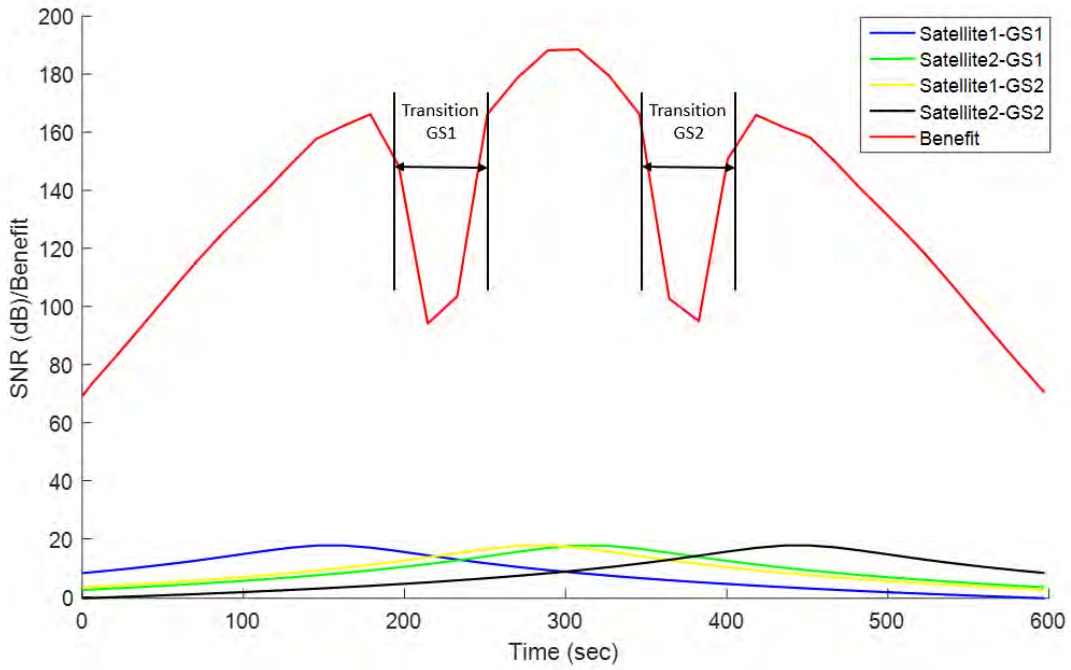


Figure 59. Benefit curve for two satellite two ground station problem

The azimuth and elevation trajectories show consistent behavior with those of the single ground station model. Figure 60 shows the output of the interpolated trajectories as a function of simulation time in UTC. Making trajectories as a function of unique points in time allows for real-world applicability, as the cases in Chapter V will discuss.

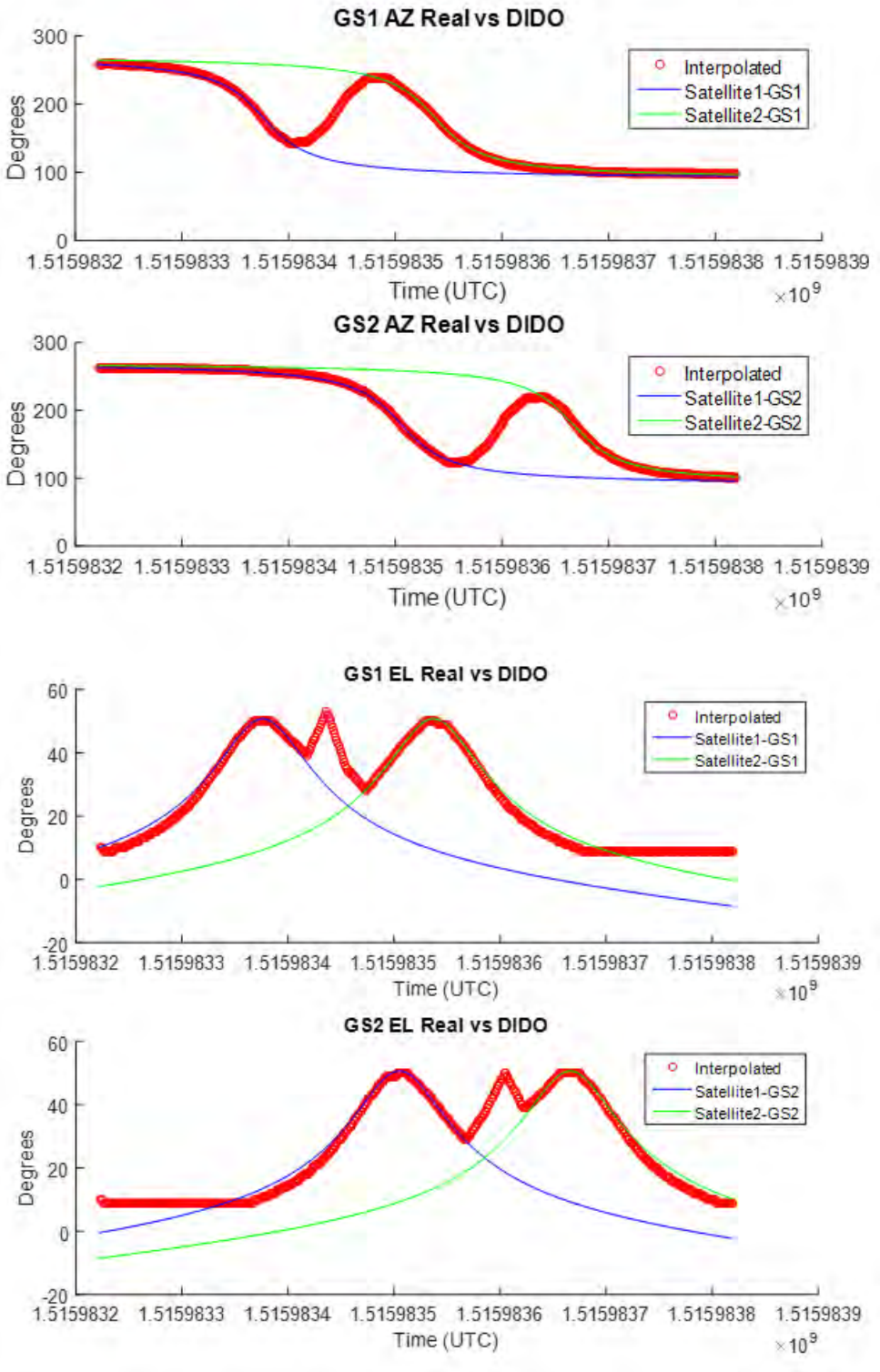


Figure 60. Azimuth and elevation trajectories for the satellites (blue, green) and interpolated outputs (red).

E. THREE SATELLITES, TWO GROUND STATIONS

The final validation scenario involves two ground stations servicing three satellites. The GS1 and GS2 stations remain in the same positions as the scenario described in Section D. Likewise, the addition of a third satellite is identical to the single ground station example of Section B. Figure 61 presents an illustration of this example.

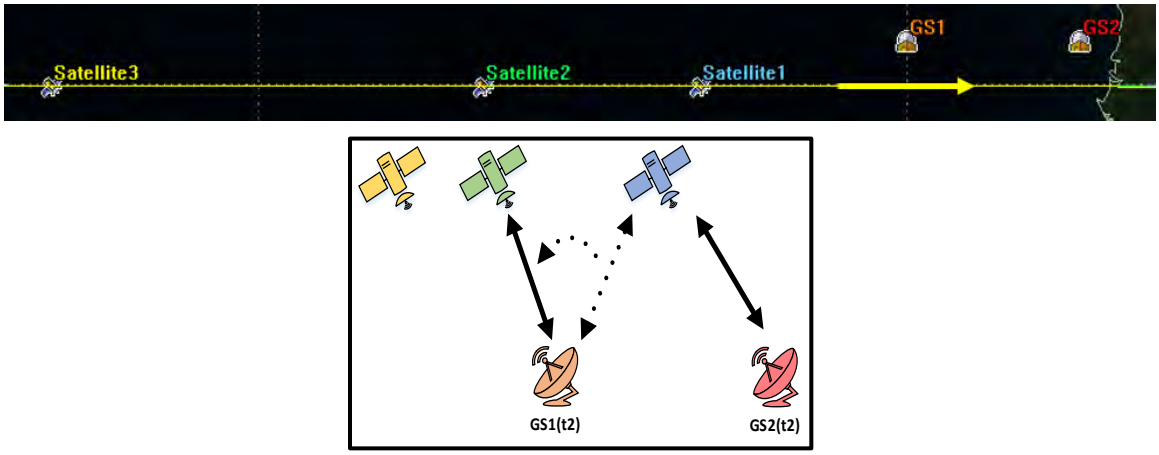


Figure 61. Three satellite/two ground station scenario.

Both stations can simultaneously track the three satellites moving from west to east. Though multiple sites are downlinking simultaneously, a typical mission CONOP would enable only one station to transmit to a satellite. Such constraints are intended to be implemented outside of the optimization routine.

Figures 63, 64, and 65 illustrate an animation of the three satellites and two ground stations. Each snapshot in time shows Gaussian targets representing the satellites on a polar plot of each station with a red circle noting the current position of the antenna. Benefit value plots are shown below each polar plot. The red curve shows the total summed benefit between all satellites and stations. The cyan curve is specific to each station. The blue, green, and yellow curves represent idealized link margins as shown in the previous examples.

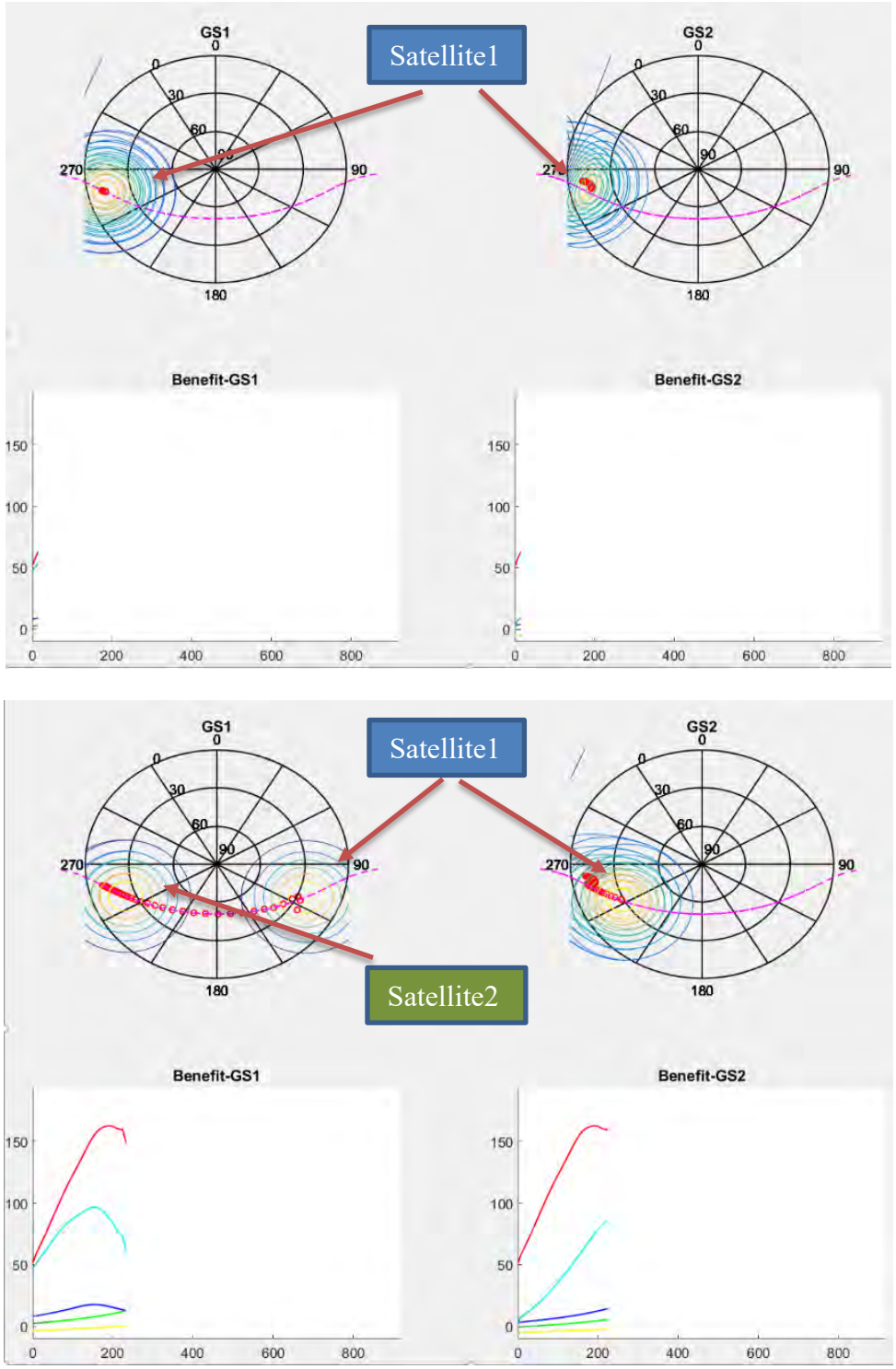


Figure 62. Scenario start (top) and first transition at GS1.

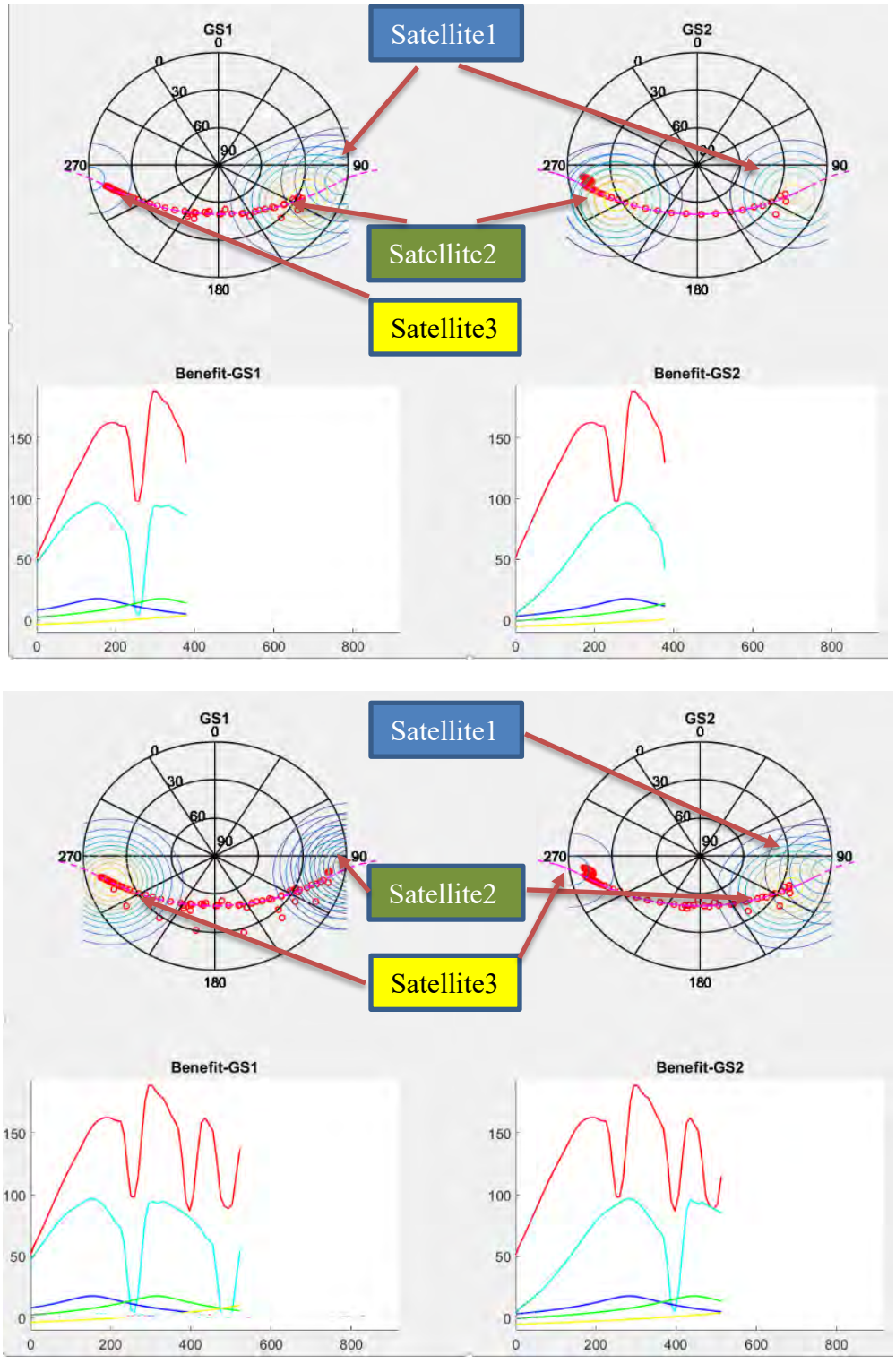


Figure 63. First transition at GS2 (top) and second transition at GS1 (bottom).

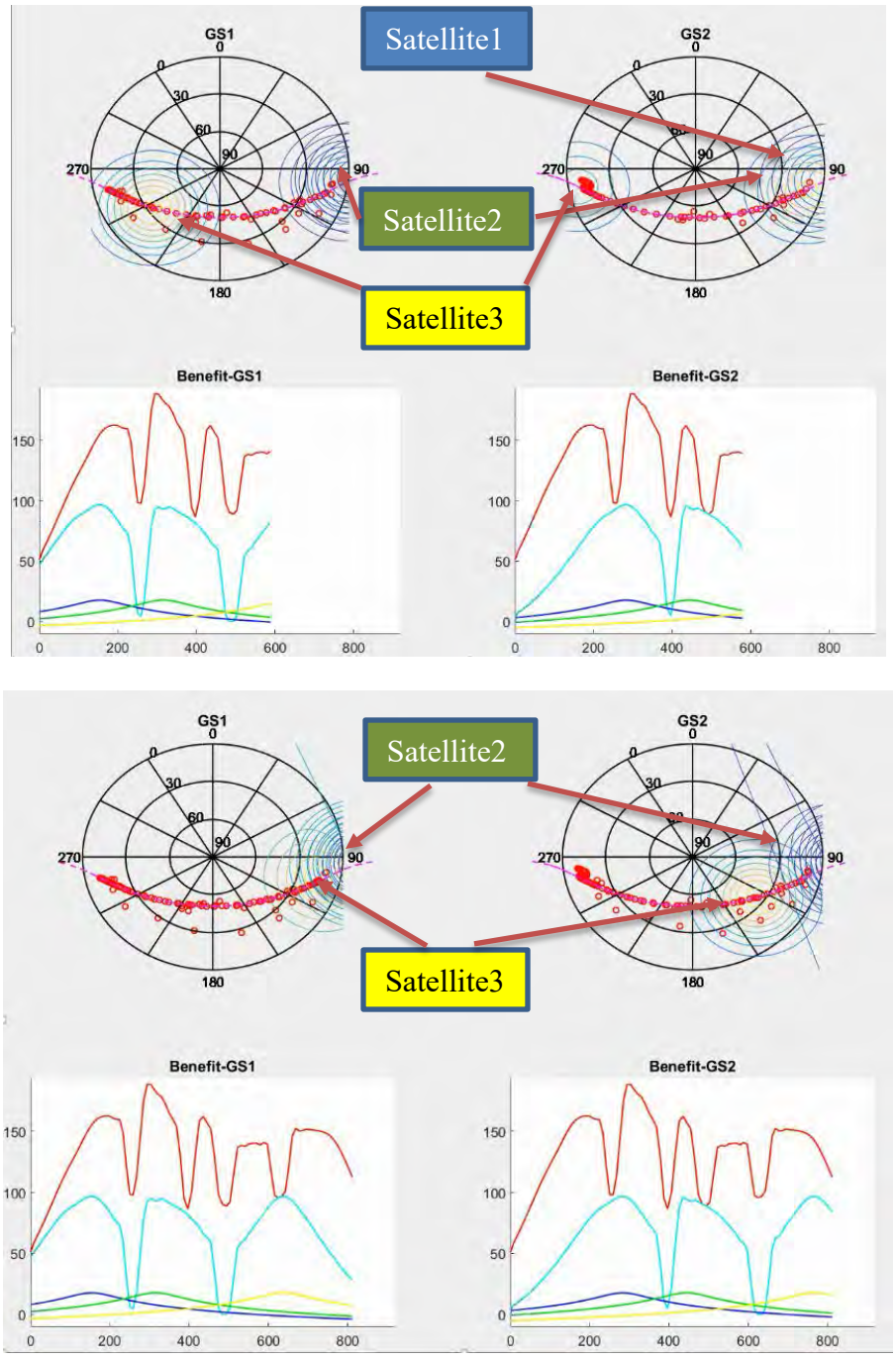


Figure 64. Approaching second transition at GS2 (top) and near LOS at GS2 (bottom).

The progressions create the following benefit function, shown in Figure 65. Figure 66 presents the corresponding interpolated antenna position outputs.

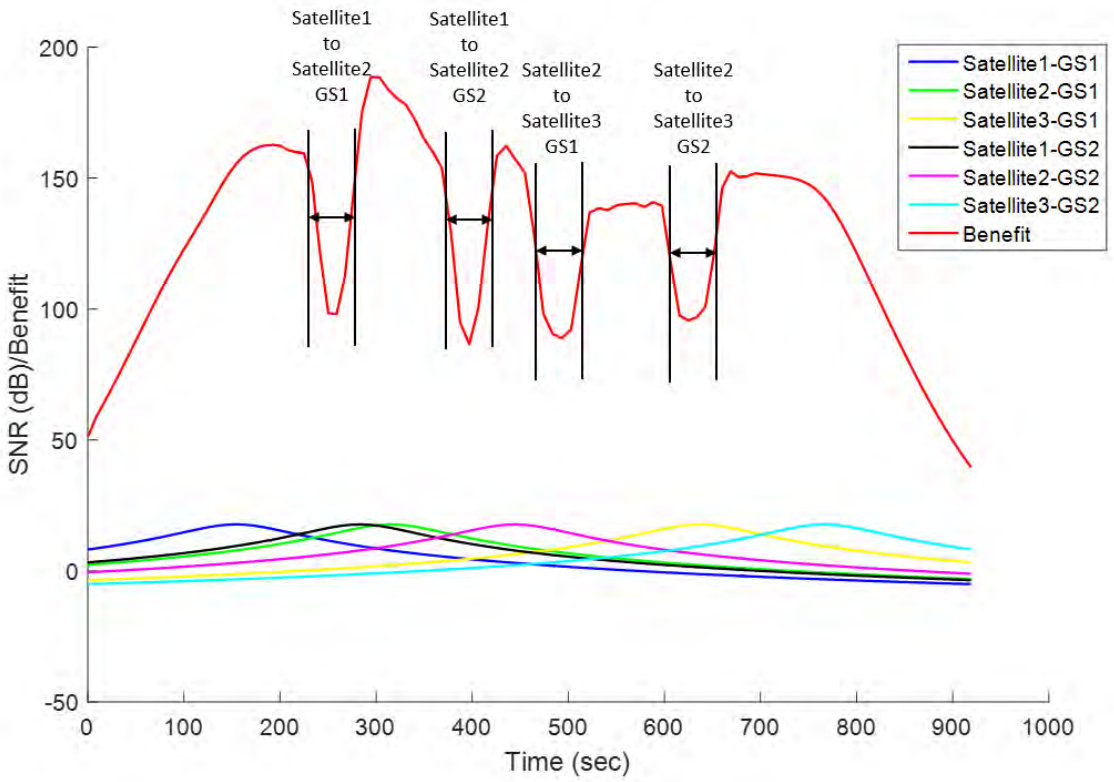


Figure 65. Benefit plot and link margins for three satellite/two station simulation.

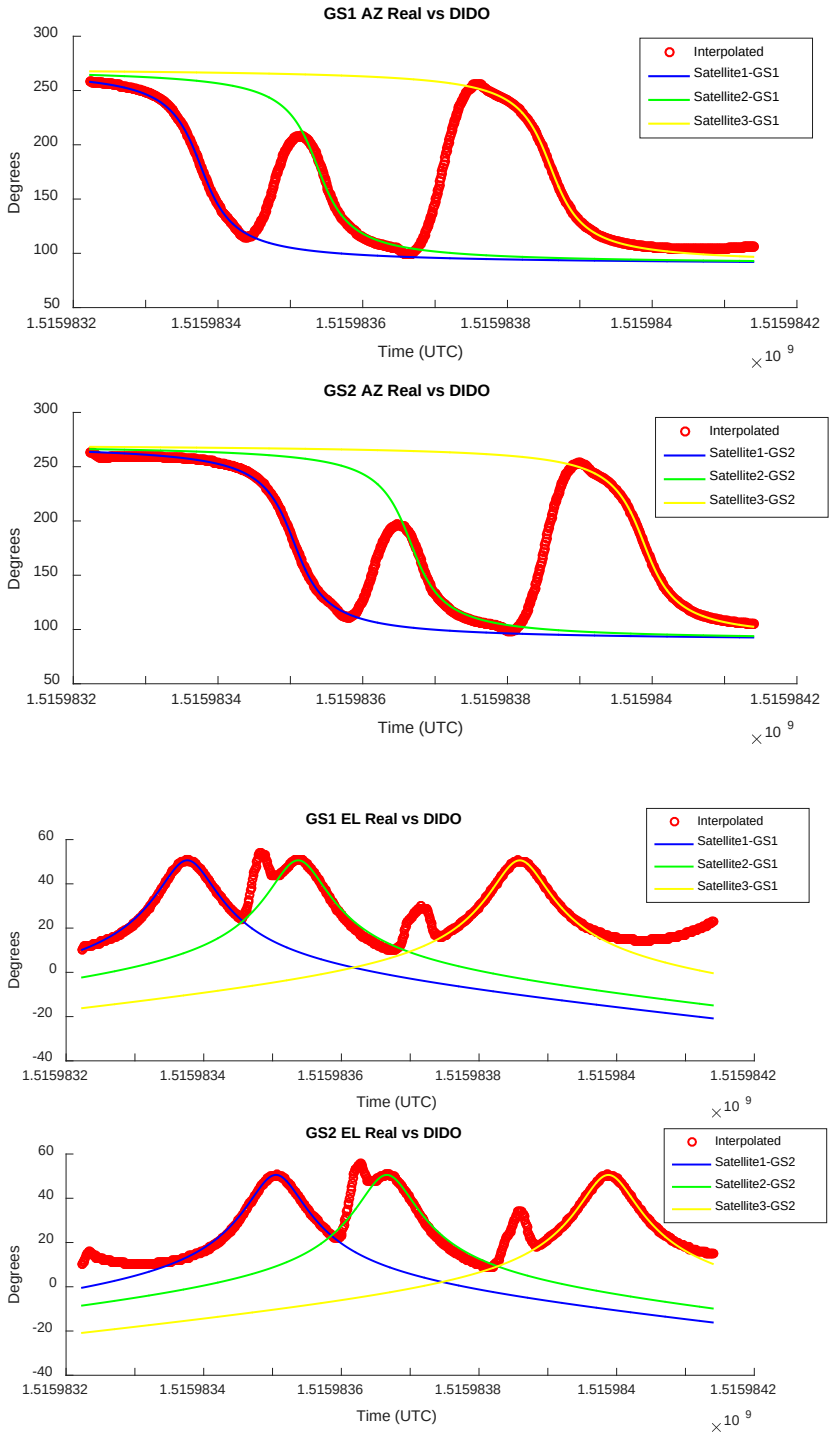


Figure 66. Azimuth (top two) and elevation plots for each ground station, passing satellites, and superimposed interpolated output.

The summed mission benefit in Figure 65 immediately increases when GS1 acquires Satellite1. A temporary dip occurs when transitioning between Satellite1 and Satellite2 at GS1, but results in an even greater total benefit as GS1 and GS2 simultaneously service satellites with the best available link. Once transitions occur between Satellite2 and Satellite3 for both GS1 and GS2, the summed benefit decreases somewhat due to the wider spacing of the satellites. Even in a simple scenario, complexities begin to arise from factors such as orbital position as well as scaling factors in the Gaussian targets. Tracking the peaks of the benefit curve allows for an accurate abstraction of these complex details embedded within the model.

F. TEN SATELLITES, ONE GROUND STATION

This case demonstrates a single ground station servicing a substantially increased quantity of satellites. The goal of the scenario is to introduce more targets than can be handled by GS1, thus creating a trajectory that only considers opportunities with greatest benefit. Figure 67 shows a snapshot from this scenario and Figure 68 shows the corresponding benefit plot. Appendix C shows a systematic progression of the solution.

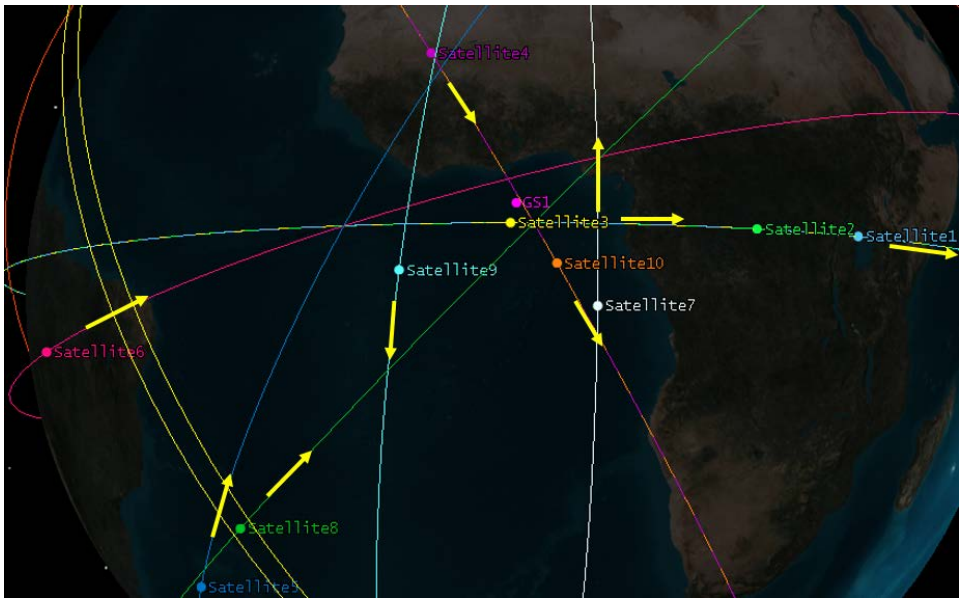


Figure 67. Ten satellite, one ground station scenario. This simulation is meant to saturate the station and force the tracking of the highest benefit targets only.

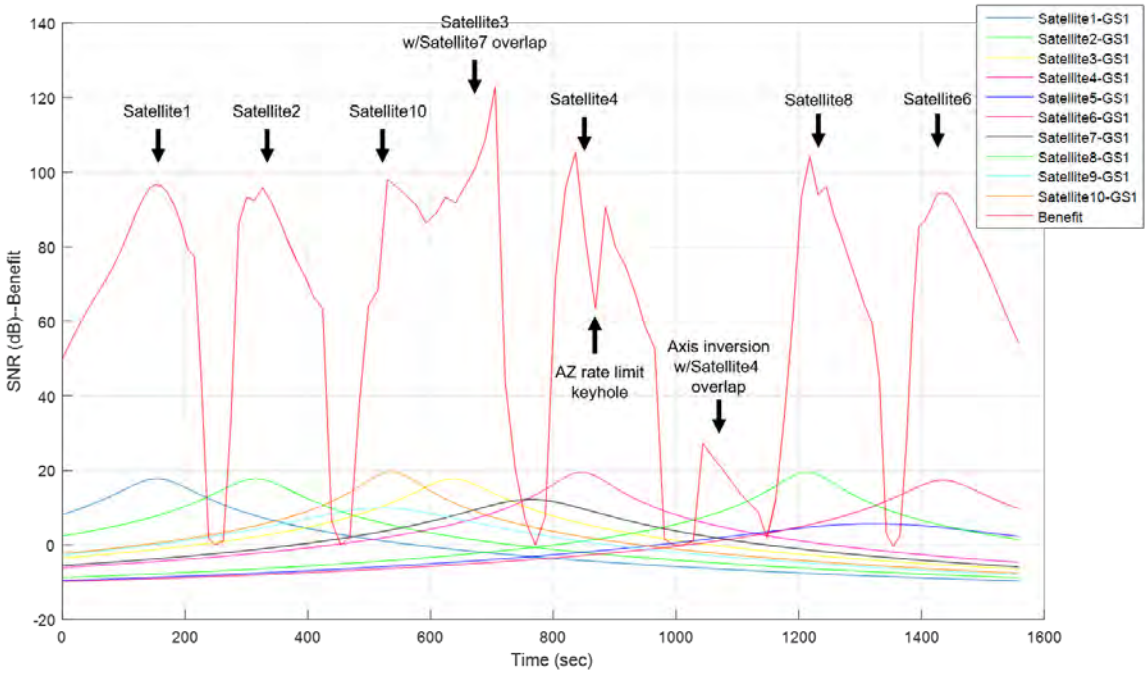


Figure 68. Benefit plot for ten-satellite case.

Within the scenario duration of 25 minutes, ten satellites in various orbits cross GS1 and are eligible for tracking. Only a subset of these satellites could be serviced within the scenario duration. All target Gaussians were weighted equally, with the calculated link margin being the only variable. A breakdown of the progression generated by the optimization software is shown in Table 1. Appendix C contains step-by-step snapshots of the scenario as it progresses.

Satellite1 and Satellite2 were initially tracked as in previous examples after which Satellite10 was intercepted. Subsequently, Satellite3 was acquired, coincidentally resulting in a benefit spike with the concurrent overlap of Satellite7, which was not tracked. The trajectory then moved to Satellite4 which crossed nearest to GS1, resulting in a high elevation angle and high azimuth rotation rate. The rotor reached its maximum angular velocity and fell behind momentarily, losing benefit, resulting in a keyhole typical of the AZ/EL rotor configuration. The trajectory then required an axis inversion to track Satellite8

and Satellite6. In total, the solution gave at least one minute of antenna access to seven satellites, and at least two minutes to six satellites.

Table 1. Tracking progression of ten-satellite case.

Duration (sec)	**Scenario period (sec)**	**Satellite Tracked**
226	0-226	Satellite1
137	287-424	Satellite2
126	514-640	Satellite10
65	656-721	Satellite3
162	803-965	Satellite4
-	997	Axis inversion
134	1043-1177	Satellite4 (waiting for better targets)
126	1205-1331	Satellite8
168	1390-1558	Satellite6

A traditional method of only performing horizon-to-horizon contacts ignoring overlapping passes would have serviced four satellites, as shown in Table 2.

Table 2. Solution for horizon-to-horizon contacts with no overlap.

Duration (sec)	**Scenario period (sec)**	**Satellite Tracked**
302	0-302	Satellite1
295	383-678	Satellite10
299	681-980	Satellite4
303	1060-1363	Satellite8

The traditional method of only taking passes without overlap yielded an average contact time of five minutes per satellite. In an application where the link margin is strong hear the horizon (at long distances), this method is effective for servicing four out of the ten spacecraft in this scenario. Such a CONOP is routine for large ground stations communicating with spacecraft that have substantial transmission power. The other satellites would have been scheduled for contacts later. A low-cost ground station servicing small satellites with limited downlink power would waste the majority of these long duration contacts, as the link margin would be insufficient to establish a closed link.

By employing a benefit-based objective, more satellites were serviced in the same contact duration. Since the contact time was reduced from the traditional case, additional ground stations could add capacity and give each mission an equivalent amount of contact as the traditional method utilizing a large ground station antenna. Furthermore, the trajectory generated by the optimization software enables autonomous operations. The solution automatically incorporated position, velocity, and torque constraints. When appropriate, the rotor axes were inverted to service additional satellites. Rotor inversion will be discusses in further detail in Chapter V. In a real-world scenario, this 10-satellite conflicting case would be one of potentially dozens that occur each day. Manually deconflicting these events would be overwhelming for human operators, and utilizing a

traditional method of running only non-overlapping horizon-to-horizon contacts does not provide a full use of the network's capacity.

G. CHAPTER SUMMARY

This chapter presented benchmark cases with multiple satellites and ground stations to allow evaluation of the problem formulation derived in Chapter III when implemented in software. Even in specially designed cases with predictable outcomes, the complexity of the solution quickly increases with the addition of new satellites and ground stations. The additional V&V performed for these cases demonstrates the formulation's feasibility for addressing the satellite-tracking problem over traditional methods. The following chapter expands on these results and presents several real-world examples.

V. EXPANDED EXPERIMENTATION

This chapter incorporates data from real-world CubeSats to validate the models developed in Chapter IV. The Chapter IV models were updated with parameters from the mission including satellite positions and ground station locations. The results of these optimization runs were formatted to drive the MC3 system, thereby allowing them to be tested in the field.

The MC3 ground stations available for validation testing included HSFL, NPS, SDL, UNM, and AFIT. Three satellites operated by the MC3 network were used for the demonstrations described in this chapter. The satellites are part of the Picosats Realizing Orbital Propagation Calibrations Using Beacon Emitters (PROPCUBE) mission seeking to aid ground-based ionospheric observations and satellite orbit determination for the Naval Research Laboratory (NRL) [55]. Two of the three satellites, named Flora and Merryweather, launched into a 500 x 800 km orbit with 63º inclination in October 2015. A third PROPCUBE, called Fauna, was deployed into a 51º inclination, 450 km circular orbit from the Cygnus OA-8 ISS resupply vehicle in December 2017.



Image provided by Tyvak Nano-Satellite Systems, Inc.

Figure 69. PROPCUBE satellites undergoing preparations for flight.

Despite launching on the same vehicle, Merryweather and Flora drifted into slightly different altitudes over time. Consequently, Merryweather slowly overtakes Flora in their nearly identical orbital plane for a three-week window approximately every three months. This offers a unique opportunity to test deconfliction algorithms when the positions of Flora and Merryweather overlap, as they are simultaneously in view of ground assets that are constrained to communicate with only one satellite at a time. Subsequent tests also included overlapping Fauna contacts creating opportunities to deconflict up to three satellites simultaneously from multiple ground stations. Though a ground-based, directional antenna is constrained to communicate with a single satellite at any point in time, an optimized antenna slew seeks to maximize contact time with each satellite as a function of its BVF. Figure 70 shows an illustration of one such conflicting event involving three satellites and the SDL, UNM, and NPS ground stations.



Figure 70. Illustration of MC3 stations and three PROPCUBE satellites.

The event displayed in Figure 70 shows Fauna (catalog number 43052) passing over the UNM and SDL stations, offering a routine contact opportunity. Minutes after its contact window opens, Merryweather (catalog number 90738) and Flora (catalog number 90736) also move into view of SDL and UNM, as well as NPS. A human operator can deconflict this scenario by picking and choosing stations to service each particular satellite. An example solution could require that UNM contact Fauna, SDL contact Flora, and NPS contact Merryweather. This conflict resolution does not leave any satellite unserviced, and may be completely adequate for the requirements of the mission. However, this ad-hoc solution may not be taking advantage of additional capacity in the system – a requirement for networks that will handle hundreds or thousands of targets.

Though only one station can transmit to one satellite at a time (to avoid sending conflicting/interfering commands), multiple stations often listen for downlinks, thereby increasing space-to-ground throughput. Due to the inherently weak signal originating from CubeSats, marginal gain from small/low-cost antenna apertures that service them, and interference from terrestrial noise, incomplete or corrupted data packets are common. Multiple stations simultaneously tracking a satellite offer an increased opportunity for successfully decoding any particular transmission from the satellite. The PROPCUBE satellites (as well as many other CubeSats) utilize the AX.25 communication protocol [97] with no bit error correction, thus significantly increasing the probability of corrupted data.

Programming routine operational constraints, such as those described in this example, into an optimization algorithm can enable non-intuitive tracking trajectories that allow one ground station to transmit to a satellite, and one or more stations to listen simultaneously for responses. Likewise, the point at which the stations break from tracking one satellite and transition to the next may not be immediately intuitive and/or requires substantial human-in-the-loop planning. Reconfiguring the ground stations to service diverse satellites with custom waveforms and communications protocols can be tedious if not performed on the fly. Scaling these considerations to potentially hundreds of events in a day clearly requires automation given the cost-constrained nature of small satellite missions.

Section A presents an overview of the operational MC3 environment to provide context for the experimental implementation. Section B then discusses the simulation environment created for testing the first variants of the satellite/ ground station network optimization algorithms. Examples are presented in order of increasing complexity. Finally, experimental results are shared from several testing campaigns using the orbiting PROPCUBE satellites and the MC3 stations that support them.

A. MC3 OPERATIONAL TESTING ENVIRONMENT

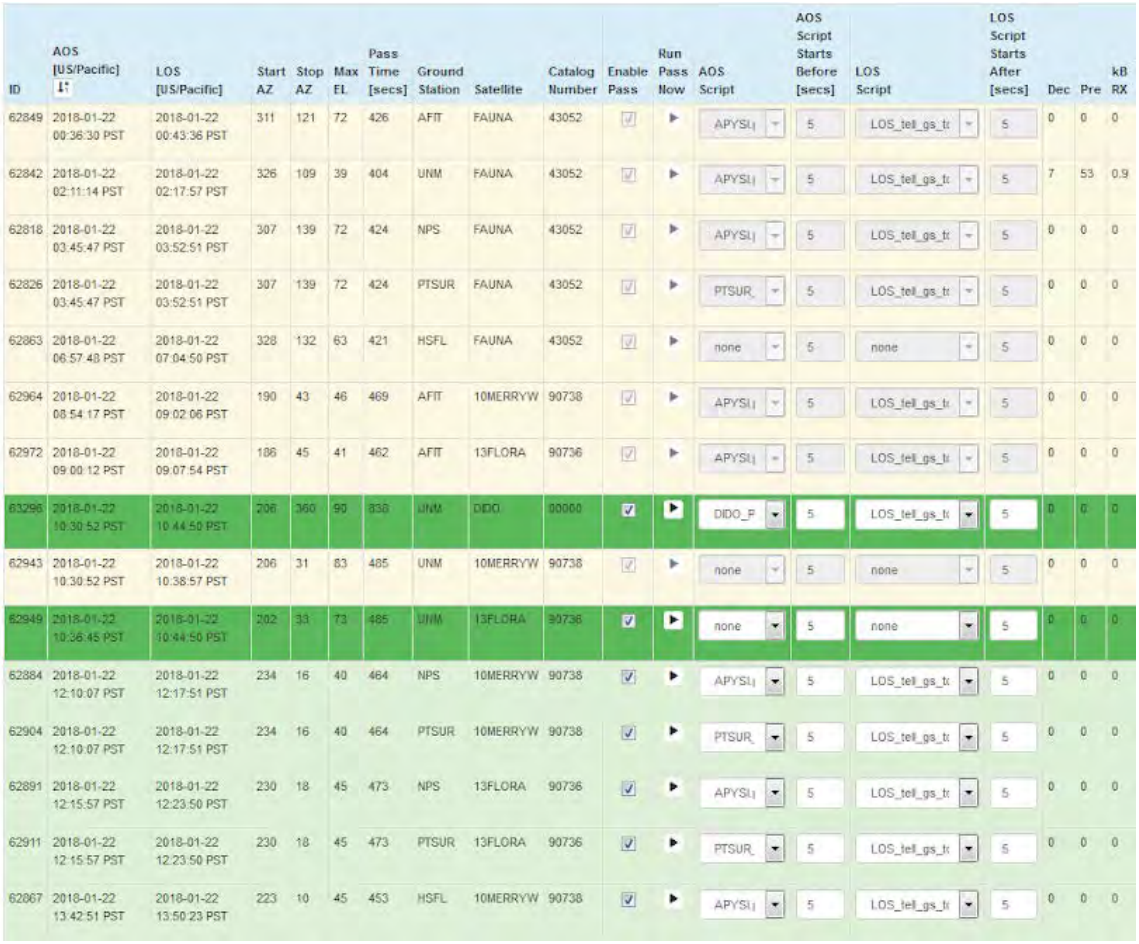
Since 2015, the network utilized the MC3 Picosat Interface Pipeline Extension (MPIPE) [39] software developed at NPS. The software is primarily written in Python with MATLAB governing ephemeris propagation and optimization algorithm implementation. At the time of this writing, a new software architecture called the Satellite Agile Transmit Receive Network (SATRN), described in Chapter I, is in the early stages of trials with the MC3 network and will eventually replace MPIPE. The implementation this dissertation describes is compatible with the MPIPE software; relatively minor modifications of data outputs and formatting can enable compatibility with SATRN or virtually any other mission operations software.

1. At the NPS SOC

The MPIPE software allows operators to have bent-pipe communications with their spacecraft from a SOC or MOC through a remote MC3 ground station. The contact windows, shown in Figure 71, are computed based on the location of each ground station and the projected ground track of the corresponding satellite(s). During a given contact window, MPIPE is also responsible for driving any peripherals that enable communications with the satellite. This includes driving antenna rotors, configuring radios, powering amplifiers, and actively compensating for Doppler shift.

The bent-pipe nature of MPIPE allows third-party mission operations software integration by relying on common Transmission Control Protocol/Internet Protocol (TCP/IP) communications. For example, Tyvak Inc., the developers of the PROPCUBE satellites, use proprietary software called Satnet for C2. Satnet relies on both User

Datagram Protocol (UDP) and TCP internet communications protocols. Given the ubiquity of these protocols, integrating Satnet into the MPIPE architecture was straightforward.



ID	AOS [US/Pacific]	LOS [US/Pacific]	Start AZ	Stop AZ	Max EL	Pass Time [secs]	Ground Station	Satellite	Catalog Number	Enable Pass	Run Pass Now	AOS Script	AOS Script Starts Before [secs]	LOS Script	LOS Script Starts After [secs]	Dec	Pre	kB RX
62849	2018-01-22 00:36:30 PST	2018-01-22 00:43:36 PST	311	121	72	426	AFIT	FAUNA	43052	☑	▶	APYSU	5	LOS_tell_gs_tc	5	0	0	0
62842	2018-01-22 02:11:14 PST	2018-01-22 02:17:57 PST	326	109	39	404	UNM	FAUNA	43052	☑	▶	APYSU	5	LOS_tell_gs_tc	5	7	53	0.9
62818	2018-01-22 03:45:47 PST	2018-01-22 03:52:51 PST	307	139	72	424	NPS	FAUNA	43052	☑	▶	APYSU	5	LOS_tell_gs_tc	5	0	0	0
62826	2018-01-22 03:45:47 PST	2018-01-22 03:52:51 PST	307	139	72	424	PTSUR	FAUNA	43052	☑	▶	PTSUR	5	LOS_tell_gs_tc	5	0	0	0
62863	2018-01-22 06:57:48 PST	2018-01-22 07:04:50 PST	328	132	63	421	HSFL	FAUNA	43052	☑	▶	none	5	none	5	0	0	0
62964	2018-01-22 08:54:17 PST	2018-01-22 09:02:06 PST	190	43	46	469	AFIT	10MERRYW	90738	☑	▶	APYSU	5	LOS_tell_gs_tc	5	0	0	0
62972	2018-01-22 09:00:12 PST	2018-01-22 09:07:54 PST	186	45	41	462	AFIT	13FLORA	90736	☑	▶	APYSU	5	LOS_tell_gs_tc	5	0	0	0
63298	2018-01-22 10:30:52 PST	2018-01-22 10:44:50 PST	206	360	90	838	UNM	DIDO	00000	☑	▶	DIDO_P	5	LOS_tell_gs_tc	5	0	0	0
62943	2018-01-22 10:30:52 PST	2018-01-22 10:38:57 PST	206	31	83	485	UNM	10MERRYW	90738	☑	▶	none	5	none	5	0	0	0
62949	2018-01-22 10:36:45 PST	2018-01-22 10:44:50 PST	202	33	73	485	UNM	13FLORA	90736	☑	▶	none	5	none	5	0	0	0
62884	2018-01-22 12:10:07 PST	2018-01-22 12:17:51 PST	234	16	40	464	NPS	10MERRYW	90738	☑	▶	APYSU	5	LOS_tell_gs_tc	5	0	0	0
62904	2018-01-22 12:10:07 PST	2018-01-22 12:17:51 PST	234	16	40	464	PTSUR	10MERRYW	90738	☑	▶	PTSUR	5	LOS_tell_gs_tc	5	0	0	0
62891	2018-01-22 12:15:57 PST	2018-01-22 12:23:50 PST	230	18	45	473	NPS	13FLORA	90736	☑	▶	APYSU	5	LOS_tell_gs_tc	5	0	0	0
62911	2018-01-22 12:15:57 PST	2018-01-22 12:23:50 PST	230	18	45	473	PTSUR	13FLORA	90736	☑	▶	PTSUR	5	LOS_tell_gs_tc	5	0	0	0
62867	2018-01-22 13:42:51 PST	2018-01-22 13:50:23 PST	223	10	45	453	HSFL	10MERRYW	90738	☑	▶	APYSU	5	LOS_tell_gs_tc	5	0	0	0

Contacts in the past are denoted in tan, future contacts in light green, and active contacts in dark green. Notable columns are AOS, LOS, satellite, ground station, data packets decoded (Dec), and preambles detected (Pre).

Figure 71. MPIPE web-based user interface.

Though Satnet is installed at the NPS SOC, the VPN architecture of the MC3 network enables Satnet and command and control software like it to reside at a remote MOC. This powerful concept enables MC3 to service a diverse user base through common infrastructure without the need of a dedicated watch floor, saving considerable cost.

Operating the PROPCUBE satellites using the MC3 network required the integration of several major components, shown in Figure 72. Tyvak's Satnet software came with an accompanying program, called SatComm, which was responsible for configuring the TCP/IP communications of Satnet. Interfacing Satnet into MPIPE required the development of a front-end processor (FEP). Implementations involving FEPs are common in mission operations environments when integrating software with inherently different functions.

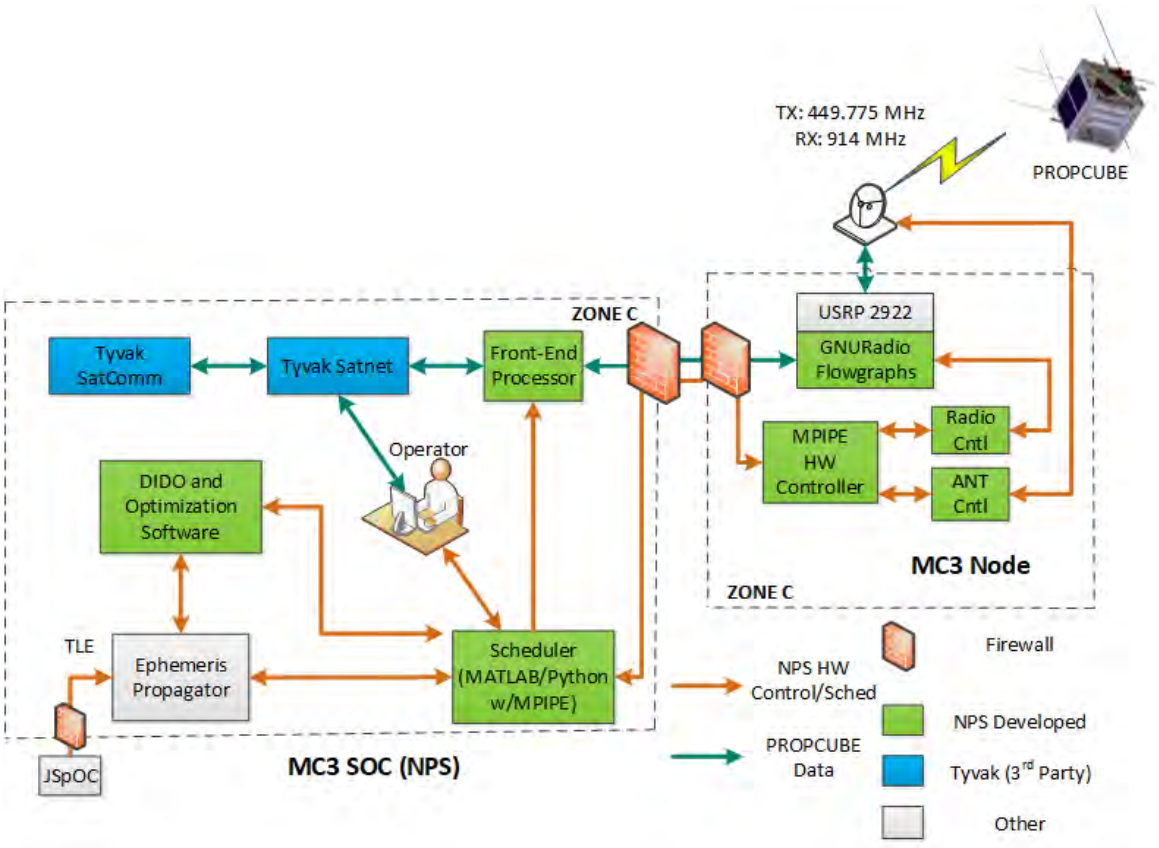


Figure 72. MC3 block diagram for SOC and remote station.

The MPIPE Scheduler, illustrated in Figure 71, created unique events for each contact opportunity at a ground station and satellite pairing. The system was designed to execute a script at AOS and another at LOS. The contents of the scripts depend on the mission, system configuration, and any additional desired outcomes. The scripts are highly

customizable by design, enabling prototyping and experimentation with little overhead. However, multiple years of usage and refinements have created a stable system enabling day-to-day automated contacts with valuable orbiting assets and manageable risk. The PROPCUBE operator typically maintains a supervisory, rather than active role, by setting up automated scripts rather than manually issuing commands during a pass.

An SGP4 propagator feeds the MPIPE scheduler system that pulls daily TLE updates from the JSpOC website [98]. The propagator this research used was STK 10, as it provided a MATLAB Application Programming Interface (API) that was convenient for integration into the DIDO optimization package.

2. At the Remote MC3 Node

Due to their compatibility with the PROPCUBE satellites, five of the MC3 stations comprised the testing campaign. Though each node has site-specific considerations, the communications-enabling components were identical, thus allowing scalability with minimal rework required.

The antennas used for UHF communications with the PROPCUBE satellites were primarily directional Yagi elements, shown in Figure 73. These low-cost, directional antennas provide moderate gain and relatively wide beam angles. The wide beam angle allows a lower-cost and less precise rotator for pointing the directional antenna at a passing satellite. A higher gain system concentrates energy from a smaller portion of the sky, requiring higher precision rotators, thus driving up cost. For the scope of UHF CubeSats such as PROPCUBE, reduced performance was acceptable in exchange for lower cost. Future instantiations of this optimization research will take advantage of more precise, higher gain parabolic dishes that are in the commissioning process on the MC3 network as more CubeSats move to S-band and X-band downlinks.

The MC3 sites primarily utilize two radios for communicating with PROPCUBE-class vehicles. Legacy ICOM IC-9100 radios, paired with Kantronics KWM-9612+ modems have been in use for a number of years on the MC3 network. They follow a long-time trend within the CubeSat community to leverage amateur radio protocols and

equipment to ease integration and save development cost. These radios were used for uplinking data to the PROPCUBE satellites in the 450 MHz band. Given the increased maturity in the CubeSat community as well as frequency allocation restrictions, the MC3 network has almost entirely phased out the IC-9100 radios in favor of software-defined radios (SDRs).



Multiple Yagi antennas mounted on a crossboom driven by a low-cost Yaesu G5500 rotor. The single Yagi antenna (right) is used for uplink at 450 MHz. A set of four Yagis installed together for downlink at 902–928 MHz.

Figure 73. Typical PROPCUBE-compatible UHF configuration for an MC3 station.

The SDRs in use for PROPCUBE operations at the MC3 sites were the National Instruments Universal Software Radio Peripheral (USRP) 2922, shown in Figure 74. During the research testing campaign, these SDRs were primarily in use for PROPCUBE

downlinks. Future upgrades to the MC3 design will phase out the ICOM IC-9100 uplink and operate the SDRs exclusively for both transmitting and receiving operations [99]. At the time of testing, the HSFL and NPS sites had been updated to include SDR uplinks.



Image retrieved from http://www.ni.com/en-us/support/model.usrp-2922.html

Figure 74. National Instruments USRP 2922.

The SDR shown in Figure 74 is part of a family of low-cost, COTS transceivers compatible with a wide range of frequencies and waveforms common to the satellite industry. There exist several methods and programming languages for configuring the SDRs including C++, GNU Radio, Pothos, and LabView. The NPS Small Satellite Laboratory has been leveraging GNU Radio [100] which is a free and open-source programming environment that allows block-based assembly of flowgraphs that enable digital signal processing. A default installation of GNU Radio contains the majority of functionality required to process digital radio signals, however it also has the capability of incorporating custom-built blocks and functionality. The PROPCUBE SDR downlink implementation incorporates a combination of both stock and custom blocks, shown in Figure 75 [101].

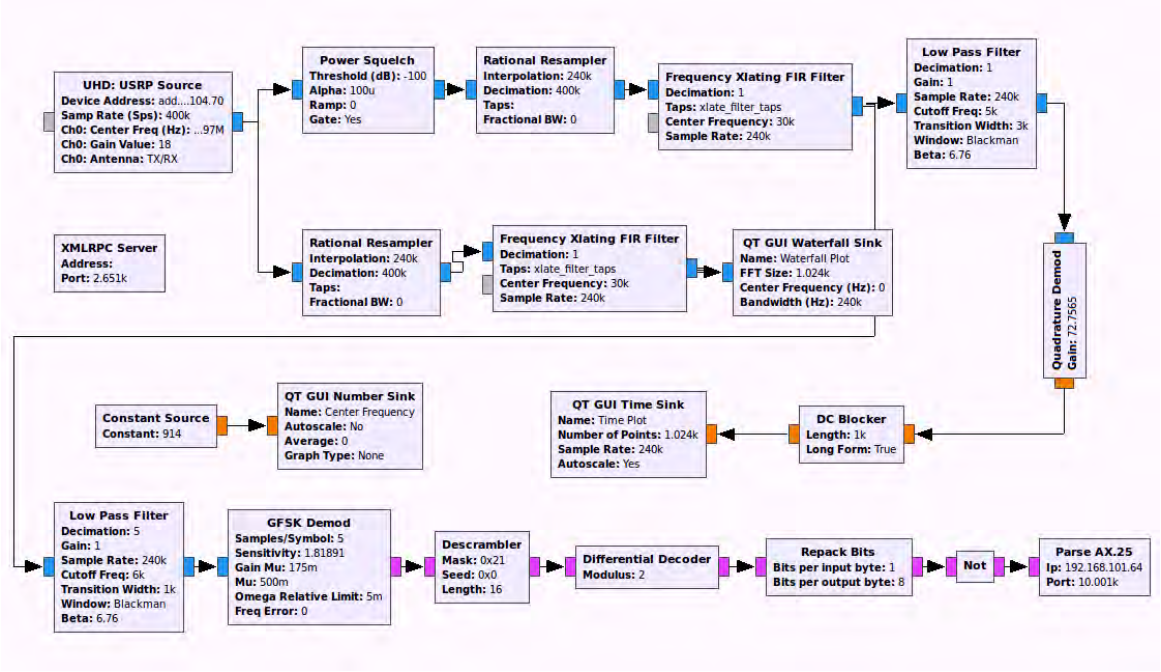


Figure 75. GNU Radio flowgraph for PROPCUBE downlink.

The flowgraph also provides visual feedback by means of a waterfall plot to the operator for real time situational awareness, shown in Figure 76. The waterfall plot is an effective tool for evaluating whether or not the satellites are responding to commands sent from the ground. Given the available link margins, the signal is often too weak to be decoded; however, responses can often be detected visually, thus providing a qualitative assessment of signal strength, Doppler shift [102], and other effects, such as environmental noise or satellite rotation.

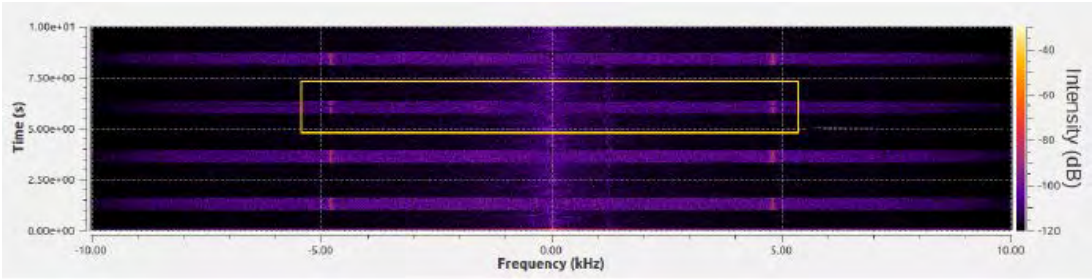


Figure 76. Waterfall plot showing received RF energy of PROPCUBE satellite (yellow box). Source: [102].

A complete lack of responses may indicate improper ground antenna pointing, an anomaly with the ground station, or an anomaly with the satellite. Spacecraft and ground station anomalies occur periodically and, once detected, are often correctable through operator intervention. The success of antenna trajectories generated in DIDO were evaluated, in part, by assessing whether or not the satellites were receiving uplink requests and how often response packets would be received and decoded by a ground station. Ruling out any ground or satellite anomalies, a lack of data produced during a contact utilizing DIDO-generated trajectories suggests erroneous implementation.

3. Pass Planning and Deconfliction

The MATLAB software environment generated deconfliction trajectories using the DIDO computational package. The model described in Chapters II and III was translated into MATLAB syntax for use with DIDO. Inputs into the model were generated using STK 10 propagation software, which allows ephemerides of satellites and their ground stations to be calculated for a given time period. After finding times in which the Flora, Merryweather, and Fauna satellites were simultaneously in view of one or more ground stations, the ephemerides for these events were exported from STK into MATLAB for processing by the DIDO optimizer.

Generating a deconflicted contact schedule requires several steps, outlined in Figure 77. This process, even with just a few orbiting satellites and multiple ground stations, can be labor-intensive for a human operator. The operator first chooses which satellites and ground stations to consider. The most recent satellite TLEs, ground station locations, and orbital propagator then dictate the outputs of the full schedule.

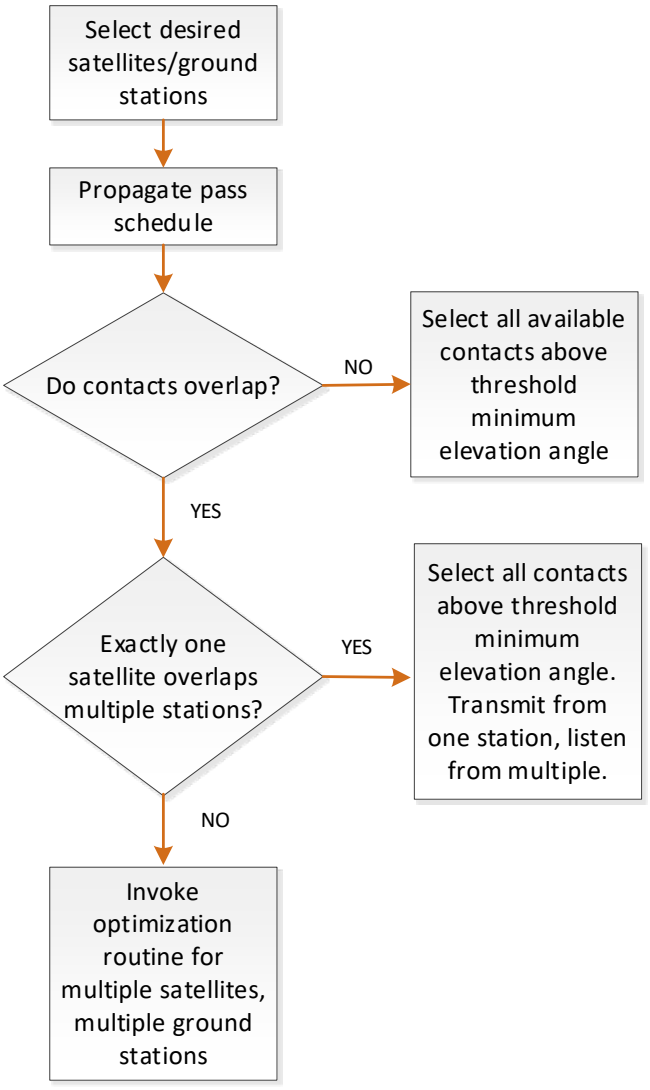


Figure 77. Pass planning flowchart.

The extent of the future schedule is user-selectable, called the *planning horizon*. The planning horizon for this research spanned four to six days due to small uncertainties in the satellite position knowledge that result in projected schedules of decreasing accuracy. Overlapping contact windows are detected from the list of available events and flagged as potential conflicts. Figure 78 demonstrates several examples for contact windows in the multi-satellite/two ground station case.

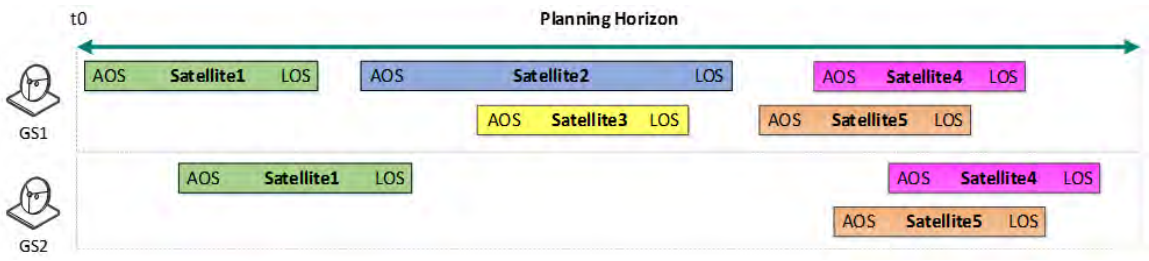


Figure 78. Example of overlapping contact windows.

Detecting a conflicting pass requires comparing the AOS and LOS of each event to each other over the planning horizon. For a given satellite or ground station, if a contact AOS falls between another AOS and LOS, a possible conflict may exist. Referring to Figure 78, Satellite1 enters view of GS1 and later GS2. This is not necessarily a conflict, as multiple stations are able to receive signals from a single satellite. Allowing only one of the two stations to radiate may be prudent (depending on the mission) so as to avoid self-induced RF interference at the spacecraft. This case would not require invoking the optimization software.

A second type of overlap occurs when Satellite2 becomes available at GS1. Initially, GS1 can track Satellite2 because no other resource constraint exists at GS1. However, the appearance of Satellite3 during the Satellite2 contact requires deconfliction because the station is only capable of servicing one satellite at a time. This event is not visible to GS2; therefore, it cannot add support capacity. To deconflict these contacts, the optimization software is given the outer event bounds (AOS and LOS of Satellite2 in this case) and the names of the satellites and ground station(s) involved. The software produces the two satellite, one ground station trajectory based on each satellite's orbital parameters and operator-defined Gaussian sizes. The resulting deconflicted trajectory then becomes available in the master schedule for execution.

Multiple satellites may also overlap multiple ground stations, as Satellite4 and Satellite5 do in Figure 78. An *ad-hoc* deconfliction may involve selecting GS1 to service one satellite while GS2 services the other, though it may not maximize network capacity. Invoking the optimizer generates the expected solution for the two satellite, two ground

station cases that previous chapters described. The scenario runs from Satellite5 AOS at GS1 to Satellite4 LOS at GS2 (the outer bounds of the contact event). The DIDO-generated outputs are loaded into the scheduler system as separate trajectories for each ground station. Figure 79 presents a flowchart of the optimizer's software implementation for the MC3 system.

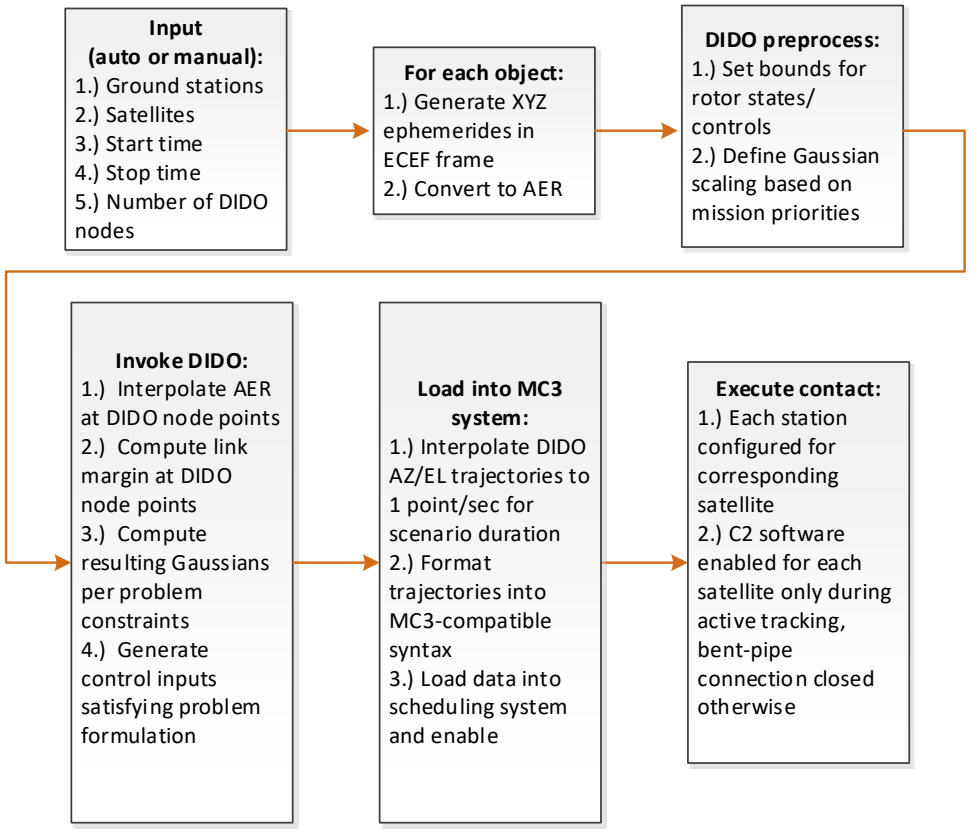


Figure 79. Flowchart for DIDO-integrated system on MC3.

Having initially applied the flowchart from Figure 77 for a real contact schedule, such as the one shown in Figure 71, or the idealized conflicting cases in Figure 78, any identified conflicts invoke the optimization routines by providing the satellites, ground stations, start/stop times, and desired number of nodes. The MATLAB software configures STK to produce ephemerides in Earth Centered Earth Fixed (ECEF) XYZ coordinates for a given deconfliction event. In an ECEF reference frame, a ground-based site remains

constant while the coordinates of a satellite vary as a function of the orbit, Earth's rotation, spacecraft maneuvers, and orbital perturbations. The XYZ vectors were then processed into Azimuth, Elevation, and Range (AER) values centered at each ground station.

Preprocessing steps for DIDO primarily include configuring scaling relationships and defining boundary conditions. This application uses rotor constraints, such as torques, angular velocity, and position boundaries, and time constraints, such as start and stop times.

The optimizer chooses unevenly distributed nodes spread over the scenario duration and their location is not user-selectable. Therefore, a smooth interpolant was required to determine where each satellite was located along the propagated positions generated in STK. Using MATLAB's *interp1* function with *pchip* cubic spline interpolation, the optimization software was able to choose where an instantaneous solution would be generated and where the targets were located for each time point in the scenario. Using this information, the target Gaussians could be evaluated, which converged on the final desired control torques and resulting antenna positions.

Once a solution was converged upon, the antenna positions were formatted to specifically drive the MC3 system and loaded into the MPIPE program. The new optimized passes were listed in the MC3 MPIPE Scheduler (Figure 71) under the "DIDO" designator for each ground station. In the case of the PROPCUBE demonstrations, a user would select a pass to execute with the real satellites and stations. At pass time, the stations performed any satellite-specific configurations, such as enabling amplifiers, programming the SDR, and opening up network connections to the Satnet C2 software. The system also monitored which satellite was actively tracked at each station, and informed the Satnet FEP to automatically direct communications to the appropriate station and satellite. Transitions between satellites then redirected communications from Satnet automatically.

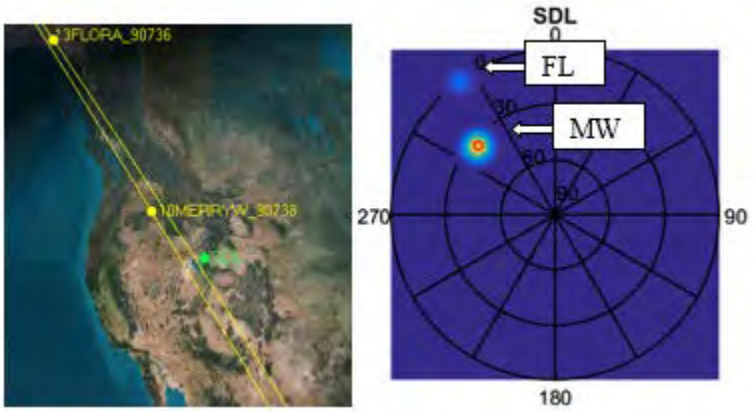
B. REAL-WORLD SCENARIOS

The following section describes testing performed with the PROPCUBE satellites and MC3 ground stations.

1. Two Satellites, One Ground Station

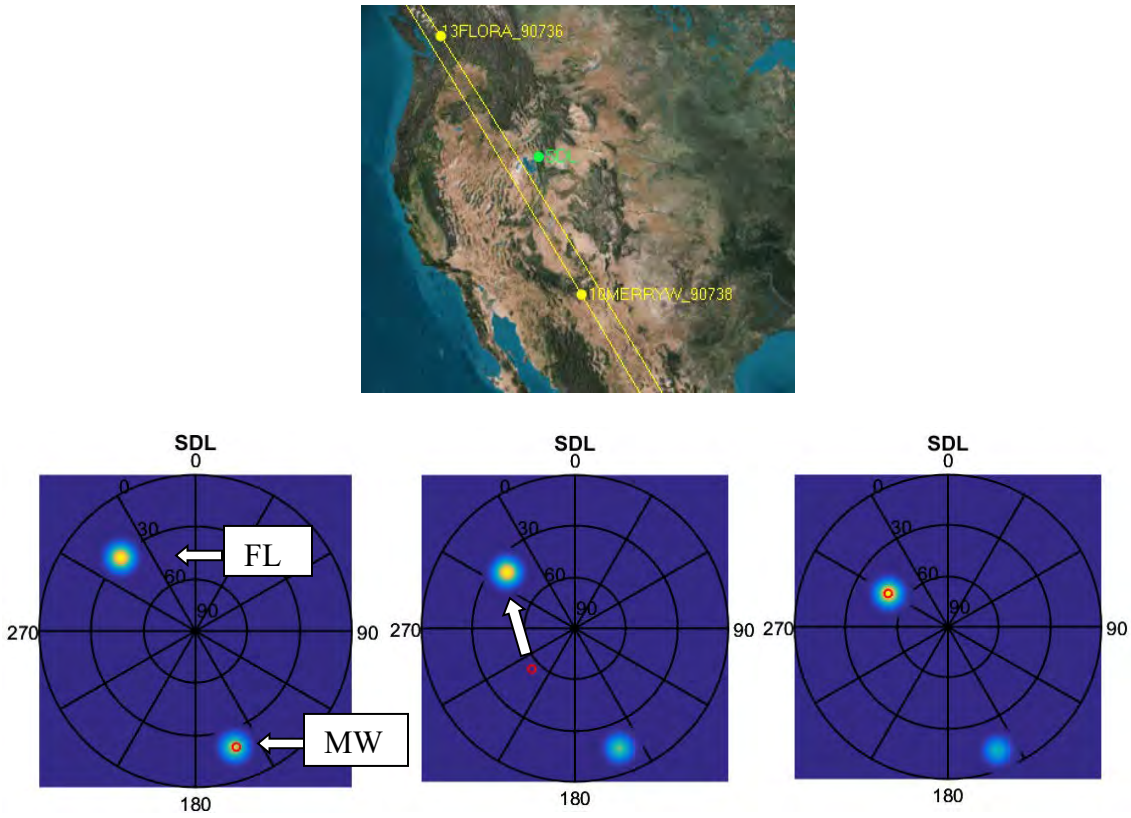
The initial deconfliction model was built with minimized complexity such that it would be possible to verify that the outputs were indeed optimal. This was particularly useful for determining model accuracy and associated scaling factors. The simplest deconfliction model consists of two satellites that are simultaneously in view of a single ground station.

The following example illustrates an equally weighted BVF for both satellites with link margin as the only variable. The benefit value of each satellite is therefore greatest at its closest approach. The two satellites in this scenario are separated by approximately five minutes. The station antenna tracks one satellite until doing so is less valuable than transitioning to the trailing satellite. Figure 80, Figure 81, and Figure 82 show a progression of this scenario. Each figure shows a realistic snapshot from STK (left) and the associated Cartesian azimuth/elevation polar plot. The polar plot displays Gaussians whose color intensity corresponds to each satellite's BVF, as well as a red circle indicating the antenna target location.



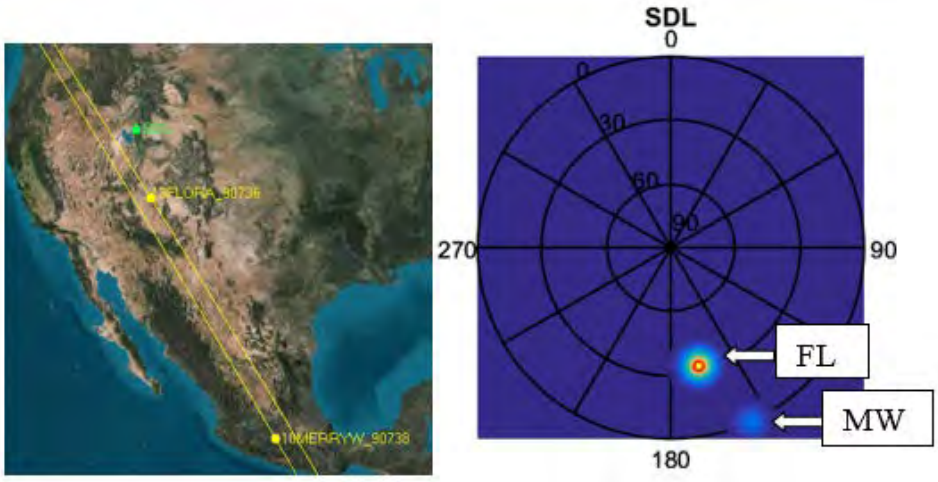
Merryweather satellite in view of SDL station. Flora is on the horizon and provides a much lower benefit than Merryweather. The antenna therefore tracks Merryweather.

Figure 80. Scenario start.



The station transitions from tracking Merryweather to Flora. The decreasing benefit of Merryweather has been surpassed by the increasing benefit from Flora.

Figure 81. Scenario midpoint.



Merryweather below horizon. Flora tracked to loss of signal.

Figure 82. Scenario end.

This scenario illustrates a simple example to validate algorithm development. The optimal solution is intuitive; to maximize benefit, a station should track a satellite until it becomes more advantageous to switch to the other. Figure 83 shows the calculated link margin and superimposed total benefit accrued for the scenario.

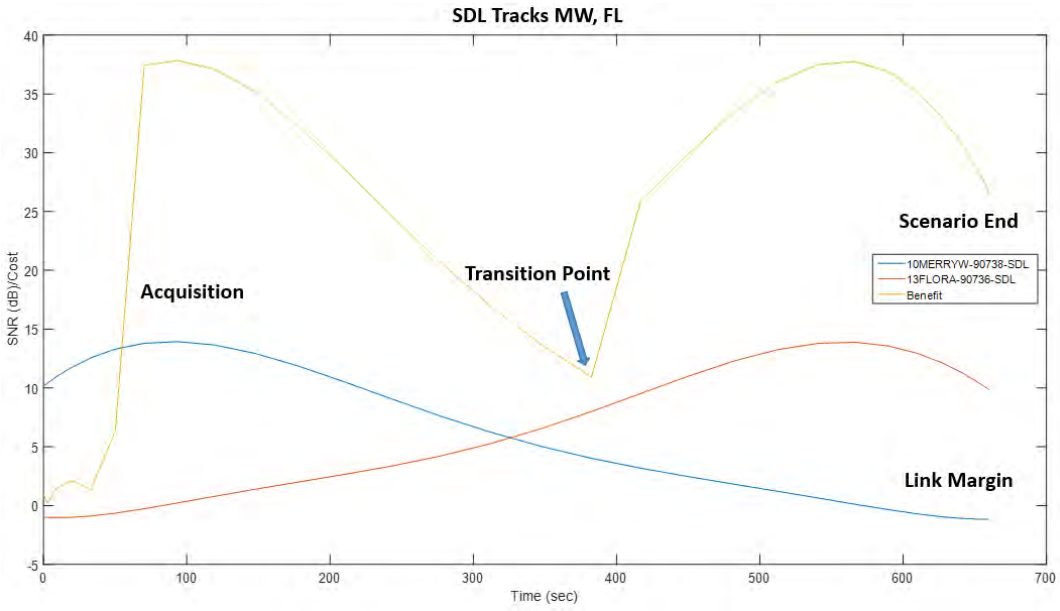


Figure 83. Link margin and benefit plot for the two satellite, one station example.

With all other weight factors kept equal, the overall benefit is maximized when the antenna tracks the peak signal strength from each satellite. The benefit dip in the middle of the scenario corresponds to no benefit being accrued in the transition between satellites. It is therefore advantageous to minimize this transition time as much as possible.

2. Two Satellites, Two Ground Stations

To compare and contrast the above example, we now introduce the exact same scenario but with the addition of a second ground station, UNM, located in Albuquerque, NM. The model, associated value weights, and problem scaling remain the same. Additionally, there is no constraint requiring that only one ground station service one

satellite at a time. This allows antennas at both sites to simultaneously point at a single satellite if it maximizes the benefit. Though only one station at a time can transmit, both can passively receive and forward data to the operator. Future missions may require different constraints, which can be factored into a more complex model.

As Merryweather flew past the SDL site, it continued to approach UNM. At this point, as with the above single station example, SDL transitioned to Flora. UNM continued tracking Merryweather as it continued to be most beneficial. Once SDL fully acquired the Flora target, UNM then began the slew to acquire Flora. The optimizer chose the transition points that maximized the combined benefit from tracking the two satellites. Only one transition occurred at a time, minimizing downtime.

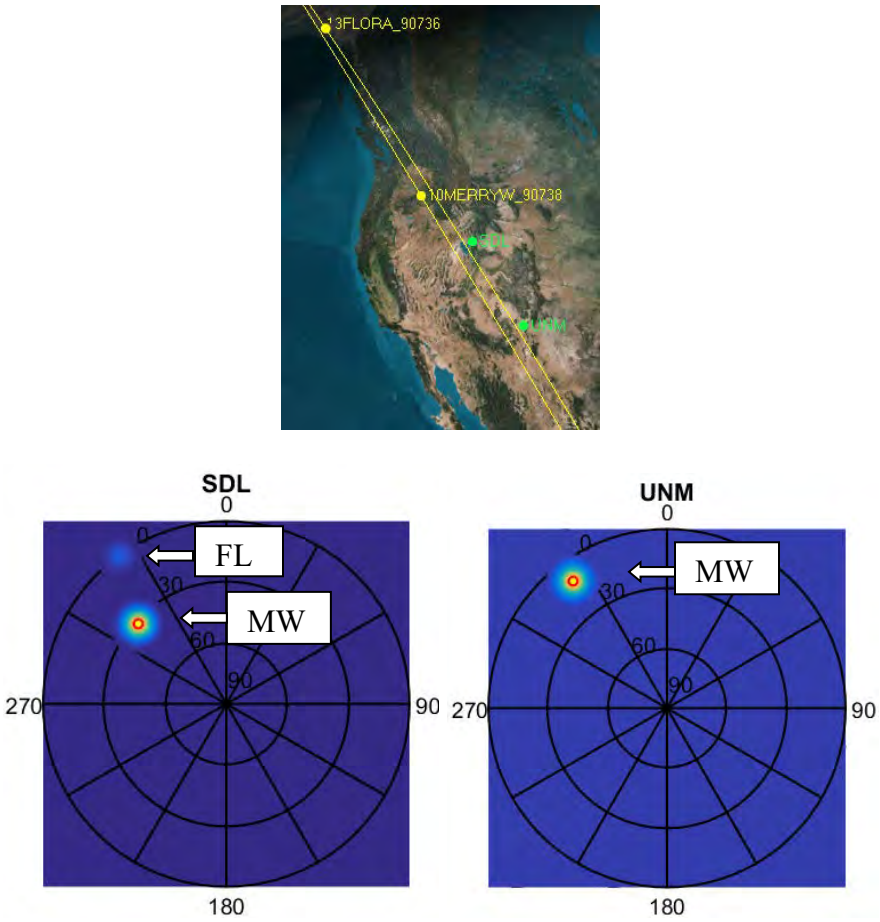


Figure 84. Scenario start: Merryweather in view for both sites. Flora is on horizon for SDL and out of view for UNM. Both sites are tracking Merryweather.

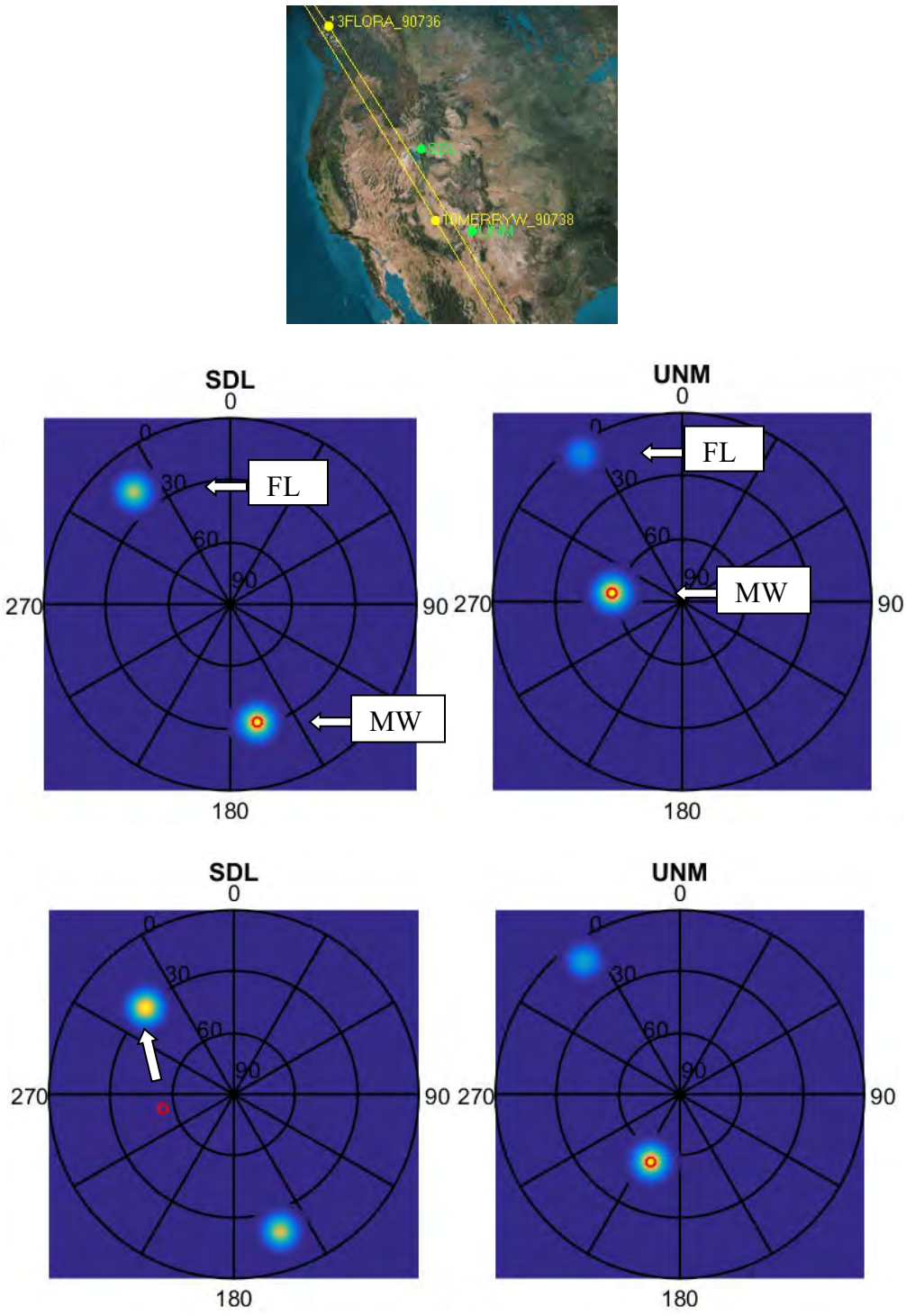


Figure 85. First transition: The SDL site slews from the Merryweather to Flora target. UNM continues tracking Merryweather.

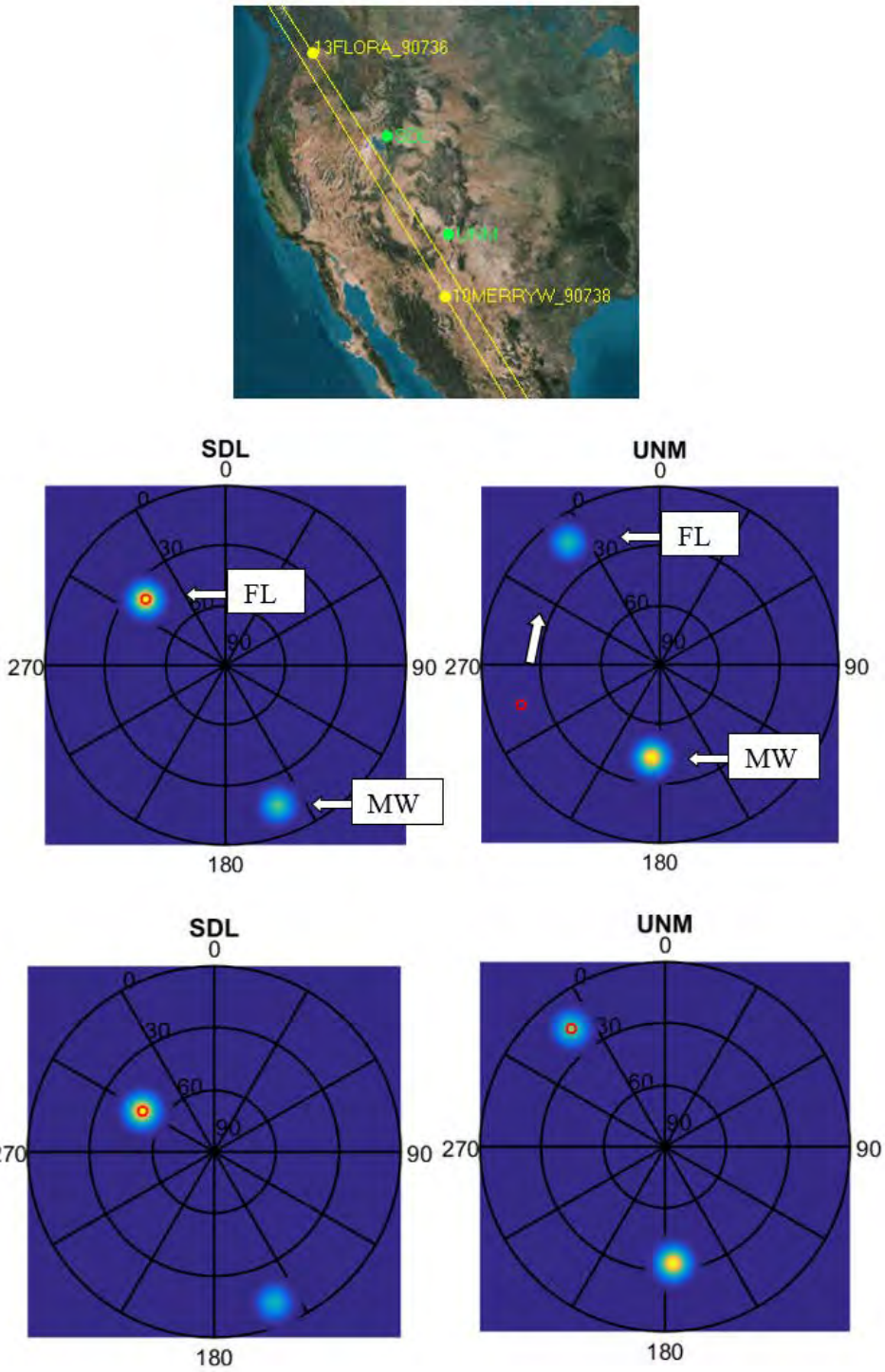
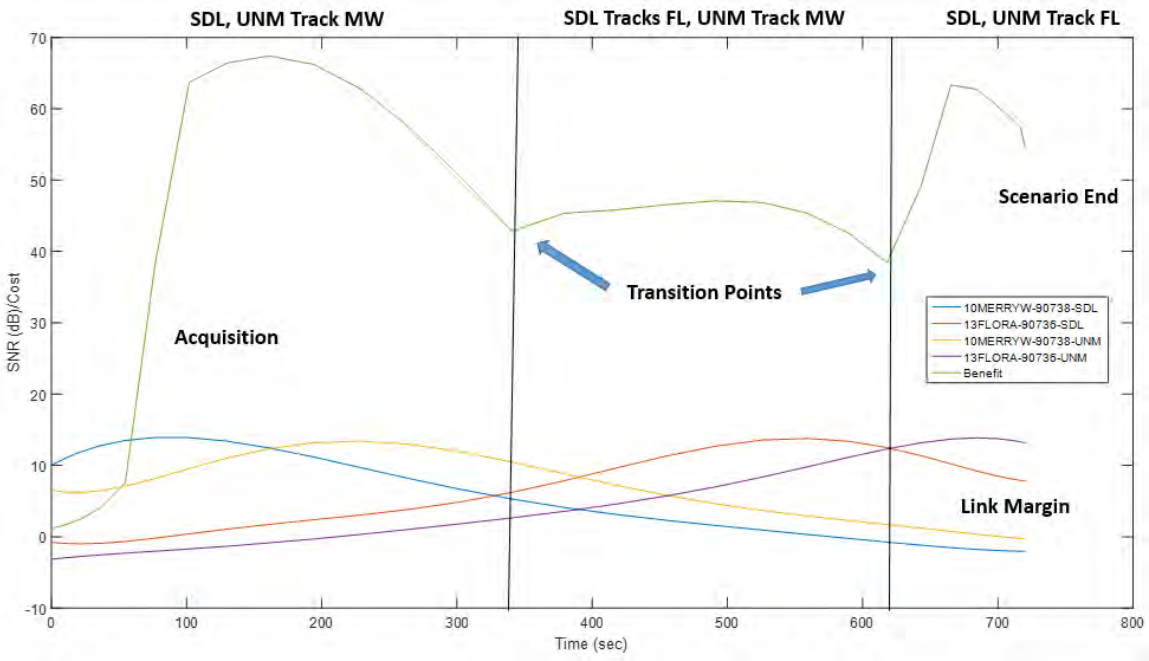


Figure 86. Second transition: Immediately after Flora was acquired at SDL, the UNM site began its slew from Merryweather to Flora. The Flora satellite was then tracked by bo5th stations until the end of the scenario.

Figure 87 displays the calculated link margins and overall mission benefit for both satellites at each ground site.



Note the two dips in benefit corresponding to antenna slews at each station between the two satellites.

Figure 87. Link margin and benefit plot for the two-satellite, two-station example.

Even a simple problem, such as this example, becomes increasingly complex if attempting to plan manually or with a heuristic. The optimizer's solution of only performing one slew at a time between satellites was an unexpected but logical result. An operator could have easily missed an opportunity to extend the UNM station's Merryweather collection slightly longer to deliver the overall mission the most benefit.

These solutions were implemented using the MC3 ground station network to aid real-time operations for the Flora and Merryweather satellites. Both the SDL and UNM sites were utilized in these real demonstrations. Figure 88 shows a screenshot of the real-time STK animation and adjacent video streams of the antennas performing tracks at SDL and UNM.



Video feeds of the antennas are sent to operators at the MC3 satellite operations center. UNM and SDL antennas (framed with red boxes) are tracking the two satellites, whose positions are shown on the left.

Figure 88. Screenshot of actual PROPCUBE operations performed using SDL and UNM ground stations.

The video feeds and STK animations are useful for providing situation awareness to MC3 operators as they conduct passes. While monitoring the contact, operators will also refer to the received signal waterfall plots, such as the one shown in Figure 76. The final figure of merit useful for evaluating the quality of the contacts is the number of decodes and preambles detected by the SDRs installed at the ground stations.

3. Crossing Orbits

The following example describes a two-satellite, one ground station scenario involving crossing orbits, presented in Figure 89. Both Fauna, and later Flora, flew over the UNM ground station at overlapping times. The added complexity in this scenario involved Fauna's northerly crossing of the UNM station. The station's azimuth rotator gearbox has a lockout at 0° (true north) and a total range of 360° due to cable wrap and software limits. These constraints could prevent smoothly tracking northerly passes. To mitigate this constraint, the elevation rotor's range spans from 0–180° allowing the inversion of both azimuth and elevation axes to perform northerly tracks without encountering the lockout. The inversion subsequently limits southerly passes (such as Flora's in this example), therefore, the rotor axes would need to be quickly reverted to their

typical orientation before servicing Flora. Automating this process is routine for a single satellite contact but complex when combined with a second target.



Figure 89. Two satellite, one ground station example with crossing orbits.

When setting up the problem for a DIDO-generated solution, it was only necessary to constrain the total range of motion for the antenna system. The MC3 system requires a 0–360° azimuth constraint and 0–180° in the elevation axis. As each target Gaussian moved across the map, the optimization algorithm determined whether to invert the antenna axes to satisfy the antenna position constraints. The resulting trajectory is not immediately intuitive yet still valid for performing this de-confliction. Figure 90 shows the satellite positions and superimposed DIDO-generated solution interpolated to one point per second, as executed on the MC3 system. The antenna directions required inversion to track Fauna across the lockout zone. The trajectory then resumed non-inverted directions after the transition to Flora.

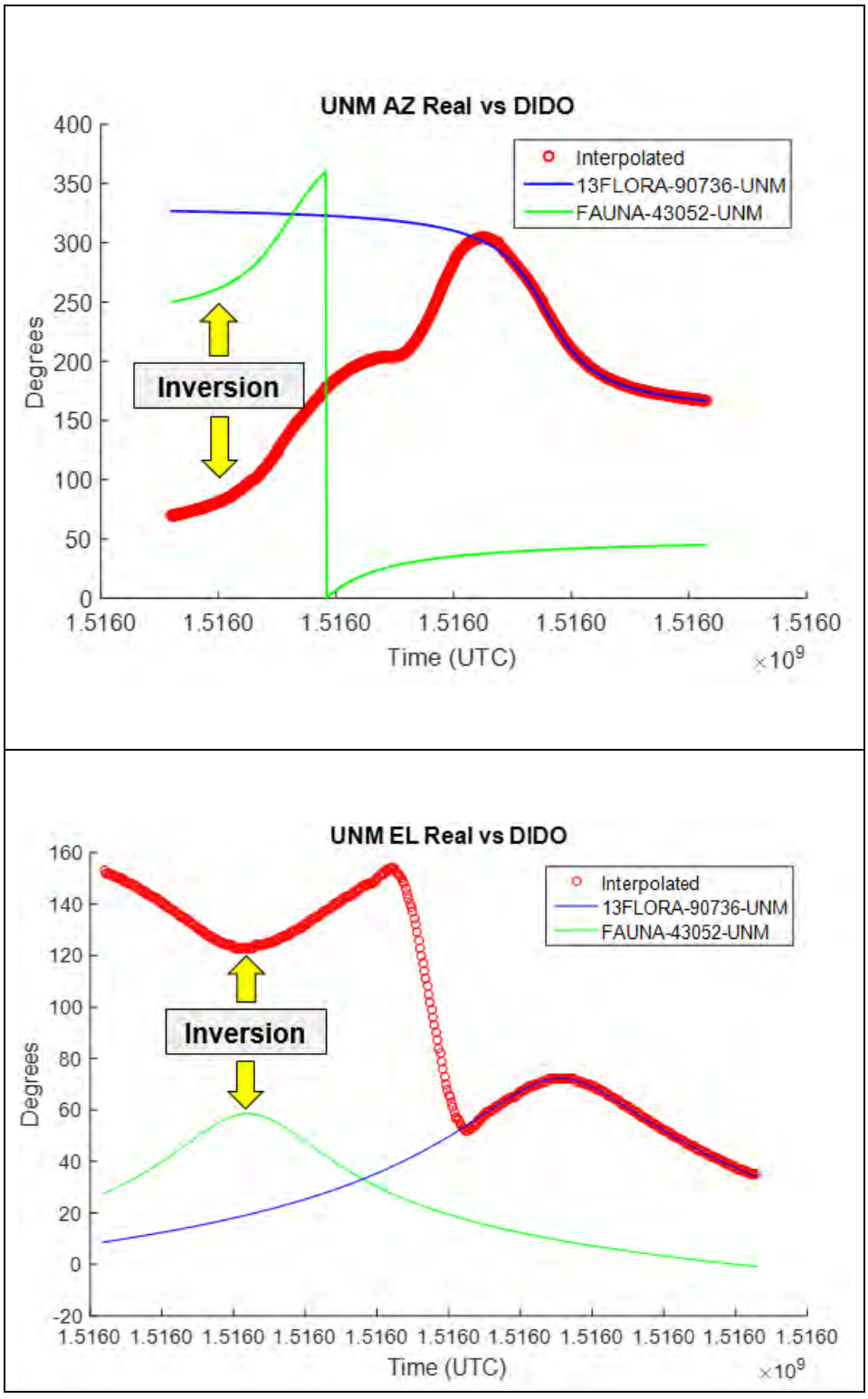


Figure 90. Trajectory plots for azimuth and elevation axes in crossing orbit scenario with automatic pointing inversion.

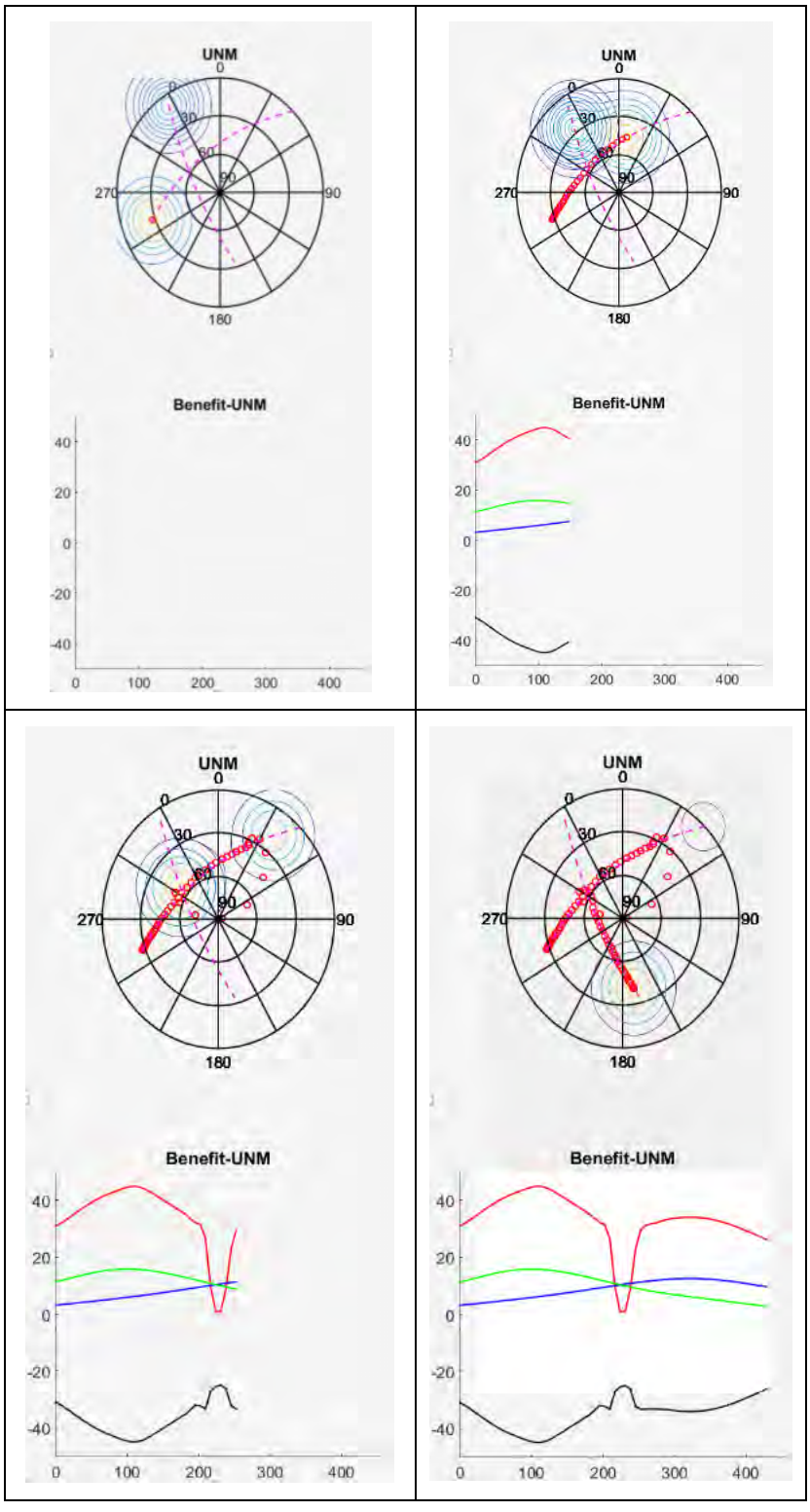


Figure 91. Four-panel progression of crossing orbit scenario.

4. Three Satellites, Two Antennas

Another useful demonstration involved the addition of a third satellite. All three PROPCUBE CubeSat orbits occasionally (though rarely) precess into simultaneous view of one or more ground stations, allowing for real-world testing of the three-satellite deconfliction algorithm. The following test utilized the HSFL ground station and two separate antennas co-located at the site. The objective was to demonstrate that multiple antennas with different configurations could be driven with optimizer-generated trajectories in a three-satellite scenario. One antenna was used for a dedicated uplink and the other for dedicated downlink. Each antenna was connected to a software-defined radio, and the remote ground station computer ran a mission-specific GNU Radio flowgraph customized to communicate with the two SDRs. During the contact, the MPIPE script governing the rotor positioning and Doppler shift corrections also sent messages to the Satnet C2 software, automatically reprogramming the FEP depending on which satellite was being tracked. As a new target was acquired, Satnet could therefore automatically start the pre-loaded command list unique to each satellite for the duration that it was tracked.



Screenshot from the test shows real-time STK animation (left), video feed (top right), Satnet C2 software (bottom middle), and GNU Radio waterfall plots for downlink and uplink (bottom right).

Figure 92. Three satellite/two antenna test at HSFL.

The automatic configuration messages sent to the Satnet C2 software resulted in continuous transitions between all three satellites. The satellite-specific command lists were executed independently of the antenna positioning scripts. This effectively created a bent-pipe system, in which optimal trajectories slewed the antennas and automatic scripts configured radios and amplifiers, enabling the radios, antennas, and supporting network to pass any messages sent either to or from the satellites. Though this test only utilized one C2 software package for all three satellites, future instantiations of this research should develop a standard messaging protocol that can inform multiple C2 systems belonging to different stakeholders when it is time to command their respective vehicle(s), and when the antennas have moved to the next satellite for servicing.

Counting the total number of packets decoded and preambles detected provided an objective measurement of contact quality. Preambles represent unique transmissions from the satellite indicating the beginning of a downlinked data packet. A fully decoded packet means that the data were successfully transmitted and verified with a cyclic redundancy check (CRC). Since there is no bit error correction in the PROPCUBE packet structure, it is difficult to decode packets due to surrounding interference in the 902–928 MHz band, which easily overwhelms the faint downlink, causing flipped or missing bits in the detection routines of the SDR.

The total contact duration for this demonstration was 1105 seconds. In this time, six packets in total were decoded and 125 preambles detected. Two decodes and 47 preambles came from Merryweather, one decode and 15 preambles from Fauna, and 3 decodes and 63 preambles from Flora. These statistics represent an average performance for contacts using the HSFL station. Occasionally there are more decoded packets, though often there are fewer. The results suggest that there was no noticeable degradation in performance when using the optimized antenna slews, and the ability to autonomously service three satellites in under 19 minutes from a single ground station can dramatically increase the capacity of small satellite ground networks.

VI. CONCLUSION

This dissertation presented a new way of approaching the "many satellites, few ground stations" problem. Leveraging techniques used in optimal control allowed the development of new equations, creating a model that scales to an arbitrarily large population of small satellite constellations that rely on distributed ground station networks. The satellite targeting parameters reflect both physical behaviors and qualitative arbitration decisions that human operators commonly use.

This research specifically addressed the development of the problem formulation, algorithms, and techniques to optimally slew ground-based antennas between multiple satellites that are simultaneously in view of one or more ground stations. Using models that Chapter II developed, Chapter III formulated and solved a corresponding dynamic optimization problem. The problem formulation was then solved using pseudospectral optimal control theory to produce deconflicted ground antenna slew trajectories as a function of parameters and constraints used commonly by satellite operators. These parameters included space-to-ground link budgets, mission priority, asset availability, and onboard health. Traditional methods employ heuristics to generate a subset of available targets and a separate process to check feasibility of the solution. However, the method this dissertation describes deterministically solves the problem in a single step.

Chapter IV presented validation examples using simulated scenarios. This enabled real-world implementation in Chapter V, which integrated the optimization solution into the MC3 ground station system and tested it with up to three orbiting CubeSats. The results of this work enabled a proof-of-concept demonstration showing how low-cost, rapidly reconfigurable ground stations can provide communications services to large amounts of diverse small satellites. When centrally coordinated with a tool that automatically deconflicts and efficiently allocates station and SOC resources, human operators are left to guide the system "on-the-loop" rather than being forced to drive the system "in-the-loop." This dissertation only begins to address the potential for automation when concerning communications with many complex targets, each with unique requirements. The

following section will discuss future work that enables continued progress in this compelling field. Then, the succeeding section presents how and where this work may impact certain applications and their CONOPS.

A. FUTURE WORK

1. Automatic Scaling and Balancing

When running the optimization software for increasing numbers of satellite targets and ground stations, the processing time varied from several minutes for less than five satellites, to several hours for ten targets and two stations or more when using a modern computer with average performance. If the problem is not well scaled, the solution does not converge quickly and the software continues to iterate for prolonged periods until it reaches an extremal. When changing the target quantities, ground stations considered, or mission-specific scaling in the Gaussian terms, the problem would often need to be re-scaled to reduce processing time. Since scaling is an iterative process, the first few tries may take substantially longer, thus negating any advantage in automatically deconflicting the event.

One approach may be to create a look-up table routine that automatically applies scaling parameters specific to the problem, categorized by number of satellites and ground stations. Generating this look-up table for each class of problem may be time-consuming initially, but it would save substantial time in routine operations.

A second approach involves bootstrapping the solution from a reduced problem, and using this solution to compute the appropriate scaling parameters. The scaling parameters are then applied to the full-complexity problem. Finally, the solution to the initial problem is used as a guess for the full problem, resulting in fewer iterations to compute the solution.

2. Binary Variables and Extended Testing with Narrow-Beam Dishes

The Yagi antennas used in initial demonstrations have approximately 15° half-power beamwidths. The generated trajectories were shown to be within one degree of

propagated satellite positions and therefore sufficient for the available ground station hardware. The next phase of this research should involve the inclusion of higher gain parabolic dishes with half-power beamwidths of three degrees or less. Several steps are necessary to accomplish high-precision solutions of sub-degree accuracy. The number of nodes should be increased to provide smaller errors in the interpolation of the trajectory as fed into the antenna control system. Increasing nodes will increase processing time, further highlighting the need for proper scaling, as discussed in the previous section.

The inclusion of a binary variable to turn on/turn off contributions from other Gaussians will be necessary so only one target at a time is considered. The formulation, as presented in Chapter III, sums all available targets together, achieving the maximum benefit when the antenna targets the peak of the available problem space. If only one satellite is in a particular location, the peak corresponds to the satellite with most benefit. However, if two satellites are nearby, or flying in formation, the antenna will point to the midpoint of the overlapping Gaussians since this corresponds to the highest peak. To force only one contribution at a time, a binary variable can be used to "turn off," or multiply by zero, the Gaussian not beneficial for tracking. A variation of this method could scale down, but not completely eliminate the Gaussian contributions of other targets.

Similarly, the utilization of binary variables enables additional problem constraints to increase real-world relevance for missions operated by the network. The minimum contact time imposed for each satellite is one such constraint. This mission-specific value guarantees network users enough time to execute a series of commands and attempt data downlinks before the antenna moves to the next target. This constraint would fundamentally prevent the optimization algorithm from initiating contacts on the order of a few seconds while the antennas are enroute to tracking a more beneficial target. This would also benefit the satellites utilizing the network by not draining onboard resources for contacts of little value.

Binary variables are discontinuous, either assuming a value of zero or one. The optimization software used in this research requires smooth functions, therefore requiring mathematical analogs with similar characteristics. One approach for accomplishing this

was described in [54] where a bump function, and its offshoot, the step function, produce a smooth equivalent of the binary variable with a small mathematical penalty. Integrating these smooth functions into the problem formulation will allow the optimization software to appropriately scale the ν_{ij} term of the Gaussian BVF as described in Equation (32).

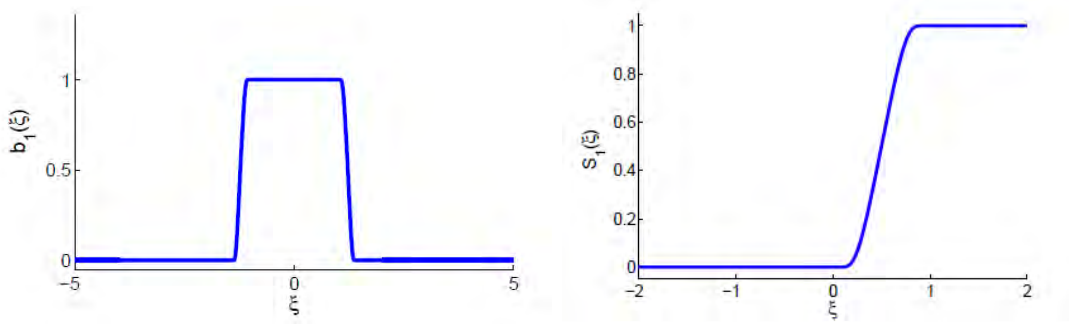


Figure 93. Infinitely differentiable bump (left) and step (right) functions. Source: [54].

After improving the accuracy of the solution to sub-degree precision, high gain (and high value) antennas can be included in the technical demonstrations of the optimal trajectory algorithms. This research can benefit existing high-value infrastructure, like that found on the NASA NEN and AFSCN networks, by providing trajectories formatted specifically for the respective antenna control units. Additional integration is necessary to automatically switch station configurations between multiple targets, and initiate the appropriate C2 software for the different targets. Creating a standard messaging format for diverse C2 packages providing information related to automated bent-pipe communications could ease future integration efforts. The messages would contain information such as satellite and ground station identifiers, network message routing details (such as IP addresses of the ground station(s) in use), pass geometry details, and start/stop times of the contact.

The X/Y rotator arrangement has become increasingly common for moderately sized antennas. This research addressed AZ/EL rotor configurations, which have applications outside of satellite tracking including rocket/missile launches and aircraft

tracking. The AZ/EL system is capable of high azimuth rotation rates; however, it is prone to keyholing near the maximum elevation of the satellite contact, reducing communication performance and increasing wear of the rotors. The X/Y positioners mitigate the keyholes at the expense of rapid slewing. Using the X/Y orientation, substantially slower slew rates are required in the satellite tracking application than those of aerial vehicles. A kinematic transformation is required to move from the topocentric ENU frame (Chapter II) used to compute the AZ/EL angles into corresponding joint angles used in the X/Y system. Incorporating this transformation into the problem formulation will enable the use of an entire class of medium-sized parabolic dishes, which are increasingly common in small satellite operations.

3. Mapping of Mission Parameters into Target Gaussian Modeling

To make the tools developed in this research most applicable for day-to-day operations, mapping the parameters specified in the problem formulation of Chapter III to real-world decision-making allows for a model of higher fidelity, and therefore higher quality autonomy. Human operators often have to employ qualitative arbitration criteria when making decisions regarding orbiting vehicles. Some proposed examples of utilizing the model for such operations follow.

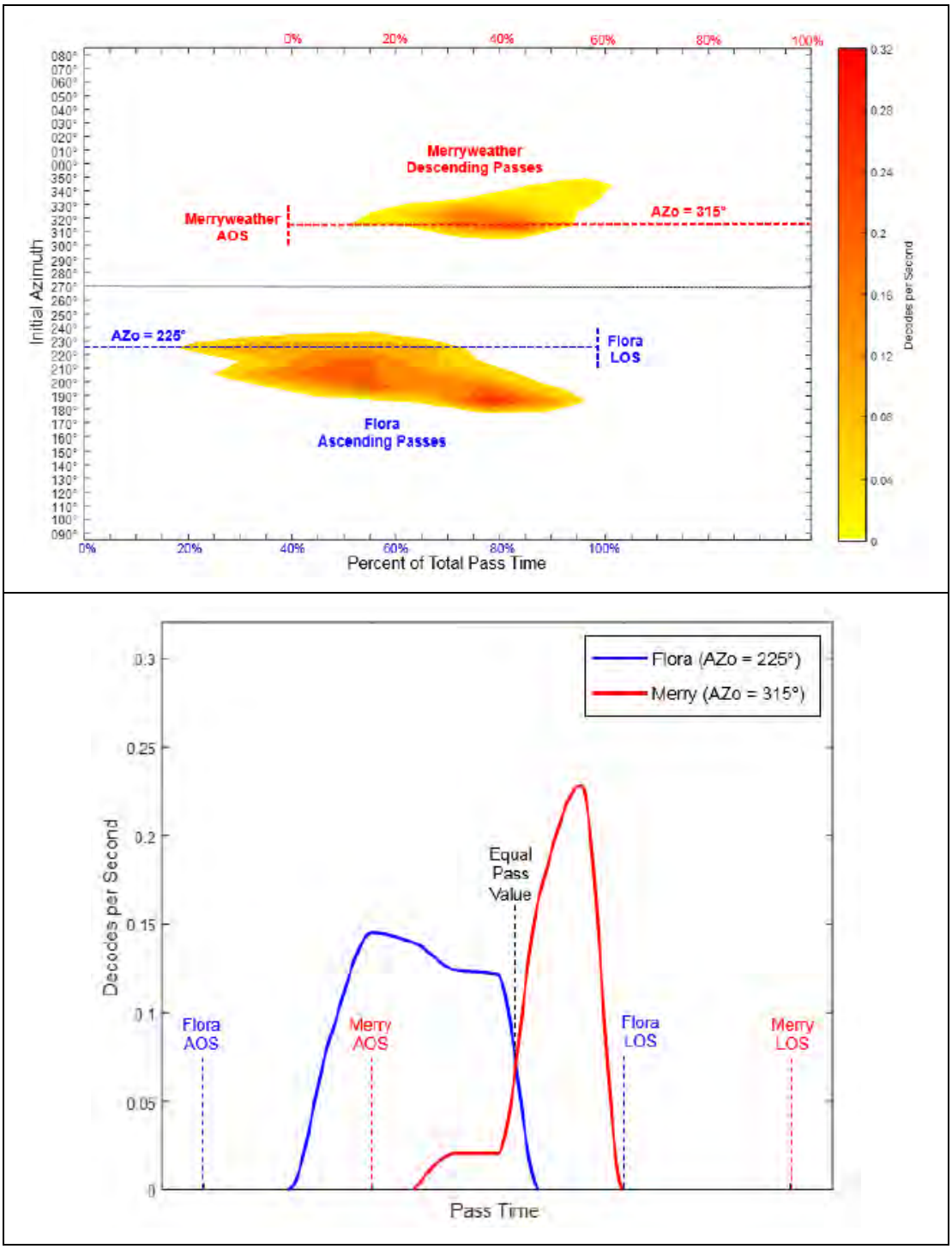
Ground networks may employ a tiered system of user prioritization when operating spacecraft. In the private sector, this can translate to higher paying customers receiving priority on ground assets in the event of conflicting needs. For government or military operations, the value that the mission provides may dictate the prioritization. This ranking system can be captured in scaling the ν_{ij} term of the Gaussian BVF from Equation (32). A higher paying customer, or higher rank stakeholder, would therefore be assigned a scaling factor greater than others would. This would preferentially weigh contact time towards those spacecraft. The scaling factor can be changed by operators depending on the evolving needs of the mission.

Spacecraft health can vary and anomalies may require rapid operator intervention. In a system of hundreds or thousands of orbiting vehicles, it may not be possible to take

over a ground site for immediate commanding due to potential impact to existing plans of other missions. Expanding the $\sigma_{x,y}$ Gaussian footprint terms makes the ground segment favor communicating with the ailing vehicle sooner. It may also prioritize geometries that may not be typically advantageous, such as lower elevation angles. Combining the larger footprint with a higher value v_{ij} scaling increases the consideration of the target, giving operators more time for bent-pipe communications to resolve the anomaly.

Downlink margins involving small satellites are typically the limiting factor for communications due to reduced onboard power, lower-cost (lower gain) antennas, and restrictions in licensing space-ground transmission power and bandwidth. This makes the downlink particularly susceptible to poor geometries and RF interference on the ground. There may be contacts whose poor performance is counterintuitive; the expectation of data transfer was not met due to factors such as satellite orientation (*i.e.,* passively magnetically stabilized CubeSat antennas pointing inefficiently), and terrestrial obstructions or RF interferers. Spending valuable ground resource time on these contacts may take away from other, more advantageous passes.

Leveraging predictive modeling from historical ground station and satellite performance data allows for additional realism in the optimization model. Research conducted in [103] mapped downlink performance for the MC3 stations and PROPCUBE satellites. Using these mappings, a predictive model was developed for each satellite/ground station pairing depending on the ground track geometry of the pass. Using the example case shown in 0, the historical downlink rates for Flora and Merryweather depended on the initial azimuth angle and percentage of contact time (unless the orbit changes, there is a fixed relationship between initial azimuth, maximum elevation, final azimuth, and pass duration).



Total performance shown (top) and pass-specific cross-section (bottom).

Figure 94. Flora/Merryweather predictive model for PTSUR ground station based on initial azimuth angle and historical downlink performance. Source: [103].

Directly scaling the ν_{ij} term of the Gaussian BVF is possible using the cross section plot in 0, rather than solely relying on estimated link margins as done in this dissertation's proof-of-concept. This would place a strong emphasis on contacting the satellite when, statistically speaking, there is a greater likelihood of successfully downlinking data from it. Collecting these data requires many contacts, and the spacecraft would need to have been in stable operation on the order of weeks or months. This model would not be used as part of early operations and checkout activities, but could be highly advantageous when guiding the communications activities of a massive constellation day-to-day.

B. PROPOSED CONOPS FOR FUTURE APPLICATIONS

The following section covers areas of research that can directly benefit from the techniques developed in this dissertation.

1. Satellite Tracking Problem

Much like the cost of the small satellite has fallen dramatically in recent years, so too has the cost of a ground station. However, the performance for both small satellites and their ground stations have increased notably in recent years, with more vendors aiming their development efforts in producing quality equipment at lower costs. The small satellite industry is close to actualizing several new paradigms. A shift away from traditional ground segment design is potentially one of those new paradigms.

The same design philosophies that enable constellations of small satellites to revolutionize revisit rates around the planet without replacing the performance of their larger counterparts, apply to a network of distributed low-cost ground stations to increase contact opportunities without making obsolete existing high value antenna assets. High value ground stations can cost several million dollars or more, while small ground terminals cost between one and two orders of magnitude less. This means that many reduced-performance stations can be built for the price of a single high value station. Due to reduced link margins, the smaller stations will have reduced capacity, thus limiting contact time. However, geographically distributing the stations restores total contact time

capacity and enables the servicing of many satellites in diverse orbits. In other words, a new paradigm for operating a distributed ground segment involves building small terminals anywhere there is available power and an active internet connection. Visibility overlap of multiple stations from orbit is irrelevant because the majority of small satellites only establish two-way closed-link communications when close to the station due to the reduced slant ranges in LEO. This shortens contact times to a few minutes per station; however, with many stations in the field there will soon be another ground resource available, further along in the satellite's ground track, to resume data downlinks. Though each contact would only last minutes, multiplying several contacts around the orbit would actually produce more contact time than a single high value ground station. Operators would also need to wait less time to contact their satellites, as there would be opportunities distributed around the planet.

These low-cost ground stations occupy little space, and could even be manifested on ocean-transiting ships. Power would be supplied from the ship, and the data link to the operators would be provided through existing broadband internet satellite links. At the expense of some latency, this CONOP would open up to 70% of the planet's surface to hosting space-ground communications for a diverse population of small satellites. This application has both commercial and military implications.

Scaling up the ground segment requires substantial automation and a hands-off approach to guiding, rather than driving, the system. Operators should expect only 1–2 minutes of bent-pipe communications from each ground station, and must therefore architect the next generation of C2 software to stream data much how peer-to-peer software passes files across the internet. The source of the data is almost irrelevant to the user, and it is envisioned that small chunks of a larger file stream are passed from multiple sources and reassembled at the operations center. A task list is created to execute commands and request onboard data. Operators would no longer be responsible for planning the contact schedules of their satellites, only assembling their task lists. The ground system automatically arbitrates the short-duration contacts based on the orbit and scaling parameters for each mission's BVF, automatically notifying the operators' C2 software

when it is time to resume their mission-specific tasking. Due to the many available ground stations, there should be small quantities of mission data flowing back to the operations center for the majority of each orbit.

2. Satellite Imaging Problem

This dissertation described an application where orbiting targets rapidly moved past a terrestrial station. In an inertial reference frame, both satellites and ground station were moving, though at substantially different rates. The satellites' equations of motion (and corresponding Gaussians) conformed to Kepler's Laws, while the ground station rotated with the Earth. The propagator utilized the ECEF coordinate frame, as described in Chapter II, and was agnostic to the motions and locations of the targets and antennas.

Inverting this dissertation's application to handle terrestrial satellite imaging is a natural evolution of the work. The orbiting vehicle would be assigned equations of motion conforming to Kepler's Laws while the Gaussian targets on the ground would slowly rotate on the surface of the planet. The state-space representation of the satellite controls and corresponding states would change to accommodate for a three-axis attitude control system and thrusters, reaction wheels, or control moment gyros. Terrestrial Gaussians would still be assigned mission-specific shapes and sizes, thus prioritizing the slewing maneuvers of the orbiting vehicle as its ground track moved across the targets.

3. Autonomous Vehicle Guidance

Targeting algorithms covered by this research need not only apply to space-based applications. Since the algorithms described in this dissertation focus on control inputs to govern the optimal selection of targets of varying benefit, the equations of motion can apply to other autonomous vehicles. For example, a ridesharing/carpooling application may assign its customers certain Gaussian sizes based on their location or membership status. This will dictate the driver's optimal route to these targets with drop-off points using path constraints generated from existing road and highway databases. With the advent of all electric, self-driving vehicles, translating the optimal trajectories into control inputs for these vehicles is a natural extension. Lower level autonomy will handle obstacle avoidance

and navigation. Additional constraints can be included in the problem formulation such as battery charge monitoring and path planning to charging stations.

Since the optimized trajectories are not dependent on equations of motion, these algorithms also apply to future unmanned aerial vehicles such as air taxis. The air taxi's trajectory to highest value Gaussian target passenger is computed based on path constraints such as airways and building locations, as well as altitude and velocity limits. Placing a time constraint in the cost function arrives at destinations and transports passengers in minimum time, and restricting flight time between charges to a maximum value (much like the step function governing minimum communications time in the satellite tracking problem), will give the air taxi sufficient margin to drop off its final passenger before autonomously flying to the nearest available charging station.

C. FINAL SUMMATION

This dissertation presented a compelling proof-of-concept for generating antenna-pointing trajectories in target-rich environments using optimal control methodologies. Additional stakeholders will benefit from the work once the research transitions from technology demonstrations to operational missions. To broaden the reach of this work, the problem formulation and corresponding implementation described in this dissertation have also been filed with the U.S. Patent and Trademark Office, application 62/528,699 (pending) [104]. Augmenting the existing problem formulation along with modifying the equations of motion for other applications will continue to create compelling contributions to the field of autonomous targeting and mission planning.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A.

The following expressions show the full analytic relationship between an orbiting satellite in the ECI frame and the local ground station (X_{SAT}, Y_{SAT}) grid point corresponding to an instantaneous azimuth and elevation position. The satellite position equations were derived with MATLAB's symbolic toolbox after combining Equations (18) and (19). The variables used in the expressions below correspond to variables defined in Equations (10), (11), (13), (14), (16), and (17) as shown in Table 3.

Table 3. Symbol to MATLAB variable mapping.

Symbol	**MATLAB Variable**	**Notes**
t	t	Time elapsed since epoch (sec)
r	r_2D	Orbit radius (km)
$R_{\oplus}$	Re	Earth radius, 6378 km
$\omega_{\oplus}$	rotation_rate	Earth's rotation rate, 7.29×10^{-5} rad/sec
ϕ	lat	Latitude (rad)
λ	lon	Longitude (rad)
μ	mu	Earth's gravitational constant, 398600 km^3/sec^2
Ω	raan	Right ascension of ascending node (rad)
ω	argp	Argument of perigee (rad)

These expressions are specifically shown to illustrate the complexity involved with several coordinate transformations to produce a time-based analytical solution for a two-body orbital analysis. The numerical solution employed for real-world applications utilizes a propagator of higher fidelity; however, there is no guarantee that the solution is valid without a comparison to a known problem. By creating an analytic proxy, it was possible to demonstrate the same behaviors in the V&V of both analytic and numerical methods.

The below expressions for (X_{SAT}, Y_{SAT}) were left in a format that can be copy/pasted into a MATLAB script for future use. These assume a spherical Earth and circular orbit. An effective way to reference time (t = 0) would be to count from an epoch corresponding to Vernal Equinox where the Earth's $\hat{I}_{ECEF}$ axis crosses $\hat{I}_{ECI}$ and is aligned with the Point of Ares γ. The examples in Chapter III utilized an epoch of 20 Mar. 2018 12:00:00 UTC.

x_sat =

(2*(pi/2 + asin((Re*(cos(lon)*(sin(lat)^2 + cos(lat)^2*cos(lon)) - cos(lat)*sin(lon)^2) - r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) + cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) + sin(rotation_rate*t)*cos(lat)*cos(lon)) + (cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) - sin(rotation_rate*t)*cos(lat)*sin(lon)) + sin(argp)*sin(incl)*sin(lat)) + r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) - sin(rotation_rate*t)*cos(lat)*sin(lon)) + (sin(argp)*sin(raan) - cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) + sin(rotation_rate*t)*cos(lat)*cos(lon)) - cos(argp)*sin(incl)*sin(lat)))/((Re*(cos(lon)*(sin(lat)^2 + cos(lat)^2*cos(lon)) - cos(lat)*sin(lon)^2) - r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) +

cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lat)*cos(lon)) + (cos(argp)*cos(raan) -
cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) -
sin(rotation_rate*t)*cos(lat)*sin(lon)) + sin(argp)*sin(incl)*sin(lat)) +
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) +
cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) -
sin(rotation_rate*t)*cos(lat)*sin(lon)) + (sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lat)*cos(lon)) - cos(argp)*sin(incl)*sin(lat)))^2 +
(Re*(cos(lon)*sin(lon) + cos(lat)*cos(lon)*sin(lon)) -
r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) +
sin(rotation_rate*t)*cos(lon))*(cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan)) -
(cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))) +
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) +
sin(rotation_rate*t)*cos(lon))*(sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan)) -
(cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))))^2 + (Re*(cos(lon)*(cos(lat)*sin(lat) -
cos(lat)*cos(lon)*sin(lat)) + sin(lat)*sin(lon)^2) -
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) +
cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) -
sin(rotation_rate*t)*sin(lat)*sin(lon)) + (sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lon)*sin(lat)) + cos(argp)*cos(lat)*sin(incl)) +
r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lon)*sin(lat)) + (cos(argp)*cos(raan) -
cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) -
sin(rotation_rate*t)*sin(lat)*sin(lon)) -
cos(lat)*sin(argp)*sin(incl)))^2)^(1/2)))*(real((Re*(cos(lon)*sin(lon) +

cos(lat)*cos(lon)*sin(lon)) -
r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) +
sin(rotation_rate*t)*cos(lon))*(cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan)) -
(cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))) +
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) +
sin(rotation_rate*t)*cos(lon))*(sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan)) -
(cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))))/((Re*(cos(lon)*(sin(lat)^2 + cos(lat)^2*cos(lon)) -
cos(lat)*sin(lon)^2) - r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lat)*cos(lon)) + (cos(argp)*cos(raan) -
cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) -
sin(rotation_rate*t)*cos(lat)*sin(lon)) + sin(argp)*sin(incl)*sin(lat)) +
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) +
cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) -
sin(rotation_rate*t)*cos(lat)*sin(lon)) + (sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lat)*cos(lon)) - cos(argp)*sin(incl)*sin(lat)))^2 +
(Re*(cos(lon)*sin(lon) + cos(lat)*cos(lon)*sin(lon)) -
r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) +
sin(rotation_rate*t)*cos(lon))*(cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan)) -
(cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))) +
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) +
sin(rotation_rate*t)*cos(lon))*(sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan)) -
(cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))))^2 + (Re*(cos(lon)*(cos(lat)*sin(lat) -
cos(lat)*cos(lon)*sin(lat)) + sin(lat)*sin(lon)^2) -
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) +

cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) -
sin(rotation_rate*t)*sin(lat)*sin(lon)) + (sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lon)*sin(lat)) + cos(argp)*cos(lat)*sin(incl)) +
r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lon)*sin(lat)) + (cos(argp)*cos(raan) -
cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) -
sin(rotation_rate*t)*sin(lat)*sin(lon)) - cos(lat)*sin(argp)*sin(incl)))^2)^(1/2)) -
imag((Re*(cos(lon)*(cos(lat)*sin(lat) - cos(lat)*cos(lon)*sin(lat)) + sin(lat)*sin(lon)^2) -
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) +
cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) -
sin(rotation_rate*t)*sin(lat)*sin(lon)) + (sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lon)*sin(lat)) + cos(argp)*cos(lat)*sin(incl)) +
r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lon)*sin(lat)) + (cos(argp)*cos(raan) -
cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) -
sin(rotation_rate*t)*sin(lat)*sin(lon)) -
cos(lat)*sin(argp)*sin(incl)))/((Re*(cos(lon)*(sin(lat)^2 + cos(lat)^2*cos(lon)) -
cos(lat)*sin(lon)^2) - r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lat)*cos(lon)) + (cos(argp)*cos(raan) -
cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) -
sin(rotation_rate*t)*cos(lat)*sin(lon)) + sin(argp)*sin(incl)*sin(lat)) +
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) +
cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) -
sin(rotation_rate*t)*cos(lat)*sin(lon)) + (sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) +

sin(rotation_rate*t)*cos(lat)*cos(lon)) - cos(argp)*sin(incl)*sin(lat)))^2 +
(Re*(cos(lon)*sin(lon) + cos(lat)*cos(lon)*sin(lon)) -
r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) +
sin(rotation_rate*t)*cos(lon))*(cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan)) -
(cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))) +
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) +
sin(rotation_rate*t)*cos(lon))*(sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan)) -
(cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))))^2 + (Re*(cos(lon)*(cos(lat)*sin(lat) -
cos(lat)*cos(lon)*sin(lat)) + sin(lat)*sin(lon)^2) -
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) +
cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) -
sin(rotation_rate*t)*sin(lat)*sin(lon)) + (sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lon)*sin(lat)) + cos(argp)*cos(lat)*sin(incl)) +
r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lon)*sin(lat)) + (cos(argp)*cos(raan) -
cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) -
sin(rotation_rate*t)*sin(lat)*sin(lon)) -
cos(lat)*sin(argp)*sin(incl)))^2)^(1/2))))/(pi*abs((Re*(cos(lon)*(cos(lat)*sin(lat) -
cos(lat)*cos(lon)*sin(lat)) + sin(lat)*sin(lon)^2) -
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) +
cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) -
sin(rotation_rate*t)*sin(lat)*sin(lon)) + (sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lon)*sin(lat)) + cos(argp)*cos(lat)*sin(incl)) +
r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) +

sin(rotation_rate*t)*cos(lon)*sin(lat)) + (cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) - sin(rotation_rate*t)*sin(lat)*sin(lon)) - cos(lat)*sin(argp)*sin(incl)))/((Re*(cos(lon)*(sin(lat)^2 + cos(lat)^2*cos(lon)) - cos(lat)*sin(lon)^2) - r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) + cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) + sin(rotation_rate*t)*cos(lat)*cos(lon)) + (cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) - sin(rotation_rate*t)*cos(lat)*sin(lon)) + sin(argp)*sin(incl)*sin(lat)) + r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) - sin(rotation_rate*t)*cos(lat)*sin(lon)) + (sin(argp)*sin(raan) - cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) + sin(rotation_rate*t)*cos(lat)*cos(lon)) - cos(argp)*sin(incl)*sin(lat)))^2 + (Re*(cos(lon)*sin(lon) + cos(lat)*cos(lon)*sin(lon)) - r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) + sin(rotation_rate*t)*cos(lon))*(cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan)) - (cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(cos(argp)*sin(raan) + cos(incl)*sin(argp)*cos(raan))) + r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) + sin(rotation_rate*t)*cos(lon))*(sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan)) - (cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(sin(argp)*sin(raan) - cos(argp)*cos(incl)*cos(raan))))^2 + (Re*(cos(lon)*(cos(lat)*sin(lat) - cos(lat)*cos(lon)*sin(lat)) + sin(lat)*sin(lon)^2) - r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) - sin(rotation_rate*t)*sin(lat)*sin(lon)) + (sin(argp)*sin(raan) - cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) + sin(rotation_rate*t)*cos(lon)*sin(lat)) + cos(argp)*cos(lat)*sin(incl)) + r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) +

cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lon)*sin(lat)) + (cos(argp)*cos(raan) -
cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) -
sin(rotation_rate*t)*sin(lat)*sin(lon)) - cos(lat)*sin(argp)*sin(incl)))^2)^(1/2) -
((Re*(cos(lon)*sin(lon) + cos(lat)*cos(lon)*sin(lon)) -
r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) +
sin(rotation_rate*t)*cos(lon))*(cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan)) -
(cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))) +
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) +
sin(rotation_rate*t)*cos(lon))*(sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan)) -
(cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))))*1i)/((Re*(cos(lon)*(sin(lat)^2 + cos(lat)^2*cos(lon)) -
cos(lat)*sin(lon)^2) - r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lat)*cos(lon)) + (cos(argp)*cos(raan) -
cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) -
sin(rotation_rate*t)*cos(lat)*sin(lon)) + sin(argp)*sin(incl)*sin(lat)) +
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) +
cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) -
sin(rotation_rate*t)*cos(lat)*sin(lon)) + (sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lat)*cos(lon)) - cos(argp)*sin(incl)*sin(lat)))^2 +
(Re*(cos(lon)*sin(lon) + cos(lat)*cos(lon)*sin(lon)) -
r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) +
sin(rotation_rate*t)*cos(lon))*(cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan)) -
(cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))) +
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) +
sin(rotation_rate*t)*cos(lon))*(sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan)) -

```
(cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))))^2 + (Re*(cos(lon)*(cos(lat)*sin(lat) -
cos(lat)*cos(lon)*sin(lat)) + sin(lat)*sin(lon)^2) -
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) +
cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) -
sin(rotation_rate*t)*sin(lat)*sin(lon)) + (sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lon)*sin(lat)) + cos(argp)*cos(lat)*sin(incl)) +
r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lon)*sin(lat)) + (cos(argp)*cos(raan) -
cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) -
sin(rotation_rate*t)*sin(lat)*sin(lon)) - cos(lat)*sin(argp)*sin(incl)))^2)^(1/2)))
```

y_sat =

-(2*(pi/2 + asin((Re*(cos(lon)*(sin(lat)^2 + cos(lat)^2*cos(lon)) - cos(lat)*sin(lon)^2) - r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) + cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) + sin(rotation_rate*t)*cos(lat)*cos(lon)) + (cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) - sin(rotation_rate*t)*cos(lat)*sin(lon)) + sin(argp)*sin(incl)*sin(lat)) + r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) - sin(rotation_rate*t)*cos(lat)*sin(lon)) + (sin(argp)*sin(raan) - cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) + sin(rotation_rate*t)*cos(lat)*cos(lon)) - cos(argp)*sin(incl)*sin(lat)))/((Re*(cos(lon)*(sin(lat)^2 + cos(lat)^2*cos(lon)) - cos(lat)*sin(lon)^2) - r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) + cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) + sin(rotation_rate*t)*cos(lat)*cos(lon)) + (cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) - sin(rotation_rate*t)*cos(lat)*sin(lon)) + sin(argp)*sin(incl)*sin(lat)) + r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) - sin(rotation_rate*t)*cos(lat)*sin(lon)) + (sin(argp)*sin(raan) - cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) + sin(rotation_rate*t)*cos(lat)*cos(lon)) - cos(argp)*sin(incl)*sin(lat)))^2 + (Re*(cos(lon)*sin(lon) + cos(lat)*cos(lon)*sin(lon)) - r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) + sin(rotation_rate*t)*cos(lon))*(cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan)) - (cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(cos(argp)*sin(raan) + cos(incl)*sin(argp)*cos(raan))) +

r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) + sin(rotation_rate*t)*cos(lon))*(sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan)) - (cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(sin(argp)*sin(raan) - cos(argp)*cos(incl)*cos(raan))))^2 + (Re*(cos(lon)*(cos(lat)*sin(lat) - cos(lat)*cos(lon)*sin(lat)) + sin(lat)*sin(lon)^2) - r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) - sin(rotation_rate*t)*sin(lat)*sin(lon)) + (sin(argp)*sin(raan) - cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) + sin(rotation_rate*t)*cos(lon)*sin(lat)) + cos(argp)*cos(lat)*sin(incl)) + r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) + cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) + sin(rotation_rate*t)*cos(lon)*sin(lat)) + (cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) - sin(rotation_rate*t)*sin(lat)*sin(lon)) - cos(lat)*sin(argp)*sin(incl)))^2)^(1/2)))*(real((Re*(cos(lon)*(cos(lat)*sin(lat) - cos(lat)*cos(lon)*sin(lat)) + sin(lat)*sin(lon)^2) - r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) - sin(rotation_rate*t)*sin(lat)*sin(lon)) + (sin(argp)*sin(raan) - cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) + sin(rotation_rate*t)*cos(lon)*sin(lat)) + cos(argp)*cos(lat)*sin(incl)) + r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) + cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) + sin(rotation_rate*t)*cos(lon)*sin(lat)) + (cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) - sin(rotation_rate*t)*sin(lat)*sin(lon)) - cos(lat)*sin(argp)*sin(incl)))/((Re*(cos(lon)*(sin(lat)^2 + cos(lat)^2*cos(lon)) - cos(lat)*sin(lon)^2) - r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) + cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) +

sin(rotation_rate*t)*cos(lat)*cos(lon)) + (cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) - sin(rotation_rate*t)*cos(lat)*sin(lon)) + sin(argp)*sin(incl)*sin(lat)) + r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) - sin(rotation_rate*t)*cos(lat)*sin(lon)) + (sin(argp)*sin(raan) - cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) + sin(rotation_rate*t)*cos(lat)*cos(lon)) - cos(argp)*sin(incl)*sin(lat)))^2 + (Re*(cos(lon)*sin(lon) + cos(lat)*cos(lon)*sin(lon)) - r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) + sin(rotation_rate*t)*cos(lon))*(cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan)) - (cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(cos(argp)*sin(raan) + cos(incl)*sin(argp)*cos(raan))) + r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) + sin(rotation_rate*t)*cos(lon))*(sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan)) - (cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(sin(argp)*sin(raan) - cos(argp)*cos(incl)*cos(raan))))^2 + (Re*(cos(lon)*(cos(lat)*sin(lat) - cos(lat)*cos(lon)*sin(lat)) + sin(lat)*sin(lon)^2) - r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) - sin(rotation_rate*t)*sin(lat)*sin(lon)) + (sin(argp)*sin(raan) - cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) + sin(rotation_rate*t)*cos(lon)*sin(lat)) + cos(argp)*cos(lat)*sin(incl)) + r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) + cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) + sin(rotation_rate*t)*cos(lon)*sin(lat)) + (cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) - sin(rotation_rate*t)*sin(lat)*sin(lon)) - cos(lat)*sin(argp)*sin(incl)))^2)^(1/2)) + imag((Re*(cos(lon)*sin(lon) + cos(lat)*cos(lon)*sin(lon)) - r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) +

sin(rotation_rate*t)*cos(lon))*(cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan)) -
(cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))) +
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) +
sin(rotation_rate*t)*cos(lon))*(sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan)) -
(cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))))/((Re*(cos(lon)*(sin(lat)^2 + cos(lat)^2*cos(lon)) -
cos(lat)*sin(lon)^2) - r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lat)*cos(lon)) + (cos(argp)*cos(raan) -
cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) -
sin(rotation_rate*t)*cos(lat)*sin(lon)) + sin(argp)*sin(incl)*sin(lat)) +
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) +
cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) -
sin(rotation_rate*t)*cos(lat)*sin(lon)) + (sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lat)*cos(lon)) - cos(argp)*sin(incl)*sin(lat)))^2 +
(Re*(cos(lon)*sin(lon) + cos(lat)*cos(lon)*sin(lon)) -
r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) +
sin(rotation_rate*t)*cos(lon))*(cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan)) -
(cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))) +
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) +
sin(rotation_rate*t)*cos(lon))*(sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan)) -
(cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))))^2 + (Re*(cos(lon)*(cos(lat)*sin(lat) -
cos(lat)*cos(lon)*sin(lat)) + sin(lat)*sin(lon)^2) -
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) +
cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) -
sin(rotation_rate*t)*sin(lat)*sin(lon)) + (sin(argp)*sin(raan) -

cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lon)*sin(lat)) + cos(argp)*cos(lat)*sin(incl)) +
r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lon)*sin(lat)) + (cos(argp)*cos(raan) -
cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) -
sin(rotation_rate*t)*sin(lat)*sin(lon)) -
cos(lat)*sin(argp)*sin(incl)))^2)^(1/2))))/(pi*abs((Re*(cos(lon)*(cos(lat)*sin(lat) -
cos(lat)*cos(lon)*sin(lat)) + sin(lat)*sin(lon)^2) -
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) +
cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) -
sin(rotation_rate*t)*sin(lat)*sin(lon)) + (sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lon)*sin(lat)) + cos(argp)*cos(lat)*sin(incl)) +
r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lon)*sin(lat)) + (cos(argp)*cos(raan) -
cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) -
sin(rotation_rate*t)*sin(lat)*sin(lon)) -
cos(lat)*sin(argp)*sin(incl)))/((Re*(cos(lon)*(sin(lat)^2 + cos(lat)^2*cos(lon)) -
cos(lat)*sin(lon)^2) - r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lat)*cos(lon)) + (cos(argp)*cos(raan) -
cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) -
sin(rotation_rate*t)*cos(lat)*sin(lon)) + sin(argp)*sin(incl)*sin(lat)) +
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) +
cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) -
sin(rotation_rate*t)*cos(lat)*sin(lon)) + (sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lat)*cos(lon)) - cos(argp)*sin(incl)*sin(lat)))^2 +

(Re*(cos(lon)*sin(lon) + cos(lat)*cos(lon)*sin(lon)) - r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) + sin(rotation_rate*t)*cos(lon))*(cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan)) - (cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(cos(argp)*sin(raan) + cos(incl)*sin(argp)*cos(raan))) + r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) + sin(rotation_rate*t)*cos(lon))*(sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan)) - (cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(sin(argp)*sin(raan) - cos(argp)*cos(incl)*cos(raan))))^2 + (Re*(cos(lon)*(cos(lat)*sin(lat) - cos(lat)*cos(lon)*sin(lat)) + sin(lat)*sin(lon)^2) - r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) - sin(rotation_rate*t)*sin(lat)*sin(lon)) + (sin(argp)*sin(raan) - cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) + sin(rotation_rate*t)*cos(lon)*sin(lat)) + cos(argp)*cos(lat)*sin(incl)) + r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) + cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) + sin(rotation_rate*t)*cos(lon)*sin(lat)) + (cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) - sin(rotation_rate*t)*sin(lat)*sin(lon)) - cos(lat)*sin(argp)*sin(incl)))^2)^(1/2) - ((Re*(cos(lon)*sin(lon) + cos(lat)*cos(lon)*sin(lon)) - r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) + sin(rotation_rate*t)*cos(lon))*(cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan)) - (cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(cos(argp)*sin(raan) + cos(incl)*sin(argp)*cos(raan))) + r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) + sin(rotation_rate*t)*cos(lon))*(sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan)) - (cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(sin(argp)*sin(raan) - cos(argp)*cos(incl)*cos(raan))))*1i)/((Re*(cos(lon)*(sin(lat)^2 + cos(lat)^2*cos(lon)) - cos(lat)*sin(lon)^2) - r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) +

cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lat)*cos(lon)) + (cos(argp)*cos(raan) -
cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) -
sin(rotation_rate*t)*cos(lat)*sin(lon)) + sin(argp)*sin(incl)*sin(lat)) +
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) +
cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) -
sin(rotation_rate*t)*cos(lat)*sin(lon)) + (sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lat)*cos(lon)) - cos(argp)*sin(incl)*sin(lat)))^2 +
(Re*(cos(lon)*sin(lon) + cos(lat)*cos(lon)*sin(lon)) -
r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) +
sin(rotation_rate*t)*cos(lon))*(cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan)) -
(cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))) +
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) +
sin(rotation_rate*t)*cos(lon))*(sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan)) -
(cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))))^2 + (Re*(cos(lon)*(cos(lat)*sin(lat) -
cos(lat)*cos(lon)*sin(lat)) + sin(lat)*sin(lon)^2) -
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) +
cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) -
sin(rotation_rate*t)*sin(lat)*sin(lon)) + (sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lon)*sin(lat)) + cos(argp)*cos(lat)*sin(incl)) +
r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lon)*sin(lat)) + (cos(argp)*cos(raan) -
cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) -
sin(rotation_rate*t)*sin(lat)*sin(lon)) - cos(lat)*sin(argp)*sin(incl)))^2)^(1/2)))

APPENDIX B.

The link budget equations (20) - (26) were symbolically computed and are shown in this section. Table 4. introduces several new parameters in addition to Table 3.

Table 4. Symbol to MATLAB variable mapping

Symbol	**MATLAB Variable**	**Notes**
$frequency$	freq	Center frequency of signal (Hz)
P_{TX}	P_RF	Transmitter power (W)
G_{TX}	G_Tx	Antenna gain (dBi)
L_{OUT}	L_out	Output losses (dB)
G_{RX}	G_Rx	Receiver antenna gain (dBi)
L_{ATM}	L_atm	Atmospheric losses (dB)
L_{IN}	L_in	Input losses (dB)
$L_{POINTING}$	L_pointing	Pointing losses (dB)
$L_{POLARIZATION}$	L_polar	Polarization losses (dB)
k	k	Boltzmann's constant $1.38064852 * 10^{-23} \frac{m^2 kg}{\sec^2 K}$
$rate$	rate	Over-air data rate (bps)
T_s	Ts	System noise temperature, typically 220 K
$E_b / N_o _ required$	Eb_No_Req	Required signal to noise ratio

The below expressions are introduced in Chapter III and are intended to be directly copy/pasted into MATLAB for future analysis. The key takeaway is that link margin $M(t)$ and its derivative $M'(t)$ have time dependencies that manifest themselves in the Hamiltonian Evolution Equation leading to a non-zero Hamiltonian.

$M(t)$: link_margin =

G_Rx - Eb_No_Req + G_Tx - L_atm - L_in - L_out - L_polar - L_pointing - (10*log(k))/log(10) - (10*log(rate))/log(10) - (20*log(((Re*(cos(lon)*(cos(lon)*cos(lat)^2 + sin(lat)^2) - cos(lat)*sin(lon)^2) - r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) + cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) + sin(rotation_rate*t)*cos(lat)*cos(lon)) + (cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) - sin(rotation_rate*t)*cos(lat)*sin(lon)) + sin(argp)*sin(incl)*sin(lat)) + r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) - sin(rotation_rate*t)*cos(lat)*sin(lon)) + (sin(argp)*sin(raan) - cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) + sin(rotation_rate*t)*cos(lat)*cos(lon)) - cos(argp)*sin(incl)*sin(lat)))^2 + (Re*(cos(lon)*sin(lon) + cos(lat)*cos(lon)*sin(lon)) - r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) + sin(rotation_rate*t)*cos(lon))*(cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan)) - (cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(cos(argp)*sin(raan) + cos(incl)*sin(argp)*cos(raan))) + r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) + sin(rotation_rate*t)*cos(lon))*(sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan)) - (cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(sin(argp)*sin(raan) - cos(argp)*cos(incl)*cos(raan))))^2 + (Re*(cos(lon)*(cos(lat)*sin(lat) - cos(lat)*cos(lon)*sin(lat)) + sin(lat)*sin(lon)^2) -

r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) - sin(rotation_rate*t)*sin(lat)*sin(lon)) + (sin(argp)*sin(raan) - cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) + sin(rotation_rate*t)*cos(lon)*sin(lat)) + cos(argp)*cos(lat)*sin(incl)) + r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) + cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) + sin(rotation_rate*t)*cos(lon)*sin(lat)) + (cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) - sin(rotation_rate*t)*sin(lat)*sin(lon)) - cos(lat)*sin(argp)*sin(incl)))^2)^(1/2)))/log(10) + (10*log(P_RF))/log(10) - (10*log(Ts))/log(10) - (20*log(freq))/log(10) - 649/20

$M'(t)$: d_link_margin =

-(10*(2*(Re*(cos(lon)*sin(lon) + cos(lat)*cos(lon)*sin(lon)) - r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) + sin(rotation_rate*t)*cos(lon))*(cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan)) - (cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(cos(argp)*sin(raan) + cos(incl)*sin(argp)*cos(raan))) + r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) + sin(rotation_rate*t)*cos(lon))*(sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan)) - (cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(sin(argp)*sin(raan) - cos(argp)*cos(incl)*cos(raan))))*(r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((rotation_rate*cos(rotation_rate*t)*sin(lon) + rotation_rate*sin(rotation_rate*t)*cos(lon))*(sin(argp)*sin(raan) - cos(argp)*cos(incl)*cos(raan)) + (rotation_rate*cos(rotation_rate*t)*cos(lon) - rotation_rate*sin(rotation_rate*t)*sin(lon))*(sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan))) - r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((rotation_rate*cos(rotation_rate*t)*sin(lon) +

rotation_rate*sin(rotation_rate*t)*cos(lon))*(cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan)) + (rotation_rate*cos(rotation_rate*t)*cos(lon) -
rotation_rate*sin(rotation_rate*t)*sin(lon))*(cos(argp)*cos(raan) -
cos(incl)*sin(argp)*sin(raan))) +
(mu^(1/2)*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) +
sin(rotation_rate*t)*cos(lon))*(sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan)) -
(cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))))/r_2D^(1/2) +
(mu^(1/2)*sin((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) +
sin(rotation_rate*t)*cos(lon))*(cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan)) -
(cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))))/r_2D^(1/2)) + 2*(Re*(cos(lon)*(cos(lat)*sin(lat) -
cos(lat)*cos(lon)*sin(lat)) + sin(lat)*sin(lon)^2) -
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) +
cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) -
sin(rotation_rate*t)*sin(lat)*sin(lon)) + (sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lon)*sin(lat)) + cos(argp)*cos(lat)*sin(incl)) +
r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lon)*sin(lat)) + (cos(argp)*cos(raan) -
cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) -
sin(rotation_rate*t)*sin(lat)*sin(lon)) -
cos(lat)*sin(argp)*sin(incl)))*(r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan)
+ cos(argp)*cos(incl)*sin(raan))*(rotation_rate*cos(rotation_rate*t)*sin(lat)*sin(lon) +
rotation_rate*sin(rotation_rate*t)*cos(lon)*sin(lat)) - (sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))*(rotation_rate*cos(rotation_rate*t)*cos(lon)*sin(lat) -
rotation_rate*sin(rotation_rate*t)*sin(lat)*sin(lon))) +
r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))*(rotation_rate*cos(rotation_rate*t)*cos(lon)*sin(lat) -

rotation_rate*sin(rotation_rate*t)*sin(lat)*sin(lon)) - (cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan))*(rotation_rate*cos(rotation_rate*t)*sin(lat)*sin(lon) + rotation_rate*sin(rotation_rate*t)*cos(lon)*sin(lat))) - (mu^(1/2)*cos((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) - sin(rotation_rate*t)*sin(lat)*sin(lon)) + (sin(argp)*sin(raan) - cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) + sin(rotation_rate*t)*cos(lon)*sin(lat)) + cos(argp)*cos(lat)*sin(incl)))/r_2D^(1/2) - (mu^(1/2)*sin((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) + cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) + sin(rotation_rate*t)*cos(lon)*sin(lat)) + (cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) - sin(rotation_rate*t)*sin(lat)*sin(lon)) - cos(lat)*sin(argp)*sin(incl)))/r_2D^(1/2)) - 2*(Re*(cos(lon)*(sin(lat)^2 + cos(lat)^2*cos(lon)) - cos(lat)*sin(lon)^2) - r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) + cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) + sin(rotation_rate*t)*cos(lat)*cos(lon)) + (cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) - sin(rotation_rate*t)*cos(lat)*sin(lon)) + sin(argp)*sin(incl)*sin(lat)) + r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) - sin(rotation_rate*t)*cos(lat)*sin(lon)) + (sin(argp)*sin(raan) - cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) + sin(rotation_rate*t)*cos(lat)*cos(lon)) - cos(argp)*sin(incl)*sin(lat)))*(r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) + cos(incl)*sin(argp)*cos(raan))*(rotation_rate*cos(rotation_rate*t)*cos(lat)*cos(lon) - rotation_rate*sin(rotation_rate*t)*cos(lat)*sin(lon)) - (cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan))*(rotation_rate*cos(rotation_rate*t)*cos(lat)*sin(lon) + rotation_rate*sin(rotation_rate*t)*cos(lat)*cos(lon))) + r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) +

cos(argp)*cos(incl)*sin(raan))*(rotation_rate*cos(rotation_rate*t)*cos(lat)*sin(lon) +
rotation_rate*sin(rotation_rate*t)*cos(lat)*cos(lon)) - (sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))*(rotation_rate*cos(rotation_rate*t)*cos(lat)*cos(lon) -
rotation_rate*sin(rotation_rate*t)*cos(lat)*sin(lon))) -
(mu^(1/2)*sin((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lat)*cos(lon)) + (cos(argp)*cos(raan) -
cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) -
sin(rotation_rate*t)*cos(lat)*sin(lon)) + sin(argp)*sin(incl)*sin(lat)))/r_2D^(1/2) -
(mu^(1/2)*cos((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) +
cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) -
sin(rotation_rate*t)*cos(lat)*sin(lon)) + (sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lat)*cos(lon)) -
cos(argp)*sin(incl)*sin(lat)))/r_2D^(1/2))))/(log(10)*((Re*(cos(lon)*(sin(lat)^2 +
cos(lat)^2*cos(lon)) - cos(lat)*sin(lon)^2) -
r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lat)*cos(lon)) + (cos(argp)*cos(raan) -
cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) -
sin(rotation_rate*t)*cos(lat)*sin(lon)) + sin(argp)*sin(incl)*sin(lat)) +
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) +
cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lat)*cos(lon) -
sin(rotation_rate*t)*cos(lat)*sin(lon)) + (sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*cos(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lat)*cos(lon)) - cos(argp)*sin(incl)*sin(lat)))^2 +
(Re*(cos(lon)*sin(lon) + cos(lat)*cos(lon)*sin(lon)) -
r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) +
sin(rotation_rate*t)*cos(lon))*(cos(argp)*cos(raan) - cos(incl)*sin(argp)*sin(raan)) -
(cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(cos(argp)*sin(raan) +

```
cos(incl)*sin(argp)*cos(raan))) +
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((cos(rotation_rate*t)*sin(lon) +
sin(rotation_rate*t)*cos(lon))*(sin(argp)*cos(raan) + cos(argp)*cos(incl)*sin(raan)) -
(cos(rotation_rate*t)*cos(lon) - sin(rotation_rate*t)*sin(lon))*(sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))))^2 + (Re*(cos(lon)*(cos(lat)*sin(lat) -
cos(lat)*cos(lon)*sin(lat)) + sin(lat)*sin(lon)^2) -
r_2D*sin((mu^(1/2)*t)/r_2D^(3/2))*((sin(argp)*cos(raan) +
cos(argp)*cos(incl)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) -
sin(rotation_rate*t)*sin(lat)*sin(lon)) + (sin(argp)*sin(raan) -
cos(argp)*cos(incl)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lon)*sin(lat)) + cos(argp)*cos(lat)*sin(incl)) +
r_2D*cos((mu^(1/2)*t)/r_2D^(3/2))*((cos(argp)*sin(raan) +
cos(incl)*sin(argp)*cos(raan))*(cos(rotation_rate*t)*sin(lat)*sin(lon) +
sin(rotation_rate*t)*cos(lon)*sin(lat)) + (cos(argp)*cos(raan) -
cos(incl)*sin(argp)*sin(raan))*(cos(rotation_rate*t)*cos(lon)*sin(lat) -
sin(rotation_rate*t)*sin(lat)*sin(lon)) - cos(lat)*sin(argp)*sin(incl)))^2))
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C.

This appendix shows figures from the ten-satellite test case in Chapter IV. The snapshots progress from left to right, and highlight notable events in the solution. The red circle indicates current position of the antenna.

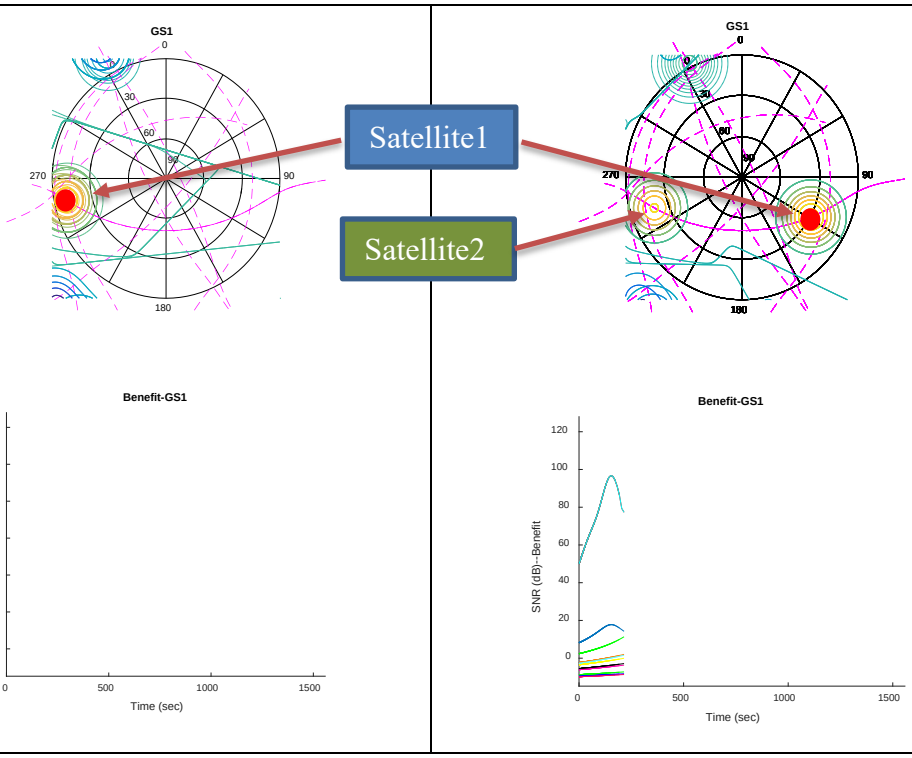
To recreate this scenario, utilize the following settings:

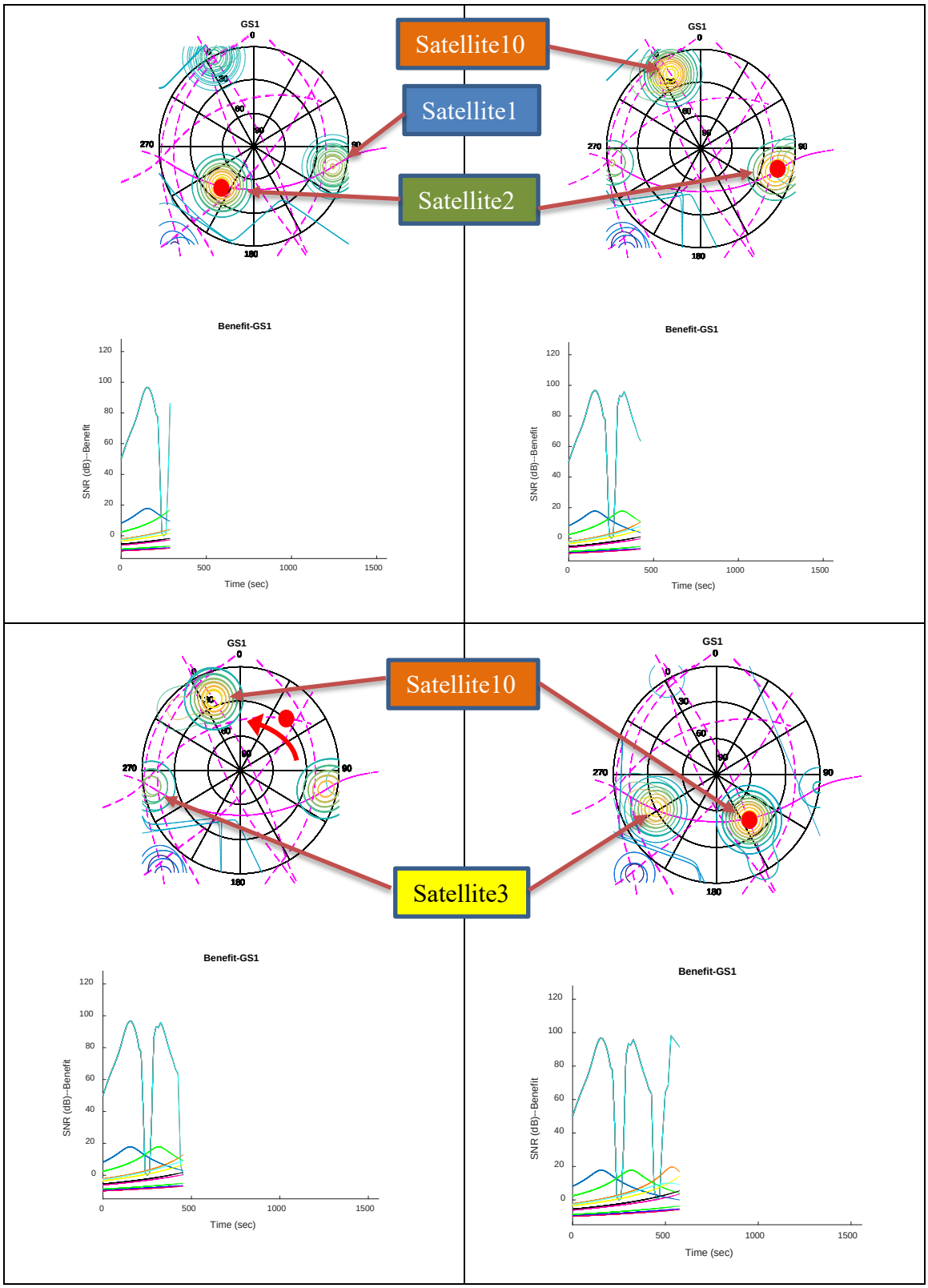
GS1 located at 2°N, 0°E, elevation 0m.

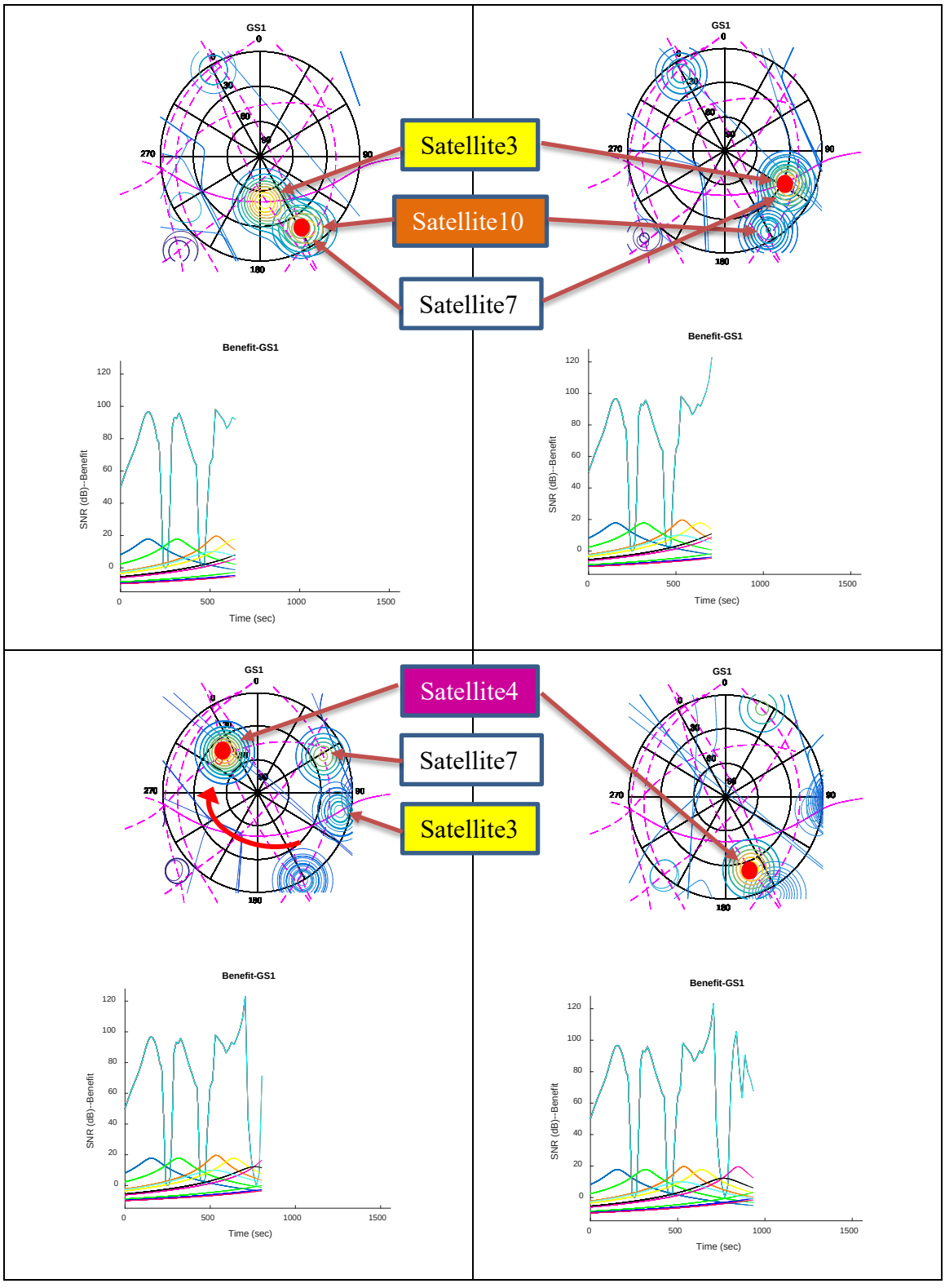
Start Time: 15 Jan 2018 02:27:02	Stop Time: 15 Jan 2018 02:53:00

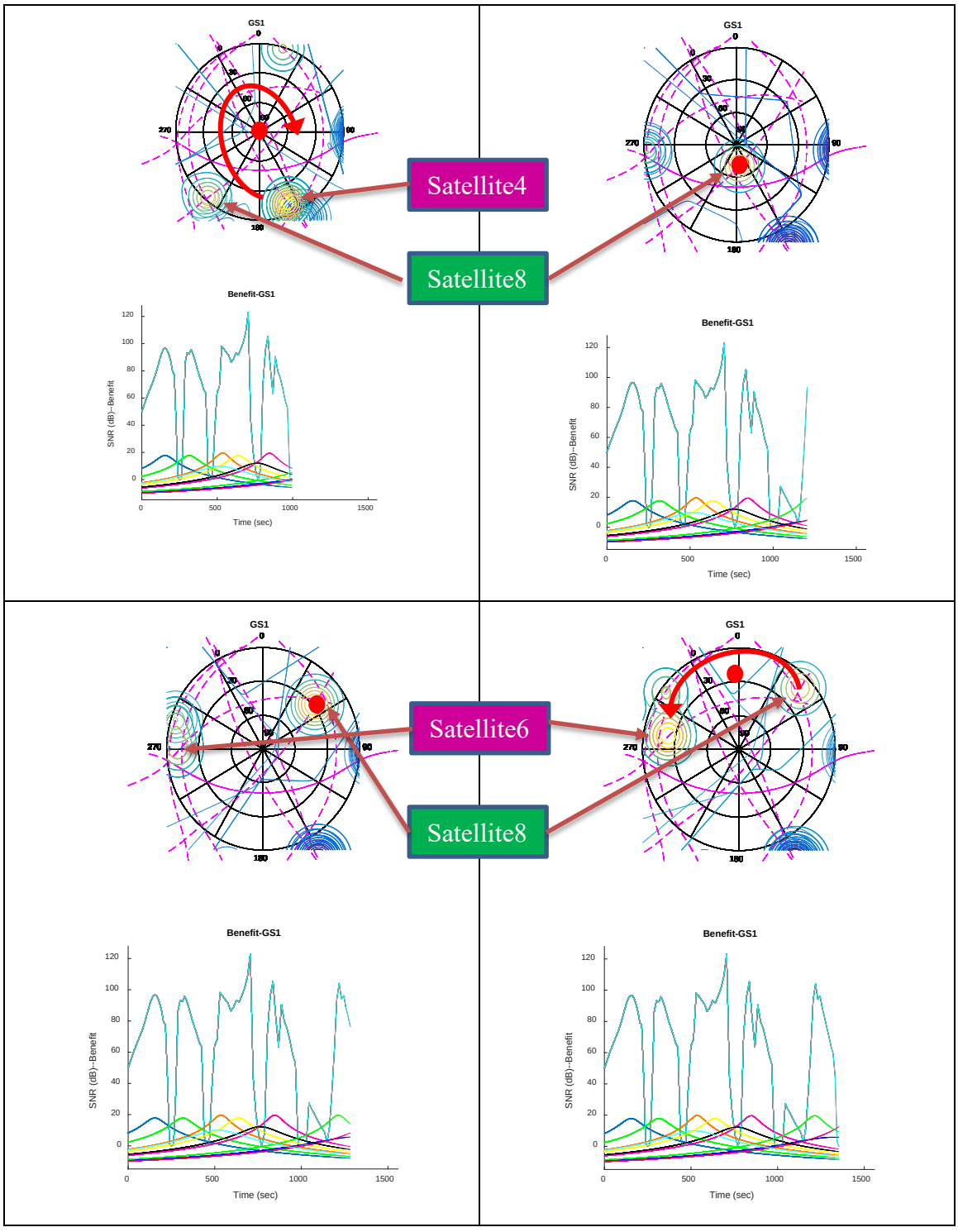
Satellite Name	**Semimajor Axis (km)**	**Eccentricity**	**Inclination (deg)**	**Argument of Perigee (deg)**	**RAAN (deg)**	**True Anomaly (deg)**
Satellite1	6678.14	0	0	0	0	0
Satellite2	6678.14	0	0	0	0	350
Satellite3	6678.14	0	0	0	0	330
Satellite4	6678.14	0	60	0	0	335
Satellite5	6678.14	0	60	0	140	160
Satellite6	6678.14	0	15	0	140	160
Satellite7	6678.14	0	90	0	160	190
Satellite8	6678.14	0	45	0	155	160
Satellite9	6678.14	0	98	0	325	20
Satellite10	6678.14	0	60	0	335	20

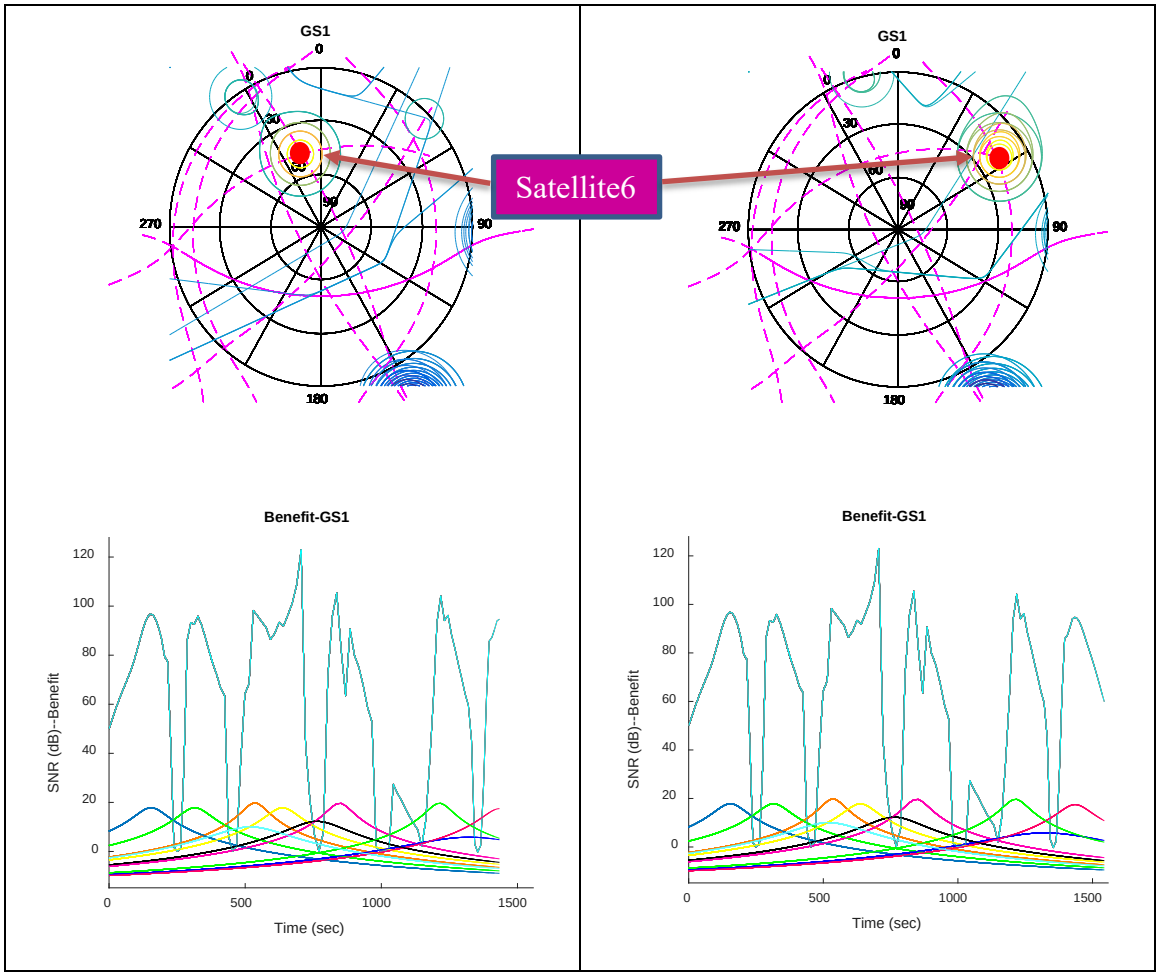

Satellite4
GS1
Satellite3
Satellite2
Satellite1
Satellite9
Satellite10
Satellite7
Satellite6
Satellite8
Satellite5


GS1
Satellite1
Satellite2
GS1
Benefit-GS1
SNR (dB)–Benefit
Time (sec)
Benefit-GS1
SNR (dB)–Benefit
Time (sec)


GS1
Satellite10
Satellite1
Satellite2
Benefit-GS1
SNR (dB)--Benefit
Time (sec)
GS1
Satellite10
Satellite3
Benefit-GS1
SNR (dB)--Benefit
Time (sec)


GS1
Satellite3
Satellite10
Satellite7
Benefit-GS1
SNR (dB)--Benefit
Time (sec)
GS1
Benefit-GS1
SNR (dB)--Benefit
Time (sec)
GS1
Satellite4
Satellite7
Satellite3
Benefit-GS1
SNR (dB)--Benefit
Time (sec)
GS1
Benefit-GS1
SNR (dB)--Benefit
Time (sec)


GS1
Satellite4
Satellite8
Benefit-GS1
SNR (dB)--Benefit
Time (sec)
GS1
Benefit-GS1
SNR (dB)--Benefit
Time (sec)
GS1
Satellite6
Satellite8
Benefit-GS1
SNR (dB)--Benefit
Time (sec)
GS1
Benefit-GS1
SNR (dB)--Benefit
Time (sec)


GS1
0
30
60
90
270
90
180
Satellite6
GS1
0
30
60
90
270
90
180
Benefit-GS1
SNR (dB)--Benefit
120
100
80
60
40
20
0
0
500
1000
1500
Time (sec)
Benefit-GS1
SNR (dB)--Benefit
120
100
80
60
40
20
0
0
500
1000
1500
Time (sec)

LIST OF REFERENCES

[1] Mission Design Staff, "Small spacecraft technology state of the art," NASA Ames Research Center, Tech. Rep. NASA/TP-2014-216648, 2014.

[2] T. Wilcox, "Office of space launch atlas V aft bulkhead carrier and operationally unique technologies satellite," in *AIAA/USU Conference on Small Satellites,* Logan, UT, Aug. 2012.

[3] G. Skrobot and R. Coelho, "ELaNa educational launch of nanosatellite providing routine RideShare opportunities," Proceedings of the *AIAA/USU Conference on Small Satellites,* SSC12-V-5, Logan, UT, Aug. 2012.

[4] J. Robinson, "Small satellite rideshares on commercial resupply missions to the international space station," in *AIAA/USU Conference on Small Satellites,* Logan, UT, Aug. 2012.

[5] SpaceWorks Commercial. "SpaceWorks Nano/Microsatellite Market Forecast, 8th Edition." [Online]. Available: http://www.spaceworkscommercial.com/.

[6] J. Straub, "In search of standards for the operation of small satellites," in *53rd AIAA Aerospace Sciences Meeting*, 2015.

[7] J. R. Wertz, "Creating a resilient space mission architecture," in *AIAA Space 2016*, Long Beach, CA, Sep. 2016.

[8] J. Cutler and G. Hutchins, "OPAL: Smaller, simpler, and just plain luckier," in *AIAA/USU Conference on Small Satellites,* Logan, UT, Aug. 2000.

[9] D. Sakoda *et al.,* "Naval postgraduate school PANSAT: Lessons learned," *Space 2001 Conference & Exposition*, Albuquerque, New Mexico, Aug. 2001.

[10] The Cubesat Program, "CubeSat design specification," California Polytechnic State University, Tech. Rep. CDS REV 13, Feb. 20, 2014.

[11] C. G. Turner, "NPS TINYSCOPE Program Management." , Master's Thesis, Naval Postgraduate School, Monterey, CA, 2010.

[12] C. Fossa *et al.,* "An overview of the iridium low earth orbit (LEO) satellite system," *Proceedings of the IEEE 1998 National Aerospace and Electronics Conference. NAECON 1998*, Dayton, OH, July 17, 1998.

[13] C. Boshuizen *et al.,* "Results from the planet labs flock constellation," in *AIAA/USU Conference on Small Satellites,* Logan, UT, Aug. 2014.

[14] F. Morring Jr, “Spaceflight Industries spinoff promises ‘hourly’ Earth images,” *Aerospace Daily & Defense Report,* vol. 252, (56)*,* Jun. 2015.

[15] E. Hand, “CubeSats promise to fill weather data gap,” *Science,* vol. 350, *(6266),* pp. 1302–1303, Dec. 11, 2015.

[16] P. B. de Selding, “One year after kickoff, OneWeb says its 700-satellite constellation is on schedule,” *Space News,* Jul. 6, 2016.

[17] A. Svitak, “Two’s Company,” *Aviation Week Space Technology,* vol. 177, *(2),* pp. 32, Feb. 2015.

[18] M. Sorgenfrei and M. Nehrenz, “Operational considerations for a swarm of CubeSat-class spacecraft,” in *SpaceOps 2014 Conference*, Pasadena, CA, May 2014.

[19] R. Rose *et al.,* “CYGNSS mission overview and preflight update,” in *AIAA/USU Conference on Small Satellites,* Logan, UT, Aug. 2016.

[20] S. Bandyopadhyay *et al.,* “A Review of Impending Small Satellite Formation Flying Missions,” *53rd AIAA Aerospace Sciences Meeting*, 2015.

[21] National Science Foundation, “CubeSat-based science missions for geospace and atmospheric research,” Oct. 2013.

[22] C. Galliand, “Study of the small: Potential for operational military use of CubeSats,” in *AIAA/USU Conference on Small Satellites,* Logan, UT, Aug. 2010.

[23] Homeland Defense and Global Security, “Space domain mission assurance: A resilience taxonomy,” Sep. 2015.

[24] S. C. Spangelo, D. Boone and J. Cutler, “Assessing the capacity of a federated ground station,” in *2010 IEEE Aerospace Conference,* 2010, pp. 1–9.

[25] E. Smith and D. Korsmeyer, “Intelligent systems technologies for ops,” in the *12th International Conference on Space Operations,* Stockholm, Sweden, 2012.

[26] G. Minelli, “Model-Based Reasoning Applied to Biological Spacecraft Payloads for Anomaly Management,” Master’s Thesis, Santa Clara University, Santa Clara, CA, 2011.

[27] C. Leonard and A. Mims, “AFSCN and other DoD ground support system,” in *Space Programs and Technologies Conference and Exhibit,* Huntsville, AL, 1993.

[28] S. Schaire *et al.,* “NASA near earth network (NEN) and space network (SN) CubeSat communications,” in *SpaceOps 2016 Conference,* Daejon, Korea, 2016.

[29] Swedish Space Corporation. “About SSC.” [Online]. Available: http://www.sscspace.com/about-the-ssc-group/ssc-companies/sscspaceus.

[30] B. Eilertsen, M. Krynitz and K. Olafsson, “NewSpace - forcing a rethink of ground networks,” in *SpaceOps 2016 Conference,* Daejeon, Korea, 2016.

[31] ATLAS Space Operations, “ATLAS Capabilities Statement,” 2016.

[32] ATLAS Space Operations, “ATLAS Freedom Software Platform,” 2016.

[33] National Aeronautics and Space Administration. “Near Earth Network (NEN).” [Online]. Available: https://www.nasa.gov/directorates/heo/scan/services/networks/nen.

[34] J. Cutler, P. Linder and A. Fox, “A federated ground station network,” in *SpaceOps Conference*, Oct. 2002.

[35] J. W. Cutler and C. A. Kitts, “Mercury - A satellite ground station control system,” in *1999 IEEE Aerospace Conference*, Aspen, CO, Mar. 1999.

[36] G. Shirville and B. Klofas, “GENSO: A global ground station network,” in *Amsat Symposium,* 2007.

[37] K. Leveque, J. Puig-Suari and C. Turner, “Global educational network for satellite operations (GENSO),” in *AIAA/USU Conference on Small Satellites,* Logan, UT, Aug. 2007.

[38] S. M. Alcaide, “Mobile Cubesat Command and Control (MC3) 3-Meter Dish Calibration and Capabilities,” Master’s Thesis, Naval Postgraduate School, Monterey, CA, 2014.

[39] A. J. Felt, “Federated Ground Station Network Model and Interface Specification,” Master’s Thesis, Naval Postgraduate School, Monterey, CA, 2014.

[40] P. Ibbitson, “Mobile Cubesat Command and Control (MC3) Architecture and CONOPS,” Master’s Thesis, Naval Postgraduate School, Monterey, CA, 2012.

[41] K. Pohl, “Ground System Development – Enhancing AFIT Capabilities and Joining the MC3 Network,” Master’s Thesis, Air Force Institute of Technology, Dayton, OH, 2013.

[42] R. C. Griffith, “Mobile CubeSat Command and Control (MC3),” Master’s Thesis, Naval Postgraduate School, Monterey, CA, 2011.

[43] G. C. Morrison, “Mobile CubeSat Command and Control: Assembly and Lessons Learned,” Master’s Thesis, Naval Postgraduate School, Monterey, CA, 2011.

[44] A. Writt, "Optimization of CubeSat Ground Station for Improved Satellite Numbers," Master's Thesis, Naval Postgraduate School, Monterey, CA, 2018.

[45] A. D. Santangelo, "The LinkStar system, a Globalstar based radio and flight system for satellites in LEO - architecture and test results," in *55th AIAA Aerospace Sciences Meeting*, Grapevine, TX, Jan. 2017.

[46] M. Rice, "Space-based networking using CubeSat technologies," in *33rd AIAA International Communications Satellite Systems Conference and Exhibition*, Queensland, Australia, Sep. 2015.

[47] G. Minelli, M. Karpenko, and I. M. Ross, "Autonomous operations of large-scale satellite constellations and ground station networks," in *AAS/AIAA Astrodynamics Specialist Conference,* Stevenson, WA, Aug. 2017.

[48] I. M. Ross, M. Karpenko and R. Proulx, "A Nonsmooth Calculus for Solving Some Graph-Theoretic Control Problems," *International Federation of Automatic Control,* vol. 49, *(18),* Aug. 2016, pp. 462–467.

[49] J. Greenslade, "An Optimal Control Approach to Mission Planning Problems: A Proof of Concept," Master's Thesis, Naval Postgraduate School, Monterey, CA, 2014.

[50] S. Spangelo, "Modeling and Optimizing Space Networks for Improved Communication Capacity," PhD Dissertation, University of Michigan, Ann Arbor, MI, 2013.

[51] L. Barbulescu *et al.,* "Scheduling Space-Ground Communications for the Air Force Satellite Control Network," *Journal of Scheduling,* vol. 7, *(1),* Jan. 2004, pp. 7–34.

[52] S. Damiani *et al.,* "Automated Allocation of ESA Ground Station Network Services," *International Workshop on Planning and Scheduling for Space,* Oct. 22–25, 2006.

[53] J. Sun and F. Xhafa, "A genetic algorithm for ground station scheduling," in *International Conference on Complex, Intelligent and Software Intensive Systems (CISIS),* Seoul, Korea, 2011.

[54] I. M. Ross *et al.,* "Dynamic Optimization for Satellite Image Collection," *26th AAS/AISS Space Flight Mechanics Meeting,* Feb.14-18, 2016.

[55] P. Bernhardt, "The Largest Ionospheric Disturbances Produced by the HAARP HF Facility." [Online]. Available: http://ies2015.bc.edu/wp-content/uploads/2015/05/040-Bernhardt-Slides.pdf.

[56] M. B. Wall, “A Genetic Algorithm for Resource-Constrained Scheduling,” PhD Dissertation, Massachusetts Institute of Technology, Boston, MA, 1996.

[57] G. Wu, M. Ma, J. Zhu, and D. Qiu, “Multi-satellite observation integrated scheduling method oriented to emergency tasks and common tasks,” *Journal of Systems Engineering and Electronics,* vol. 23, *(5),* Nov. 2012, pp. 723–733.

[58] F. Marinelli, S. Nocella, F. Rossi, and S. Smriglio, “A Lagrangian heuristic for satellite range scheduling with resource constraints,” *Computers and Operations Research,* vol. 38, *(11)*, Nov. 2011, pp. 1572–1583.

[59] S. Spangelo and J. Cutler, “Optimization of single-satellite operational schedules towards enhanced communication capacity,” in *AIAA Guidance, Navigation, and Control Conference*, Minneapolis, MN, Aug. 2012.

[60] R. Dempsey, “The International Space Station, Operating an Outpost in the New Frontier,” National Aeronautics and Space Administration, 2018.

[61] A. Rich, “Flight planning and procedures,” University of Texas at Austin, Austin TX, May, 2016.

[62] J. Frank, P. Morris and J. Greene, “The challenge of evolving mission operations tools for manned space flight,” in *Proceedings of the 9th International Symposium on Artificial Intelligence, Robotics, Automation in Space*, 2008.

[63] E. E. Smith, D. J. Korsmeyer and V. Hall, “Exploration technologies for operations,” in *SpaceOps 2014*, Pasadena, CA, May, 2014.

[64] L. Barbulescu, L. D. Whitley and A. E. Howe, “Leap before you look: An effective strategy in an oversubscribed scheduling problem,” in *Proceedings of the 19th National Conference on Artificial Intelligence,* San Jose, CA, 2004, pp. 143–148.

[65] P. Soma, S. Venkateswarlu, S. Santhalakshmi, T. Bagchi, and S. Kumar, “Multi-satellite scheduling using genetic algorithms,” in *SpaceOps 2004 Conference*, Montreal, Quebec, Canada, May, 2004.

[66] W. Martin, “Satellite Image Collection Optimization,” *Optical Engineering,* vol. 41, *(9)*, Sep. 2002, pp. 2083–2087.

[67] J. C. Pemberton and I. Galiber Flavius, “A constraint-based approach to satellite scheduling,” in *DIMACS Workshop on Constraint Programming and Large Scale Discrete Optimization,* Rutgers University, Piscataway, NJ, 2001, pp. 101–114.

[68] S. Spangelo, J. Cutler, K. Gilson, and A. Cohn, "Optimization-based Scheduling for the Single-satellite, Multi-ground Station Communication Problem," *Computers and Operations Research,* vol. 57, *(C),* May, 2015, pp. 1–16.

[69] J. King, "A New Approach for Optimizing Image Collection from Space," PhD Dissertation, Naval Postgraduate School, Monterey, CA, 2014.

[70] P. Palmer, *The Complete Orienteering Manual.* Ramsbury, Wiltshire, England: Crowood Press, 1997.

[71] D. Feillet, P. Dejax and M. Gendreau, "Traveling Salesman Problem with Profits," vol. 39, *(2),* May, 2005, pp. 188–205.

[72] P. Vansteenwgen, W. Souffriau and D. Van Oudheusden, "The orienteering problem: A survey," *European Journal of Operational Research,* vol. 209, *(1)*, Feb. 2011, pp. 1–10.

[73] I. M. Ross, *A Primer on Pontryagin's Principle in Optimal Control.* (2nd ed.), Collegiate Publishers, Carmel, CA, 2015.

[74] J. Longuski, J. Guzman and J. Prussing, *Optimal Control with Aerospace Applications.* (1st ed.), Microcosm Press, El Segundo, CA, 2014.

[75] F. Fahroo and I. M. Ross, "Advances in pseudospectral methods for optimal control," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, HI, 2008.

[76] I. M. Ross and M. Karpenko, "A review of pseudospectral optimal control: From theory to flight," *Annual Reviews in Control,* vol. 36, *(2),* Dec. 2012, pp. 182–197.

[77] I. M. Ross, C. D'Souza, F. Fahroo, and J.B. Ross, "A fast approach to multi-stage launch vehicle trajectory optimization," in *AIAA Guidance, Navigation and Control Conference,* Austin, TX, 2003.

[78] N. Bedrossian, S. Bhatt, M. Lammers, L. Nguyen, and Y. Zhang, "First ever flight demonstration of zero propellant maneuver attitude control concept," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, Hilton Head, SC, 2007.

[79] M. Karpenko, S. Bhatt, N. Bedrossian, A. Fleming, and I. M. Ross, "First Flight Results on Time-Optimal Spacecraft Slews," *Journal of Guidance, Control, and Dynamics,* vol. 35, *(2)*, Mar. 2012, pp. 367–376.

[80] M. A. Hurni, “An Information-Centric Approach to Autonomous Trajectory Planning Utilizing Optimal Control Techniques,” PhD Dissertation, Naval Postgraduate School, Monterey, CA, 2009.

[81] D. Vallado, “Kepler’s equation and kepler’s problem,” in *Fundamentals of Astrodynamics and Applications*, Third ed., Microcosm Press, Hawthorne, CA, 2007, pp. 49–103.

[82] H. Curtis, “Orbital position as a function of time,” in *Orbital Mechanics for Engineering Students*, First ed.Anonymous Elsevier Butterworth-Heinemann, 2005, pp. 107–144.

[83] H. Curtis, “Orbital elements and the state vector,” in *Orbital Mechanics for Engineering Students*, First ed., Elsevier Butterworth-Heinemann, 2005, pp. 158–187.

[84] H. Krag, K. Merz, T. Flohrer, S. Lemmens, B. Bastida Virgili, Q. Funke, and V. Braun, “ESA’s modernised collision avoidance service,” in *SpaceOps 2016 Conference*, Daejeon, Korea, 2016.

[85] T. S. Kelso, “Frequently Asked Questions: Two-Line Element Set Format,” *Satellite Times,* vol. 4, *(3),* March, 1998, pp. 52–54.

[86] D. Vallado and P. Crawford, “SGP4 orbit determination,” in *AIAA/AAS Astrodynamics Specialist Conference*, Honolulu, HI, Aug. 2008.

[87] B. Coffee, R. Bishop and K. Cahoy, “Propagation of CubeSats in LEO using NORAD two line element sets: Accuracy and update frequency,” in *AIAA Guidance, Navigation, and Control Conference*, Boston, MA, Aug. 2013.

[88] I. Mas and C. Kitts, “A flight-proven 2.4 GHz ISM band COTS communications system for small satellites,” in *AIAA/USU Conference on Small Satellites,* Logan, UT, Aug. 2007.

[89] G. Minelli *et al.,* “Extended life flight results from the GeneSat-1 biological microsatellite mission,” in *AIAA/USU Conference on Small Satellites,* Logan, UT, Aug. 2008.

[90] J. R. Wertz, D. F. Everett and J. J. Puschell, “Communications payloads,” in *Space Mission Engineering: The New SMAD*, 4th ed., Microcosm Press, Hawthorne, CA, 2011, pp. 455–492.

[91] G. Gordon and W. Morgan, *Principles of Communications Satellites,* John Wiley and Sons, 1993.

[92] G. Gordon and W. Morgan, "System performance," in *Principles of Communications Satellites,* John Wiley and Sons, 1993, pp. 234–265.

[93] Over-the-Air Digital TV. "Television Antennas." [Online]. Available: http://otadtv.com/antenna/index.html.

[94] Elissar Global, "Scaling and balancing optimal control problems: A must-read document for any serious user of DIDO," Elissar Global, Carmel, CA, 2016.

[95] I. M. Ross, Q. Gong, M. Karpenko, R. J. Proulx, "Scaling and Balancing for High-Performance Computation of Optimal Controls," *Journal of Guidance, Control, and Dynamics*, Aug. 2018.

[96] O. Yakimenko, "Numerical integration," in *Engineering Computations and Modeling in MATLAB/Simulink*, American Institute of Aeronautics and Astronautics, Reston, VA., 2011, pp. 583–679.

[97] J. Kopitzki, "Development and Implementation of a Communication Scheme for Software Defined Radios," Master's Thesis, Naval Postgraduate School, Monterey, CA, 2014.

[98] Joint Space Operations Center. "Satellite Position Catalog." [Online]. Available: www.space-track.org.

[99] A. Malke, "Analysis and Implementation of a Software-Defined Radio Uplink to CubeSats in LEO," Master's Thesis, Naval Postgraduate School, Monterey, CA, 2017.

[100] GNU Radio Project. "The Free and Open Software Radio Ecosystem." [Online]. Available: https://www.gnuradio.org/.

[101] J. M. Roehrig, "Development of a Versatile Groundstation Utilizing Software Defined Radio," Master's Thesis, Naval Postgraduate School, Monterey, CA, 2016.

[102] P. Lemke, "Carrier Frequency Offset Estimation with CPM CubeSat Signals at High Doppler Shifts," Master's Thesis, Naval Postgraduate School, Monterey, CA, 2017.

[103] J. Leone, "CubeSat Pass Quality Analysis and Predictive Model" Master's Thesis, Naval Postgraduate School, Monterey, CA, 2018.

[104] G. Minelli, M. Karpenko and I. M. Ross, "Systems and Methods for Autonomous Operations of Ground Station Networks," 15/827,832 (pending), July 5, 2017.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California