

**Use of Isoperformance, Constraint Programming, and Mixed Integer Linear Programming
for Architecture Tradespace Exploration of Passive Optical Earth Observation Systems**

By

Jeffrey Louis Menezes

B.S. Mechanical Engineering, University of California, Berkeley 2012

Submitted to the MIT Sloan School of Management and the Department of Aeronautics and
Astronautics in partial fulfillment of the requirements for the degrees of

Master of Business Administration
and
Master of Science in Aeronautics and Astronautics

In conjunction with the Leaders for Global Operations Program at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 2018

© Jeffrey Menezes, MMXVIII. All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic
copies of this thesis document in whole or in part in any medium now known or hereafter created.

Signature redacted

Author _____

MIT Sloan School of Management and the Department of Aeronautics and Astronautics

May 7, 2018

Signature redacted

Certified by _____

Kerri Cahoy, Thesis Supervisor

Associate Professor of Aeronautics and Astronautics

Signature redacted

Certified by _____

Roy Welsch, Thesis Supervisor

Professor of Statistics and Data Science, MIT Sloan School of Management

Signature redacted

Approved by _____

Hamsa Balakrishnan

Associate Professor of Aeronautics and Astronautics

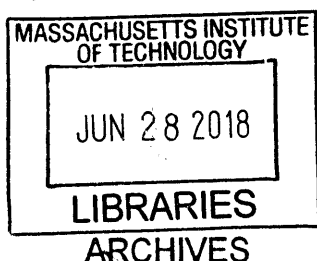
Chair, Graduate Program Committee

Signature redacted

Approved by _____

Maura Herson

Director, MBA Program, MIT Sloan School of Management



THIS PAGE IS INTENTIONALLY LEFT BLANK

Use of Isoperformance, Constraint Programming, and Mixed Integer Linear Programming for Architecture Tradespace Exploration of Passive Optical Earth Observation Systems

by

Jeffrey Louis Menezes

Submitted to the MIT Sloan School of Management and the Department of Aeronautics and Astronautics on May 22, 2018, in partial fulfillment of the requirements for the degrees of

Master of Business Administration
and
Master of Science in Aeronautics and Astronautics

Abstract

This thesis presents work performed during the course of an internship at An Aerospace Company (AAC) and research performed at Massachusetts Institute of Technology (MIT) Lincoln Laboratory as part of a fellowship. Both efforts entailed the development of architecture tradespace exploration models for space systems.

The tradespace exploration model developed at AAC, called the Earth Observation Architecture Isoperformance Model (EO-AIM), uses automation techniques, isoperformance, and constraint programming to rapidly construct potential space-based passive optical EO sensor architecture concepts which meet a given set of customer requirements. Cost estimates are also generated for each sensor concept via integration with stakeholder-trusted cost modeling software allowing for cost to be treated as both an independent variable and consequence when evaluating various architecture solutions. The EO-AIM then uses simple algorithms to identify potential satellite bus options for hosting each sensor architecture in orbit. The total cost of populating an entire constellation based on the sensor architecture is finally estimated using cost estimates for the sensor, satellite bus, and the best launch vehicle option capable of lifting the satellite(s) to orbit. In general, the EO-AIM seeks to bolster AAC's capabilities for conducting architecture trade space exploration and initial proposal development given advancements in satellite bus, launch vehicle, and sensing technologies.

The tradespace exploration model developed at MIT Lincoln Laboratory is a satellite network mixed integer linear program (MILP) which is used for making system architecture decisions and estimating final architecture cost. The satellite network MILP is formulated as both an assignment problem and a network maximum flow problem which must send sensor generated data to a ground user. Results of the MILP vary with the selected objective function and provide insights on the potential benefits of architecture decisions such as sensor disaggregation and the utility of introducing additional communication nodes into existing networks. The satellite network MILP is also capable of verifying network data volume throughput capacity and providing an optimized link schedule for the duration of the simulation. Overall, the satellite network MILP model explores the general problem of optimizing use of limited resources for a given space-based sensor while ensuring mission data needs are met. It is a higher fidelity alternative to the simple satellite bus and launch vehicle compatibility algorithm used in EO-AIM.

Both models are shown to improve architecture tradespace exploration of space-based passive-optical EO systems. With a simple demonstration, it is exhibited that using the EO-AIM can increase sensor architecture concepts generated by a factor of ten or more by creating all feasible sensor architecture concepts given user inputs and settings. Furthermore, the use of the satellite network MILP to examine alternative network architecture options for NASA's HypIRI mission resulted in a system architecture with 20% higher data throughput for marginally less cost.

Thesis Supervisor: Kerri Cahoy
Associate Professor of Aeronautics and Astronautics

Thesis Supervisor: Roy Welsch
Professor of Statistics and Data Science, MIT Sloan School of Management

THIS PAGE IS INTENTIONALLY LEFT BLANK

Acknowledgements

The work presented in this thesis was only possible because of the support, mentorship, and dedication I received from multiple people throughout my journey at MIT.

First, I'd like to thank my thesis supervisors, Professor Roy Welsch and Professor Kerri Cahoy, for their guidance and advice while writing this thesis. Their continuous support while I was away from campus was incredibly helpful in keeping me on track and focused.

Second, I'd like to thank MIT Lincoln Laboratory. In particular, I could not have attended MIT without the help of Mr. John Kuconis, to whom I owe much. While performing research at MIT Lincoln Laboratory, I also received amazing coaching from Dr. Thomas Roy. Dr. Roy continuously took time from his busy schedule to discuss where my project was going and give meaningful advice on where to go next. Also having a formative role on my Lincoln Laboratory research were fellow Course Sixteen students Emily Clements and Andrew Kennedy.

Of course, this thesis wouldn't be possible without all of the outstanding people at An Aerospace Company (AAC) whom I interacted with on a daily basis. A big thanks to the amazing engineers and LGO alumni who took time to humor my ideas, teach, give feedback, and work with me. The work presented in my thesis is largely thanks to their help.

The biggest thanks for the work presented in my thesis goes to my AAC project supervisor, Amber P. Amber is one of the best technical leaders I've met and served as an amazing coach that gave me both room to explore and guidance when I needed it. My internship was a success because of her.

Finally, I'd like to thank my fellow LGO classmates and the LGO program office. To my classmates, thanks for all of the good times! We made it through as a team. To the program office, thanks for this amazing opportunity. It's been fun.

THIS PAGE IS INTENTIONALLY LEFT BLANK

The author wishes to acknowledge the Leaders for
Global Operations Program for its support of this work.

THIS PAGE IS INTENTIONALLY LEFT BLANK

Contents

Contents.....	10
List of Figures.....	15
List of Tables.....	17
1 Introduction.....	19
1.1 Space-Based Passive Optical Earth Observation Sensor Architecture Tradespace Exploration at AAC.....	19
1.2 Use of a Satellite Network Mixed Integer Linear Program for System Architecture Tradespace Exploration.....	20
2 Background.....	21
2.1 Earth Observation Missions and Passive Optical Sensors.....	21
2.2 Developments in Architecture Options.....	24
2.2.1 Developments in Satellite Bus Technology	24
2.2.2 Developments in Launch Vehicle Options.....	24
2.2.3 Developments in Sensor Technologies.....	25
2.3 New Entrants to the Earth Observation Domain.....	25
3 Sensor Architecture Tradespace Exploration Approach and Method.....	26
3.1 Literature Review	27
3.2 Interviews and Process Analysis.....	27
3.2.1 Description of Tools Used at AAC.....	27
3.3 Review of Findings.....	28
3.4 Solution to Findings.....	30
3.4.1 Isoperformance	30
3.4.2 Identification of Key Stakeholder Requirements for Passive Optical EO Systems	31
3.4.3 Constraint Satisfaction Problems.....	32
3.5 Modeling Passive Optical Earth Observations as a Constraint Satisfaction Problem...33	
3.5.1 Sensor Architecture Decision Variables	35
3.5.2 Critical Technology Subsystem Variables and Components	35
3.5.3 Enabling Subsystems Variables.....	36
3.6 Chapter Summary	36
4 Earth Observation Architecture Isoperformance Model (EO-AIM) Implementation:.....	36
5 Validation of the EO-AIM.....	40
5.1 Validating the Sensitivity Models	40

5.2	Validating Component Parametric Equations.....	40
5.3	Validating Sensor Cost Modeling	41
6	Model Use, Results, and Discussion	41
6.1	Mission Requirements	41
6.2	Test Case Focal Planes	41
6.3	Test Case Optical Form Options	42
6.4	Test Case Thermal Regulation Options.....	43
6.5	Test Case Calibrator Options	44
6.6	Test Case Electronics Module Options.....	44
6.7	Results and Discussion	45
6.7.1	No Disaggregation Whiskbroom Solution.....	45
6.7.2	Disaggregated Pushbroom Results	46
7	Summary of the Earth Observation Architecture Isoperformance Model (EO-AIM) and Discussion of Potential Benefits.....	47
7.1	Improved Brainstorming.....	48
7.2	Total Mission Insight.....	48
7.3	Competitive Analysis and Strategic Decision Making.....	49
7.4	Codification of Corporate Knowledge.....	49
7.5	Rapid Assessment of Architectures.....	49
7.6	Improved Communication	50
8	Future Work.....	50
9	Using a Satellite Network Mixed Integer Linear Program for System Architecture Decision Analysis.....	51
9.1	The Need for a Network Model for Evaluating Architectural Decisions.....	51
9.2	Satellite Network Mixed Integer Linear Program.....	53
9.2.1	Satellite Network Model Input Files.....	54
9.3	Reading File Inputs and Creating Potential Child-Object/Location Node Assignments 57	
9.3.1	HyspIRI Example Background.....	58
9.3.2	Location Nodes.....	59
9.3.3	HyspIRI Mission Sensors.....	60
9.3.4	Satellite Busses	60
9.3.5	Ground Station.....	61
9.3.6	Communications Technology.....	61
9.3.7	Constraints and Representation of the Assignment Data Frame	62

9.4	Creating Accesses Between Network Nodes	63
9.4.1	Calculating Time-Varying Accesses.....	63
9.4.2	Time-Invariant Accesses	63
9.4.3	Combining Accesses	64
9.4.4	HyspIRI Example: Accesses	64
9.4.5	Creating a Network Problem via Network Sub-Problems	65
9.5	Generalizing the Network Problem Notation and Solving It.....	67
9.6	Use and Results	68
9.6.1	Simulation Period	68
9.6.2	Mission Sensor Orbit Inputs.....	68
9.6.3	Sensors	68
9.6.4	Ground Location.....	68
9.6.5	Satellite Busses	69
9.6.6	Ground Station.....	69
9.6.7	Communication Technologies.....	69
9.6.8	Scenario Analysis	70
9.7	Conclusion and Future Work	74
Appendix A: NASA Systems Engineering Figures.....		76
Appendix B: Satellite Busses and Launch Vehicles.....		78
1	Selected Small Satellite Bus Information.....	78
2	Emerging Commercial Small Satellite Launch Providers.....	79
Appendix C: Architecture Variables and Model Data Structure Descriptions.....		80
1	The Orbit.....	80
2	Focal Plane Disaggregation	81
3	Sensor Scan Method.....	82
4	The Focal Plane Assembly	83
5	Wavelengths.....	85
6	Optical Form	86
7	Thermal Regulation Subsystem	88
8	Electronics Subsystem.....	88
9	Calibration Subsystem.....	88
10	Sensor Data Structure Description	88
Appendix D: Useful Equations.....		90
1	Helpful Circular Orbit Equations.....	90

2	Calculating the Number of Satellite Planes to meet Global Coverage Requirements and Revisit Rate Requirements	90
3	Common Geometry Relationships for All Sensor Scan Methods.....	92
4	Pushbroom Scan Method Constraints	93
5	Photocarrier Sensitivity Model	96
6	Microbolometer Sensitivity Model	99
	Appendix E: Model Operations and Constraints	100
1	Reading Input Data and Creating Model Data Structures	100
2	Assignment of the Orbit and Creation of Orbit Data Structures.....	100
3	Assigning Disaggregation Scheme, Scan Method, and Optical Forms to Create Initial Sensor Data Structures	101
4	Assigning the Pushbroom Scan Method - Creating LEO Pushbroom Imagers.....	101
5	Pushbroom Sensitivity Performance Algorithm for Assigning Component Variables.....	102
6	Whiskbroom Sensitivity Performance Algorithm for Assigning Component Variables....	104
7	Creating an Initial Cost Estimate	105
8	Assigning Enabling Subsystems	106
8.1	Focal Plane Thermal Regulation Subsystem.....	107
8.2	Calibration Subsystem	108
8.3	Electronics Subsystem	108
9	Final Sensor Characteristics.....	109
10	Final SEER Cost Estimate of the Sensor.....	109
11	Assigning Compatible Satellite Busses and Launch Vehicles to Create Architectures ..	110
11.1	Combining Sensors and Compatible Busses to Form Satellites	110
11.2	Assessing Launch Vehicle Options	113
12	Constructing Mission Architecture Solutions.....	113
	Appendix F: VIIRS Requirements.....	114
	Appendix G: Architecture Concept Creation Model Use Case	115
1	User Inputs.....	115
2	Test Case Whiskbroom Sensor Information.....	117
3	Test Case Pushbroom Sensors	121
4	SEER Command Template Example	125
	Appendix H: HypsIRI Example for Satellite Network Mixed Integer Linear Program	131
1	HypsIRI Example Decoder.....	131
2	Network Model JSON Example	132
3	HypsIRI Example Accesses.....	133

Appendix I: Satellite Network Mixed Integer Linear Program General Form	134
1 Sets	134
2 Decision Variables	135
3 Data Flow Constraints:	136
4 Flow Balance Constraints:	136
5 Assignment Constraints:	137
6 Dependencies Constraints:	137
7 Fixed Architecture Constraints:	137
8 Power Constraints:	137
9 Mass Constraints:	138
10 Sensor Assignment Constraints:	138
11 Max Bus Constraints:	138
12 Objective Functions:	138
13 Maximizing Data to Ground -Minimizing the Maximum Data Stored on Busses:	138
14 Minimizing Total Architecture Cost:	139
Appendix J: Satellite Network MILP Use Case Example.....	140
1 Base Scenario – Max Data to Ground Objective – Link Schedule	140
2 Base Scenario – Minimizing Costs Solver Results	141
3 Network MILP Scenario – Added Input Options	142
4 Network MILP Scenario – Added Input Options, Max Data to Ground Objective.....	143
5 Network MILP Scenario – Added Input Options, Minimize Total Cost	145
6 Published HypsIRI Architecture Results	146
Bibliography	147

List of Figures

Figure 1: An illustrative example of a space-based EO system: the JPSS architecture. Source: JPSS Program Overview, 17 November 2010.....	22
Figure 2: VIIRS block diagram and computer aided design model [4].	23
Figure 3: Constraint rules for what telescopes can be used given a scan method.....	33
Figure 4: Constraint propagation eliminating "Telescope 2" from optical form/telescope variable domain.....	33
Figure 5: Branching to search for at least two different solutions using different optical forms.	33
Figure 6: Block diagram of passive optical Earth observation variables.....	35
Figure 7: Flow of assignments in the model.	38
Figure 8: Model data structure hierarchy.....	39
Figure 9: Satellite network mixed integer program flow.....	54
Figure 10: HypsIRI mission concept architecture. Source: [35]	59
Figure 11: HypsIRI sensor architecture from 2012. Source: [35]	59
Figure 12: SSTL-150 ESPA [36] and SSTL-300 Information [37].....	61
Figure 13: HypsIRI example network node representation.....	62
Figure 14: Time-varying and time-invariant accesses.	65
Figure 15: HypsIRI example network problem.....	67
Figure 16: Base scenario network MILP solver readout.....	71
Figure 17: Base scenario network MILP data volume.....	71
Figure 18: Base scenario network MILP object assignments.	71
Figure 19: Base scenario network MILP architecture total cost.....	71
Figure 20: Base scenario network MILP, minimizing cost, data throughput.....	72
Figure 21: Base scenario network MILP, minimizing cost, object assignments.....	73
Figure 22: The NASA systems engineering engine. Reprinted from NASA Systems Engineering Handbook, NASA, 2016, p. 6.....	76
Figure 23: Classical orbit elements of a celestial body. Reprinted from Wikipedia Commons. .	80
Figure 24: A whiskbroom scanning mirror sensor is show on the left. A pushbroom sensor is show on the right. Reprinted from Wikimedia Commons.	82
Figure 25: Sensor field of view as a function of number of optical forms and nadir angle.....	94
Figure 26: Satellite field of view for varying numbers of optics.....	95
Figure 27: Number of satellites required to meet required field of view as a function of number of optics on a satellite.....	96
Figure 28: Focal plane on a whiskbroom sensor with two rows of the same wavelength for multiple samples.	105
Figure 29: Focal plane on a whiskbroom sensor with a window dimension of two thus forming 2x2 aggregate pixels.....	105
Figure 30: Initial SEER-H cost estimate process.....	106
Figure 31: Assigning thermal regulation to sensor based on focal plane operating temperature rule.	108
Figure 32: Steps in final sensor SEER-H estimate.	110
Figure 33: Sensor options based on disaggregation and sensor scan method.	112
Figure 34: Model inputs sheet part 1 (excluding Wavelength requirements).	115
Figure 35: Model inputs sheet part 2 (excluding Wavelength requirements).	115

Figure 36: Test case wavelength requirements.....	116
Figure 37: Snapshot of the test case whiskbroom work breakdown structure in SEER.....	117
Figure 38: SEER-H estimate for the reflective bands sensor.	121
Figure 39: SEER-H estimate for the emissive bands sensor.	121
Figure 40: Base scenario network MILP link schedule.	140
Figure 41: Base scenario, minimizing cost objecting solver result	141
Figure 42: Added options scenario - maximize data to ground solver results.	143
Figure 43: Added options scenario - maximize data to ground - data throughput.	143
Figure 44: Added options scenario - maximize data to ground - data received at each ground station.	144
Figure 45: Added options scenario - maximize data to ground - architecture decisions.	144
Figure 46: Added options scenario - maximize data to ground - total cost.	144
Figure 47: Added options scenario – minimize cost - solver results.	145
Figure 48: Added options scenario – minimize cost - data throughput.	145
Figure 49: Added options scenario – minimize cost - architecture decisions.	145
Figure 50: Published HypsIRI Architecture	146
Figure 51: Data throughput of published HypsIRI architecture.	146
Figure 52: Cost of published HypsIRI architecture.	146

List of Tables

Table 1: Test case requirements.....	41
Table 2: Test case engineering constraints.....	41
Table 3: Test case focal plane input characteristics.....	42
Table 4: Test case optical form characteristics.....	43
Table 5: Test case thermal regulation options.....	43
Table 6: Test case calibrator options.....	44
Table 7: Test case electronics subsystem option.....	44
Table 8: Calculated Orbit variable parameters.....	45
Table 9: Test case whiskbroom imager characteristics.....	46
Table 10: Whiskbroom imager focal plane characteristics.....	46
Table 11: Test case pushbroom sensors attributes.....	47
Table 12: Test case pushbroom sensors' focal planes attributes.....	47
Table 13: Decoder data frame description.....	58
Table 14: Network MILP use scenario sensor orbits.....	68
Table 15: Network MILP use scenario sensors.....	68
Table 16: Network MILP use scenario ground station locations.....	69
Table 17: Network MILP use scenario bus inputs.....	69
Table 18: Network MILP use scenario ground station input.....	69
Table 19: Network MILP use scenario communication technologies input.....	70
Table 20: The systems engineering project phase descriptions. Reprinted from NASA Systems Engineering Handbook, NASA, 2016, p9.....	77
Table 21: Small satellite bus information.....	78
Table 22: Small satellite launch vehicles launch capacities compared to small satellite busses and payloads.....	79
Table 23: User inputs for the orbit assignment.....	81
Table 24: Variable parameters in the orbit data structure calculated during model execution.....	81
Table 25: User inputs for focal plane component definition.....	85
Table 26: Focal plane variable parameters in the orbit data structure calculated during model execution.....	85
Table 27: User input fields for wavelength data structures.....	86
Table 28: Wavelength variable parameters in the orbit data structure calculated during model execution.....	86
Table 29: Optical form user inputs.....	87
Table 30: Sensor data structure fields.....	89
Table 31: Satellite bus data structure contents.....	111
Table 32: Satellite data structures.....	112
Table 33: Architecture data structures.....	113
Table 34: Sensitivity specifications for the VIIRS instrument. Reprinted from Visible Infrared Imaging Radiometer Suite (VIIRS) Sensor Data Record (SDR) User's Guide Ver. 1.2, NOAA, 2013, p. 13.....	114
Table 35: Test case focal plane inputs.....	116
Table 36: Sensitivity information for the test case whiskbroom sensor.....	118
Table 37: SEER-H WBS Estimate for Whiskbroom Sensor.....	120

Table 38: Sensitivity results for the test case pushbroom sensors.....	122
Table 39: SEER-H WBS Estimate for Pushbroom Reflective Sensor.....	123
Table 40: SEER-H WBS Estimate for Pushbroom Emissive Sensor.	124
Table 41: SEER command file template for the Si PIN focal plane component.....	130
Table 42: Hypsiri example decoder	131
Table 43: Mission sensors JSON input file for network model.	132
Table 44: HypsIRI example access data.	133
Table 45 Additional ground sites for network MILP scenario.....	142
Table 46: Additional ground station for network MILP scenario.	142
Table 47: Additional communications technology for network MILP scenario.....	142

1 Introduction

This thesis presents work performed during the course of an internship at An Aerospace Company (AAC) and research performed at Massachusetts Institute of Technology (MIT) Lincoln Laboratory as part of a fellowship.

Both efforts entailed the development of architecture tradespace exploration models. The tradespace exploration model developed at AAC uses automation techniques, isoperformance, and constraint programming to rapidly construct potential space-based passive optical EO sensor architecture concepts which meet a given set of customer requirements. The tradespace exploration model developed at MIT Lincoln Laboratory is a satellite network mixed integer linear program (MILP) which is used for making system architecture decisions and estimating final architecture cost.

1.1 Space-Based Passive Optical Earth Observation Sensor Architecture Tradespace Exploration at AAC

AAC is a Government contractor with business units that design and manufacture sensor systems for aircraft, spacecraft, and ships. Within the spacecraft sensors category, AAC has earned a reputation for developing highly reliable and complex space-based passive optical Earth Observation (EO) sensors for critical government space missions. Such sensors are integrated on larger satellite systems and measure radiance reflected or emitted by the Earth.

The development lifecycle of a satellite system, from need identification to final delivery, follows the systems engineering process. NASA defines systems engineering as “a methodical, disciplined approach for the design, realization, technical management, operations, and retirement of a system. A ‘system’ is the construct or collection of different elements that together produce results not obtainable by the elements alone” [1]. The technical processes and phases which constitute a systems engineering project are shown in Appendix A. These system design, technical management, and product realization processes are used both iteratively and recursively throughout the product development lifecycle and at each level of the systems hierarchy.

Traditionally, AAC has only competed for performing the systems engineering activities for the sensor system and not the larger satellite system on which the sensor resides. Before releasing a request for proposal, a customer will identify the need for a passive optical sensor and specify performance requirements and constraints that allow the sensor to integrate successfully with other system components. AAC will then propose an initial sensor system concept that satisfies these requirements, a plan for delivering the sensor within a specified schedule, and the cost breakdown for performing all required activities. This proposed sensor system concept can be described by its architecture, defined by the Institute of Electronics and Electrical Engineers as “the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution” [2]. If AAC wins the competition with their proposal, they will then commence all further work necessary to deliver the sensor system according to plan.

While delivering sensors has been good business for a company like AAC, it is not uncommon for such organizations to pursue an expansion of their role to prime mission contractor. Within the EO sensors domain, this desire is enabled via recent developments in the areas of small satellite busses, new launch opportunities, and improved sensing technologies which are further described in Chapter 2. However, taking advantage of these developments to

effectively compete with other companies for prime mission contractor requires the ability to navigate an increasingly complex tradespace and propose system architecture concepts which meet customer needs in the most affordable way possible. In this regard, Chapter 3 of this thesis examines AAC's current processes for performing systems architecture concept generation of spaced-based, passive optical EO sensors, identifies capability gaps, and discusses changes required to explore new concepts enabled by ongoing developments in the space industry without increasing engineering resource expenditure. Chapter 3 subsequently proposes the application of integrated modeling, the concept of isoperformance, and constraint programming to improve the efficiency and efficacy of systems architecture concept generation and evaluation efforts.

Chapter 4 of this thesis details the implementation of a pilot EO-Architecture Isoperformance Model (EO-AIM) which uses the techniques proposed in Chapter 3 and automation to rapidly construct potential space-based passive optical EO sensor architecture concepts which meet a given set of customer requirements. For each concept, cost estimates are generated via integration with stakeholder-trusted cost modeling software, allowing cost to be treated as both an independent variable and consequence during concept evaluation and tradespace exploration. Finally, the model uses simple algorithms to identify potential satellite bus options for hosting each sensor architecture in orbit. The total cost of populating an entire constellation based on a sensor architecture is then estimated by using cost estimates for the sensor, satellite bus, and the best launch vehicle option capable of lifting the satellite(s) to orbit. Essentially, the EO-AIM creates all feasible sensor architectures from a user's inputs with varying run times depending on user settings.

Chapter 5 of this thesis discusses validation of the EO-AIM and Chapter 6 presents an example of using the model along with analysis of its results. Chapter 7 summarizes the work conducted at AAC and discusses how the ability to rapidly analyze a large number of both novel and heritage architecture concepts through the EO-AIM will allow AAC to better utilize engineering resources and provide maximum value to the customer when competing for business. Finally, Chapter 8 presents recommended future work for improving the EO-AIM. In Chapter 6's simple demonstration, it is shown that using the EO-AIM can increase architecture concepts generated by a factor of ten or more.

1.2 Use of a Satellite Network Mixed Integer Linear Program for System Architecture Tradespace Exploration

The final section of this thesis, Chapter 9, presents research conducted at MIT Lincoln Laboratory that is complementary in nature to the sensor architecture tradespace exploration work presented in Chapters 3 through 8. As an alternative to the simple satellite bus and launch vehicle compatibility algorithm discussed in Chapter 4, the use of a satellite network mixed integer linear program (MILP) for making system architecture decisions and estimating final architecture cost is explored. The satellite network MILP is formulated as both an assignment problem and a network maximum flow problem which must send sensor generated data to a ground user. Results of the MILP vary with the selected objective function and provide insights on the potential benefits of architecture decisions such as sensor disaggregation and the utility of introducing additional communication nodes into existing networks. The satellite network MILP is also capable of verifying network data volume throughput capacity and providing an optimized link schedule for the duration of the simulation. Overall, this research explores the general problem of optimizing use of limited resources for a given space-based sensor while

ensuring mission data needs are met. NASA's HypIRI mission is used as a case study to demonstrate and discuss the benefits of using the MILP. While some data was hypothetical due to lack of available information, using the MILP to examine alternative network architecture options for NASA's HypIRI mission resulted in a system architecture with 20% higher data throughput for marginally less cost.

2 Background

This chapter presents background on typical space-based passive optical system architectures and how such systems function. This chapter also discusses developments in satellite buses, launch provider options, and sensing technologies before concluding with a brief survey of new entrants into the Earth Observation (EO) domain. These entrants, who may one day compete with AAC, are also looking to capitalize on recent advancements within the space industry and offer new services.

2.1 Earth Observation Missions and Passive Optical Sensors

A typical concept of operations (CONOPS) for a satellite-based passive optical EO mission entails a satellite, or constellation of satellites, flown in an orbital configuration that enables visual access to geographic regions of interest. A passive optical sensor aboard the satellite measures spectral radiance reflected or emitted by the Earth in wavelengths ranging from visible light to longwave infrared. Depending on the configuration of the system, raw or processed data is transmitted to a ground segment which may further process the data before distributing it to users. The ground segment also monitors the satellite and sensor's overall health and status; sending commands to the Telemetry, Tracking, and Command (TT&C) subsystem aboard the satellite. A relevant example of a passive optical EO system is the Visible Infrared Imaging Radiometer Suite (VIIRS) which resides on the National Oceanic and Atmospheric Administration's Joint Polar Satellite System (JPSS), shown in Figure 1. The environmental data records produced by VIIRS and other JPSS sensors deliver "key observations for the Nation's essential products and services, including forecasting severe weather like hurricanes, tornadoes and blizzards days in advance, and assessing environmental hazards such as droughts, forest fires, poor air quality and harmful coastal waters." [3]

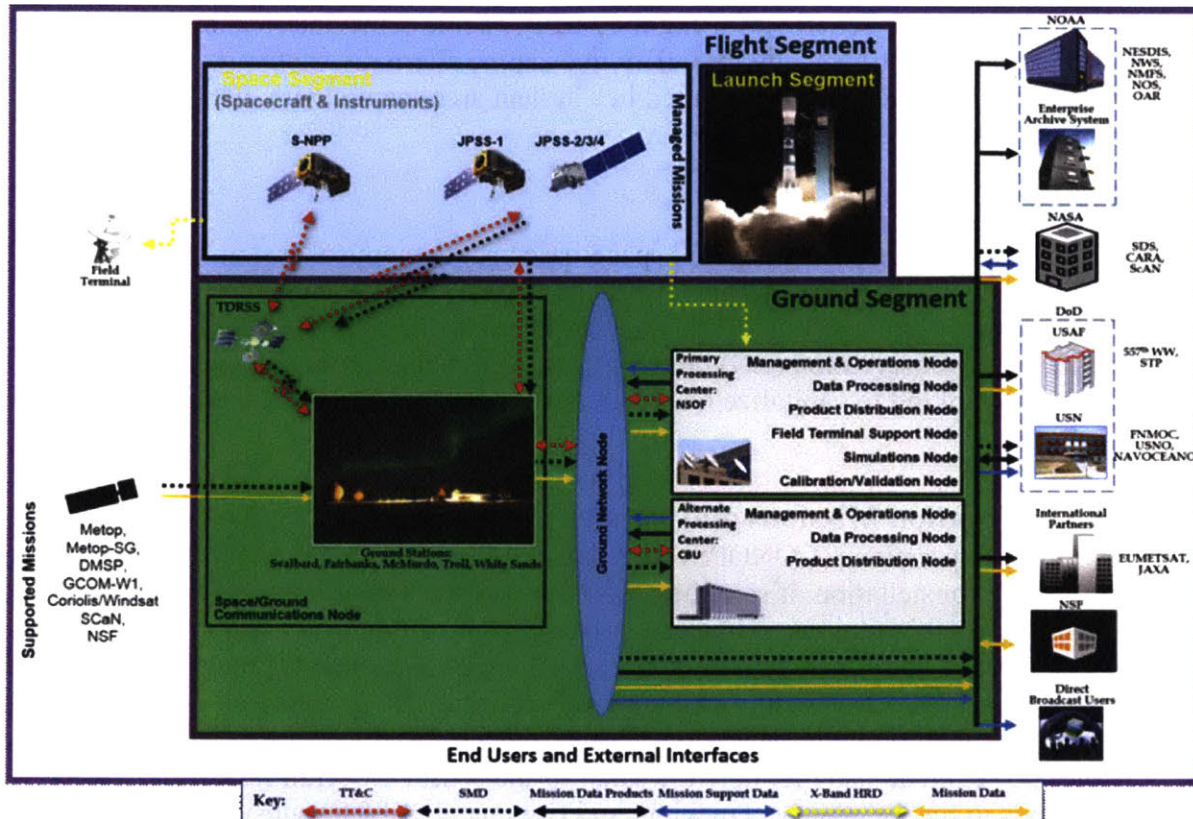
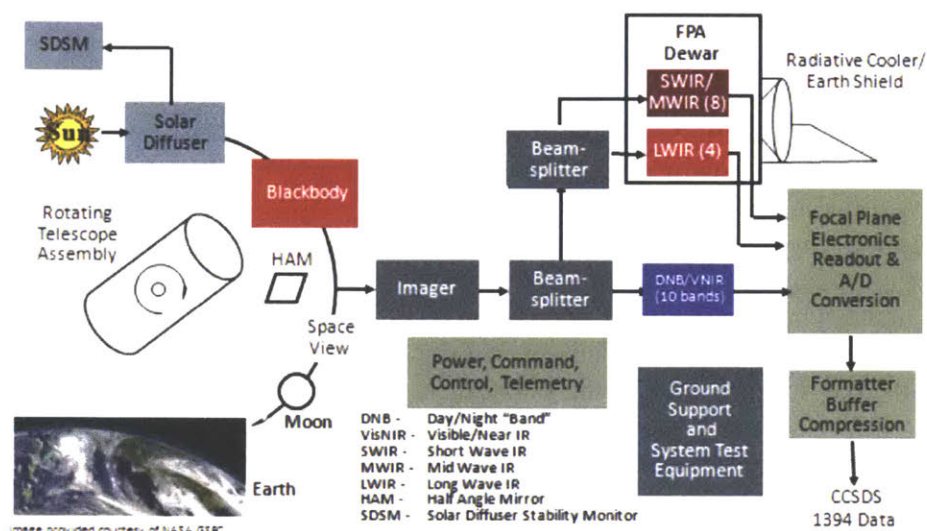


Figure 1: An illustrative example of a space-based EO system: the JPSS architecture. Source: JPSS Program Overview, 17 November 2010

VIIRS, shown in Figure 2, and other passive optical sensors work in the following way: Light is collected by a telescope and relayed to focal planes via a combination of optical elements such as mirrors, lenses, filters, etc. The focal plane array then generates an analog signal correlating to the amount of radiance detected in each spectral band. The analog signal is conditioned and digitized by focal plane interface electronics and processed, as required, by the sensor's electronics control unit. The digitized data is then transmitted to the ground segment via the satellite bus's communication subsystem or a standalone communications system belonging to the sensor. Throughout the collection process, calibration sources are periodically used to calibrate the sensor's response to detected radiance. Finally, a thermal regulation subsystem is used to cool the sensor and reduce focal plane noise sources.

In the case of VIIRS, the telescope is a three mirror anastigmat rotating telescope with a half angle mirror interface to a four mirror anastigmat aft optics assembly. Additionally, it can be seen in Figure 2 that VIIRS has focal planes for visible, shortwave infrared (SWIR), medium wave infrared (MWIR), and long wave infrared (LWIR) wavelengths. As far as calibration systems, VIIRS has a blackbody source for emissive wavelengths, and a solar diffuser for reflective wavelengths. A solar diffuser stability monitor (SDSM) is used to measure degradation of the solar diffuser over time. The thermal regulation subsystem for VIIRS is comprised of a passive cryoradiator and a dewar which encloses the SWIR/MWIR and LWIR focal planes. The dewar is a mechanical enclosure which houses the focal planes requiring cryogenic temperatures and is integrated with a thermal regulator via conductive thermal straps.

It is important to note that VIIRS is a multispectral imager which means it collects light at discrete wavelengths across a target spectrum. In contrast, a hyperspectral imager collects light continuously across the target spectrum at a much smaller spectral sampling rate, generally on the order of 10-50 micrometers. This is accomplished by an additional component called a spectrometer. Spectrometers are not discussed in this thesis, other than as an extra component required to go from multispectral to hyperspectral imaging. The addition of a spectrometer is typically accompanied by changes with the focal planes and the aft optics assembly.



(a) Block Diagram

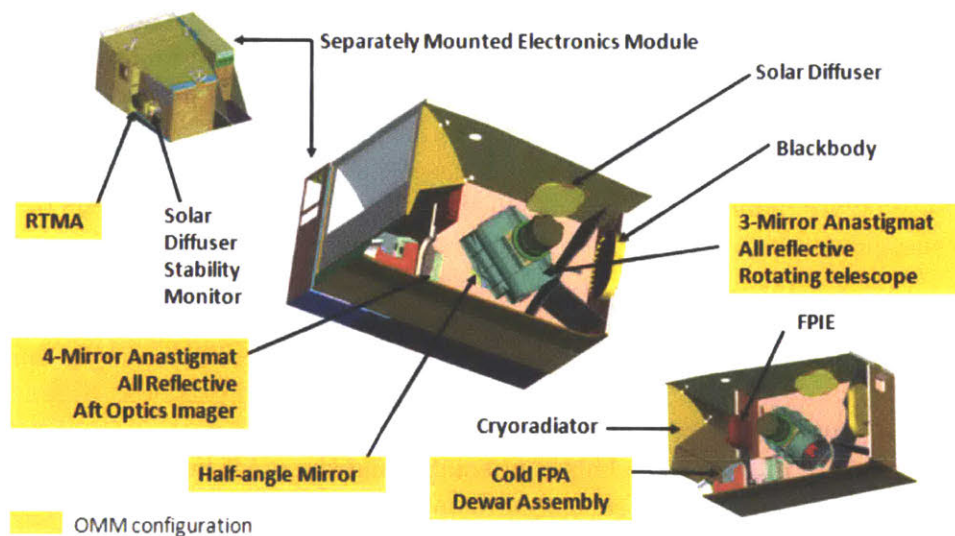


Figure 2: VIIRS block diagram and computer aided design model [4].

2.2 Developments in Architecture Options

A space-based passive optical EO sensor can be characterized by a specific type of focal plane technology and optical form. To characterize the entire mission system, one must include the spacecraft (or satellite bus) on which the sensor is integrated, the orbital plane in which the spacecraft operates, the launch vehicle that inserts the spacecraft into orbit, and the ground system that communicates with the spacecraft. This section discusses developments with satellite bus technologies, launch vehicle providers, sensor technologies and how each allows for the pursuit of new system architecture concepts.

2.2.1 Developments in Satellite Bus Technology

Over the last few decades, Government institutions, academia, and commercial space enthusiasts have been interested in making space more affordable and accessible as a whole. Academia, in its pursuit of access and hands-on education opportunities invented the cubesat standard in 1999 with 1U representing a satellite that is 10 cm by 10 cm by 10 cm and 1.5 kilograms. Since the standards inception, the number and use of common form factors has steadily increased, leading to a number of commercial component and subsystem manufacturers such as Blue Canyon Technologies and Clyde Space. Indeed, there are now entire online market places dedicated to the retail of cubesat components and launch canisters. While there are still many challenges that prevent a VIIRS like mission from being performed by a constellation of cubesats, EO and remote sensing have accounted for 43% of nano/microsatellite launches from 2009 to 2016 and are expected to account for 64% of launches from 2017 to 2019. [5]

Small satellite bus technology has also advanced in the last decade as seen in Table 21 of Appendix B, which contains data for small satellite busses advertised online. This information demonstrates that there is growing competition in providing platforms that allow for lower cost dedicated launch vehicles or rideshare opportunities enabled by compatibility with the Evolved Expendable Launch Vehicle (EELV) Secondary Payload Adapter (ESPA).

In general, taking advantage of cheaper cube satellites and small satellites enables the use of cheaper launch options, which are discussed in Section 2.2.2. This trending cost incentive may favor using distributed or disaggregated remote sensing architectures rather than the monolithic systems traditionally fielded. Such architectures also allow companies to take advantage of economies of scale and tolerate increased risk. For example, companies such as Planet and OneWeb are planning to mass produce satellites and reap the benefits of the learning curve and economies of scale. [6]

2.2.2 Developments in Launch Vehicle Options

In addition to changes in the satellite bus market, the launch services domain has continued to see dramatic changes starting with the entrance of SpaceX in the mid-2000s. Since then, SpaceX has continuously grown its business as witnessed by the company's projection of having 45% of the launch market share in 2017 [7]. This is not surprising since Air Force budgetary numbers released in 2017 indicate SpaceX offers a significantly lower cost ride to orbit than once monopolist United Launch Alliance (ULA) [8]. Other launch providers such as Europe's Arianespace have ambitious goals to compete with SpaceX on price per kilogram with their next-generation Ariane 6 rocket [9].

There are also several new ventures hoping to enter the launch services market for small satellites within the next decade, some of which are presented in Appendix B. This data includes a startup company called Spaceflight Industries which offers secondary or group launch

rideshare coordination services. A prime example of how the space industry is commercializing, Spaceflight Industry's president describes the company as providing "payload rideshare opportunities similar to how Expedia sells airline tickets. Making it easy and affordable is key to lowering the entry barriers to this market." [10]. Another achievement worth noting is Rocket Lab's successful launch of the Electron Rocket, which offers dedicated launch of satellites up to 225kg in mass for five million USD per launch, in January 2018 [11].

Overall, it can be hypothesized that these cheaper launch opportunities will lead to increased commercial satellite ventures, which will in turn lead to increased launch utility and launch provider competition; resulting in cheaper launch opportunities in a reinforcing feedback loop. Given the significant cost savings some of these launch options present, sensor designers need to consider the impact specific sensor architecture decisions ultimately have on their ability to take advantage of these launch opportunities while conducting trade space exploration.

2.2.3 Developments in Sensor Technologies

Finally, improvements in underlying sensor technologies also increase the trade space complexity of passive optical sensor systems. Trends in focal plane technology indicate pixels will continue to get smaller and array sizes will increase [12]. This advancement allows for mechanically simpler pushbroom architectures to be considered over whiskbroom imagers and has already allowed the Landsat mission to evolve from a whiskbroom imager to a pushbroom imager [13].

Another possibility is the use of novel technologies such as uncooled microbolometers. In contrast to a photoconductor (which converts photons into electrons), a microbolometer measures radiance by detecting the change in electrical resistance of the detector material as a function of the infrared radiation heating the material. A microbolometer array with 50 micrometer pixels was demonstrated in the THEMIS instrument as part of the Mars Odyssey mission. Since THEMIS, microbolometer pixel pitch size has decreased to 12 micrometers, making microbolometers increasingly attractive candidates for use in long wave infrared (LWIR) EO since their use eliminates the need for power hungry and expensive active cryocoolers [14].

Other sensor advancements may come from the Defense Advanced Research Projects Agency Wafer Scale Infrared Detectors (WIRED) program launched in 2015. WIRED aims to address current capability gaps in infrared technologies, including the need to cool medium wavelength infrared (MWIR) focal planes [15]. Outside of the United States, a CubeSat which is part of the QB50 mission was launched carrying new uncooled 50 micrometer MWIR focal plane technology from the European company New Infrared Technologies [16]. The use of uncooled technologies in the MWIR promises to make the design of MWIR sensors much more affordable.

While new sensing technologies may be riskier from lack of heritage, their employment may enable significant savings to be realized in other areas of the system (such as allowing a smaller satellite bus to be used). Thus, it is important to consider these advancements when creating passive optical remote sensing architecture concepts.

2.3 New Entrants to the Earth Observation Domain

As developments in technologies that promise to make Earth observation missions more affordable progress and the remote sensing customer base expands, the threat of new entrants and disruption increases [17]. Already there have been several startups formed in the remote sensing domain over the last several years that may one day add to AAC's current competitors.

As an example, Planet, a startup out of San Francisco, is taking advantage of CubeSat technology to field a distributed remote sensing architecture. With the goal of providing global visible imagery on a daily basis, they have already launched 88 of their 3U “Dove” satellites. Other startups of note include Spire, Satellogic, Capella Space, and Urthecast; all of which sell data as a service. Satellogic even touts itself as the “sole supplier of –high-resolution, hyper-spectral imagery in the world” since NASA EO-1 was decommissioned in March 2017. [18]

Compounding this potential threat is Government interest in harnessing these commercial developments via data service contracts. The National Geospatial Agency (NGA) has already purchased a one-year subscription to Planet’s data service which bolsters their existing data contract with DigitalGlobe [19]. NOAA has also begun looking into using data from small satellite constellations with its Commercial Weather Data Pilot program [20], called for by the Commercial Satellite Weather Bill signed into law in April 2017 [21]. While AAC has a strong reputation for providing high reliability sensors, it must be wary of new competition emerging as technology progression, such as potential uncooled MWIR focal planes, lowers barriers to entry at the same time Government customers entertain riskier (yet cheaper) missions and data contracts due to budgetary pressures.

In summary, AAC needs to evaluate how advancements in the small satellite market, the rise of more affordable launch options, and how smaller or new sensing technologies can be used to their advantage when pursuing the role of prime contractor. It is increasingly important to look at novel architectures, like disaggregating sensors, as customers increasingly look to industry to find creative solutions [22]. To this end, architecture trades studies including new developments must be assessed against mission performance and cost requirements to determine the optimal solution. If AAC fails to evaluate such options, they may one day lose out to new startups who are willing to take more risk.

3 Sensor Architecture Tradespace Exploration Approach and Method

The approach for studying how best to conduct the architecture concept selection phase given the increasingly complex tradespace included two major activities. First, it was necessary to obtain a fundamental understanding of how passive optical EO systems work. This led to extensive review of literature on EO systems and space mission architecting; including review of newer systems engineering methods for architecture concept comparison. Second, a review was performed on how AAC conducts architecture concept trade-studies to form proposals and whether established processes could support efficient evaluation of novel architectures enabled by the rapidly changing tradespace. The review included studying established AAC processes and tools, following the progress of an ongoing architecture special study, interviewing engineers and cost estimators, and reading through various internal documents.

After conducting these reviews, a solution to automate architecture concept generation using the techniques of isoperformance and constraint programming was proposed and subsequently pursued. During solution development, funding was secured from AAC's innovation fund to work with experienced engineers to create parametric relationships for certain sensor subsystem characteristics. These relationships were used to constrain the problem such that it could be solved.

3.1 Literature Review

Key texts for learning how Earth observation systems and remote sensing systems work include Schott's *Remote Sensing: An Image Chain Approach* [23], Wertz et al.'s *Space Mission Engineering: The New SMAD Design* [24], Eismann's *Hyperspectral Remote Sensing* [25], and Fortescue et al.'s *Spacecraft Systems Engineering* [26]. This was bolstered by academic articles and public database information on Earth observation systems such as EOportal.com. Reviewing system engineering methodologies for analyzing architectures resulted in the adoption of the isoperformance concept, described in Section 3.4.1, towards Earth observation space systems. Additionally, a study of available computational tools and modeling software at AAC was conducted.

3.2 Interviews and Process Analysis

To assess AAC's approach in formulating a proposal and creating an initial system architecture in response to a Government "request for information" or special study, several engineers were interviewed and ongoing efforts were analyzed. It was found that the process typically unfolds in the following way:

1. A Government requirements document, or a request for proposal, is received by AAC.
2. Senior engineers within AAC analyze the Government's requirements.
3. AAC engineers review heritage systems that could be leveraged to meet requirements.
4. An initial concept is proposed and rough order of magnitude performance models are created to verify requirements are met. The models are created with spreadsheets and are similar to those found in Chapter 17 of Wertz et al.
5. Other system tradeoffs are conducted through spreadsheet calculations, simulation, and system engineering tools such as the Pugh matrix technique. This work is bolstered by more detailed radiometric sensitivity calculations which are conducted by two senior engineers. Geometric propagation simulation is also used to ensure the proposed constellation configuration meets stakeholder coverage and revisit time requirements and has adequate access to ground stations. System tradeoffs included investigating which satellite bus or hosted payload options may be compatible with the sensor.
6. A rough order of magnitude cost estimate for building the sensor is then generated via comparison to other programs or through parametric cost modeling using SEER-H.

3.2.1 Description of Tools Used at AAC

Tools used throughout the process include Excel for radiometric sensitivity modeling, AGI Systems Tool Kit (STK) for satellite constellation modeling and simulation, and Galorath's SEER-H for cost estimation. STK and SEER-H are quickly described in this section.

AGI Software's Systems Tool Kit (STK) is used for modeling and simulation of satellite constellations. STK is a geometric propagation model that can compute multiple figures of merit for a system including satellite coverage of the Earth and access (or link) duration between any two points. STK in AAC's process is primarily used to compute the number of satellites needed to meet coverage requirements given a particular constellation configuration and to predict communication access times between the constellation configuration and a ground segment.

Golarath Systems describes SEER for Hardware, Electronics, & Systems (SEER-H) as a "decision-support tool that reliably and accurately estimates the total cost of ownership for new product development projects" [27]. SEER-H uses parametric modeling to estimate a component or subsystem's cost based on key characteristics and is used by NASA, other

Government entities, and private industry. The parametric models used by SEER-H were developed via analysis of actual costs accrued by past Government and commercial space programs. A few of the major inputs into each SEER-H parametric model are:

- Organization labor rates
- Key technical parameters for each component
- Design complexity and reuse
- Reliability standard requirements
- Company knowledge and experience with the development environment
- Company experience with the production environment
- Schedules and quantities

As with all parametric models, the quality of estimation depends on the model's inputs. To aid with this, Golarath systems provides detailed guidance for each input. Additionally, Golarath has published *SEER-H Space Guidance Ver. 2.2* ([28]) which contains instructions for setting default inputs and creating work breakdown structures for space-based instruments. The software also provides the capability for an organization to customize SEER-H and "tune" the cost model based on their specific history. Generally, an organization's SEER-H estimation is accompanied by another estimate from a different technique (e.g. the "bottoms up" approach).

3.3 Review of Findings

Analyzing AAC's process for conducting trade studies resulted in several findings. First, AAC has a tremendous amount of expertise in designing a sensor which meets specific stakeholder requirements. However, this expertise has generally relied on major architecture decisions (such as orbit definition and host bus integration constraints) being provided by the customer. This has been driven in part by the customer's need to manage requirements for several remote sensing instruments, all being built by different companies, and needing integration onto a single bus. A byproduct of these past practices is that there are no codified tools or processes for systematically evaluating different options that result from changing these architecture design parameters along a feasible range when they are not given by a stakeholder. Instead, past programs and experience anchor initial sensor designs which feed back into decision variables such as orbit and bus options. While this methodology results in a feasible solution, it favors past designs and may prematurely bound the solution search space; eliminating novel concepts which are now possible given advancements in technologies.

As an example, given a specific region of interest on the Earth, there are at least two ways of satisfying coverage and revisit time requirements. One architecture entails a few satellites in highly elliptical orbit (HEO) with a long dwell time above the region of interest. The other option is a larger constellation of satellites in circular low earth orbit (LEO) orbit. There are many differences between these two architectures, with the most notable tradeoff involving the number of satellites in the low circular orbit constellation versus the required instrument size to meet ground sampling distance and spatial resolution requirements at peak altitude in HEO. In the past, the HEO solution would likely be pursued as the manufacture of satellite busses, sensor to bus integration, and launch costs represented significant investments. However, the smaller focal plane technology of today integrated into cheaper small satellites with relatively inexpensive rideshare or dedicated launch opportunities is a promising disruptive alternative. Indeed, the lower launch cost per satellite may make the customer more risk tolerant, allowing

for lower mission assurance requirements and thus an even cheaper mission. Such concepts are covered in detail in [29].

To adequately compare these architectures using current processes, AAC would have to develop two initial sensor designs which equally meet mission requirements at the different orbits. Furthermore, there would have to be extensive knowledge of which satellite busses or hosted payloads are capable of hosting these sensors, followed by an analysis of the expected launch costs for populating the constellation. The analysis of both architectures requires a team to follow the process steps outlined in Section 3.2 twice; doubling the amount of effort expended by AAC.

Since the cost of exploring alternatives or new concepts quickly adds up, it is hard to justify looking at multiple architecture concepts in detail when the only chance of recouping such cost is winning a high value, winner take-all development contract. However, with ongoing developments in the commercial satellite and launch sector, means of exploring new tradeoffs rapidly without incurring costs associated with the current process is incumbent for AAC to remain competitive. While the HEO solution previously discussed may do the job, if a competitor is able to propose the circular LEO constellation using new bus and launch technologies at lower costs, AAC may miss opportunities. Other trade studies which warrant continuous feasibility analysis include sensor disaggregation, building to lower mission assurance standards, and the use of novel technologies such as uncooled MWIR focal planes or cube satellite busses. An added factor of complexity in conducting these analyses is the limitation of personnel to carry out these tasks, which leads to the second finding.

While AAC possesses the personnel talent to carry out the system architecting task, those with extensive knowledge are usually limited in availability. This limitation can reduce the amount of trades studies conducted, cause delays should another team member requiring clarification, or decreases responsiveness to customer changes. These impacts are amplified by lack of sound knowledge institutionalization and transfer mechanisms. If work is accomplished, being able to leverage it often depends on knowing of its existence through informal channels (e.g. word of mouth). And even if it is found, understanding the work's conclusions is often difficult due to lack of documented decision analysis or assumption rationale. In the worst case, duplication of effort occurs because there is no knowledge sharing, the work is lost when the author departs, or the work becomes indecipherable and unsubstantiated without the author. Once more, it may be the case that a key individual's departure leaves subsequent team members left wondering why a specific decision was made in the first place.

As an example, if a member within AAC conducts a small satellite trade study for hosting a sensor, there is no established database for controlling and sharing the knowledge gained on satellite bus vendor rates across AAC's teams. If the individual leaves the company, so does the knowledge and someone else may need to reinvestigate the matter at a later date. These occurrences can be common in a large organization like AAC where balancing a work force across projects leads to some amount of task and workforce volatility.

While there is an expiration date on any research, it would be helpful if there was a centralized repository where employees can institutionalize findings (with the source and date last accessed) so others can leverage the knowledge at a later date. The recommendation is also applicable towards cost modelling in SEER-H. A repository of SEER-H files for components used on previous projects would be tremendously helpful in creating future cost estimates and support standardization of work break down structures. Before embarking on any effort to create new knowledge, an employee could check to see it already exists.

A third finding related to the second, is that spreadsheet models used to make key design decisions early on are not intuitive to a general user as they are often unstandardized. While it's reasonable that only a few individuals may need to fully comprehend how conclusions are made, it would be efficient if some transparency and commentary existed within the models. A standard template, populated with explanations and references to supporting texts, would improve communication between team members and enable fresh perspectives to be involved in the initial brainstorming phases of architecture concept generation.

3.4 Solution to Findings

A solution to the findings discussed requires a means of using knowledge bases and tools to quickly construct architecture concepts with associated stakeholder-aligned evaluation metrics, like cost, while minimizing use of engineering resources. The solution should also have the flexibility to change knowledge bases or inputs using standard data formats so that new developments in areas such as satellite busses, launch vehicles, or focal planes can be input with little effort or change. Finally, the processes for creating output architecture solutions should be properly documented, transparent to the users, and leverage trusted modelling tools like SEER-H for calculating metrics such as cost.

Such a solution would not replace engineers engaged in the current processes, but rather automate repetitive spreadsheet work like sensitivity modeling and provide meaningful heuristics that engineers can use to justify spending their limited time on investigating the architecture concepts which are most promising; regardless if they're novel. Additionally, being able to examine the difference in output metrics between multiple architecture concepts allows engineers to better understand the tradespace and impact of specific decisions. The use of knowledge bases also provides a means of institutionalizing architecture options amongst the entire organization.

The systems engineering tool of isoperformance and constraint programming were ultimately used to create a pilot automated architecture creation tool, dubbed the Earth Observation Architecture Isoperformance Model (EO-AIM), which meets these requirements.

3.4.1 Isoperformance

Isoperformance is the concept of enumerating through architectures which uniformly satisfy a fixed set of requirements and evaluating those architectures' viability in terms of stakeholder criteria such as cost, risk, etc. [30]. The goal when using an isoperformance approach is to find as many solutions as possible, of which a subset will form an optimal frontier along the specified evaluation criteria, rather than solve for just one solution which might be optimal in only one criteria. This allows for engineering analysis when stakeholder requirements are clearly defined and offers insights into the inherent tradeoffs between performance, risk, and cost.

Of importance is how isoperformance only results in solutions which meet the requirements as defined. This aligns with how government contracting works: the government specifies requirements and a contractor must propose a solution which satisfies them. There is no advantage to over designing the system as the contractor will only be evaluated against the criteria specified in section M, "Evaluation Factors for Award", of the request for proposal as directed by the Federal Acquisition Regulations (FAR). In this regard, isoperformance is suited to how AAC should operate when generating architecture concepts for a government customer.

The inherently multi-disciplinary nature of passive optical sensors also makes this domain a good candidate to benefit from isoperformance. The complexity of equations,

variables, and constraints discussed in this thesis, especially in calculating sensitivity performance (which is nonlinear), make it difficult to configure a multi-objective optimization problem. Instead, isoperformance using systems engineering rules provides a framework for solution search, which is discussed further in the context of constraint satisfaction problems.

The first step to creating an isoperformance model is identifying the requirements for input into a model. Outputs of an isoperformance model are performance invariant architectures which satisfy these input requirements.

3.4.2 Identification of Key Stakeholder Requirements for Passive Optical EO Systems

To determine the isoperformance input requirements, several Earth Observation missions were examined for commonalities in stakeholder requirements. Four recurring requirements were found that significantly influenced the design of each systems architecture:

Data required by mission. This refers to the specific wavelengths, and the required sensitivity while sampling each wavelength, the system is required to collect in order to successfully produce mission data products. It is assumed that trade studies are conducted prior to a government agency's request for proposal to identify the necessary wavelengths and sensitivities required to produce accurate products. As far as sensitivity, this requirement uses one of two figures of merit: signal to noise ratio (SNR) or noise equivalent temperature difference (NEdT). Signal to noise ratio is defined by the measured signal of the observation over the noise of the sensor. Noise equivalent temperature difference describes the temperature that produces a signal equivalent to the sensor's noise. Sensitivity is generally specified in terms of SNR for reflective bands and NEdT for emissive bands.

Coverage. The ground coverage required by the Earth observation system. Coverage ranges from regional to global.

Timeliness or Revisit Rate. The maximum or threshold amount of time elapsed between data collections at a specific region or point on Earth. This is a different requirement from availability, which is the time delay from data collection at the sensor to receipt of the data by the user.

Spatial Resolution. The maximum or threshold resolution of the sensor. For instance, a 200 m x 200 m resolution means that observed area of the observation system is sampled in 200 m x 200 m pixels. Figures of merit for spatial resolution include the Modular Transfer Function (MTF) or Ground Sampling Distance (GSD).

While these are not a conclusive set of requirements for a system, they provide sufficient description such that system architecture concepts can be generated by constructing an isoperformance constraint satisfaction problem solution search using integrated modeling. It should also be noted that work covered in this thesis is relevant only to passive coverage systems, such as Earth science missions. Expanding this research to include targeting systems is discussed in future work.

3.4.3 Constraint Satisfaction Problems

Once the isoperformance requirements were defined, a systems engineering analysis was conducted to determine how such requirements are satisfied given a set of available concepts within the space-based passive Earth observation domain. Analysis of Earth observation missions revealed that optical sensors contain a common set of architectural and subsystem decisions with some decisions imposing constraints on others. For instance, every sensor has a telescope. However, the choice of which telescope to use will be constrained by the choice of sensor scan method. This leads to a helpful framework for constructing an isoperformance based architecture concept generation tool: defining the observation mission's architecture as a constraint satisfaction problem.

A constraint satisfaction problem is defined by a state of variable assignments and the required conditions, or a set of constraints on the variables, for a state to be considered a solution [31]. In this case, the state is the set of variable assignments which define the Earth Observation sensor architecture. A solution would be a state of variable assignments which ultimately meet the isoperformance requirements (performance constraints) and all other constraints which are derived from the workings of a passive optical Earth observation system (such as subsystem compatibility constraints). A variable's domain is the set of options that the variable can be assigned. A constraint is defined by the subset of variables it applies to (the constraint's scope) and a set of valid assignments to the subset of variables for the constraint to be satisfied. Mathematical equalities and inequalities derived from physics are typical constraints along with more abstract rules such as "if the focal plane operating temperature is less than T , then a cryocooler must be present." An analysis of various Earth observation system architectures was used to define variables and constraints, ultimately leading to the model discussed in this thesis.

Constraint satisfaction problems are typically solved through search and propagation of constraints. Search entails "branching" from a previous state into multiple states whenever there are multiple valid options for the next variable assignment and continuing down a solution search path with each option assigned to the preceding state. Propagation of constraints, or inference, is used to eliminate values from the domains of unassigned variables by enforcing constraints whose scopes include the just-assigned variable. Constraint propagation mirrors the step a system engineer takes in limiting the domain of system choices left based on choices previously made. However, if there are multiple valid options then a systems engineer might **search** or analyze the effect of using each option.

Figures 3 through 5 demonstrate the concepts of search and constraint propagation. Given a sensor scan method, we can create a rule as to whether a particular telescope can be used. In Figure 4 constraint propagation occurs after assignment of the whiskbroom scan method and eliminates the possibility of using "Telescope 2." Search is then used to look for at least two different solutions; one using "Telescope 1" and the other using "Telescope 3."

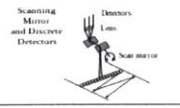
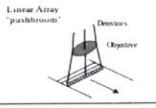
	 Scanning Mirror and Discrete Detectors	 Linear Array "whiskbroom"
Telescope 1	Allowed	Not Allowed
Telescope 2	Not Allowed	Allowed
Telescope 3	Allowed	Not Allowed

Figure 3: Constraint rules for what telescopes can be used given a scan method.

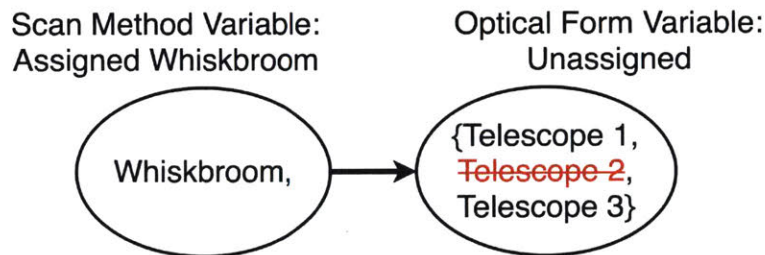


Figure 4: Constraint propagation eliminating "Telescope 2" from optical form/telescope variable domain.

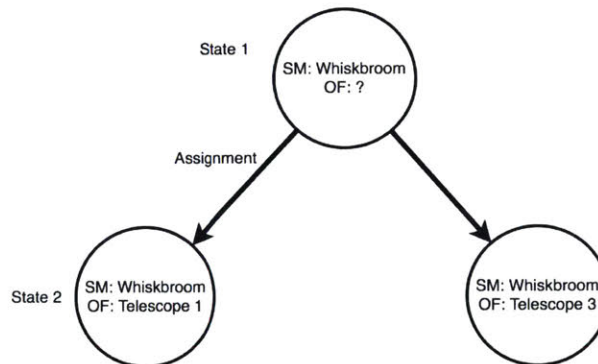


Figure 5: Branching to search for at least two different solutions using different optical forms.

3.5 Modeling Passive Optical Earth Observations as a Constraint Satisfaction Problem

As previously discussed, each sensor operates at a specified orbit and is composed of a telescope assembly which collects light via a scan method. Light is then transmitted from the telescope to one or more focal plane arrays via an optics assembly, which may include dichroic mirrors, filters, a spectrometer, etc. The focal plane arrays convert spectral radiance into a signal which is digitized and processed by an electronics module. Depending on the focal plane technology, a thermal regulation subsystem may be required to limit system noise. Additionally, calibration subsystems are used to ensure conversion of radiance into analog and digital signals is accurate. Finally, the sensor transmits its data directly to a ground segment or to its host bus. The host bus

is composed of the standard set of satellite subsystems: telemetry, tracking and command (TT&C), power, thermal, data handling, mechanism, propulsion, attitude determination and control, and structures.

Each piece of the passive optical earth observation system described and shown in Figure 6, is an architectural variable or subsystem variable. The domain of each subsystem variable is the set of all valid component combinations that, when integrated, form a subsystem capable of performing the functions required by the subsystem variable within the context of the system. A component can be a composite of fixed values and parameter variables. For instance, in the case of a focal plane, the pixel pitch is treated as a fixed value whereas the size of the focal plane array is a variable that is dynamically calculated to satisfy coverage and sensitivity constraints (requirements). The size of a focal plane array is also constrained by the choice of optical form, which is in turn constrained by the choice of sensor scan method. In general, the model is constructed such that the order in which variables are assigned allows for constraints to propagate, or be imposed, so that other variables can be solved by equation(s) or rule. Again, this intuitively mirrors the process a systems engineer would follow when solving the problem. For example, an engineer constructing an EO systems uses common sense to come up with an initial value for the required aperture diameter of a sensor's telescope only after selecting an orbit.

The order of assignments by the EO-AIM starts with what are defined in thesis as "architecture decision variables," followed by assignment of the "critical technology subsystem variables," and then finally "enabling subsystem variables."

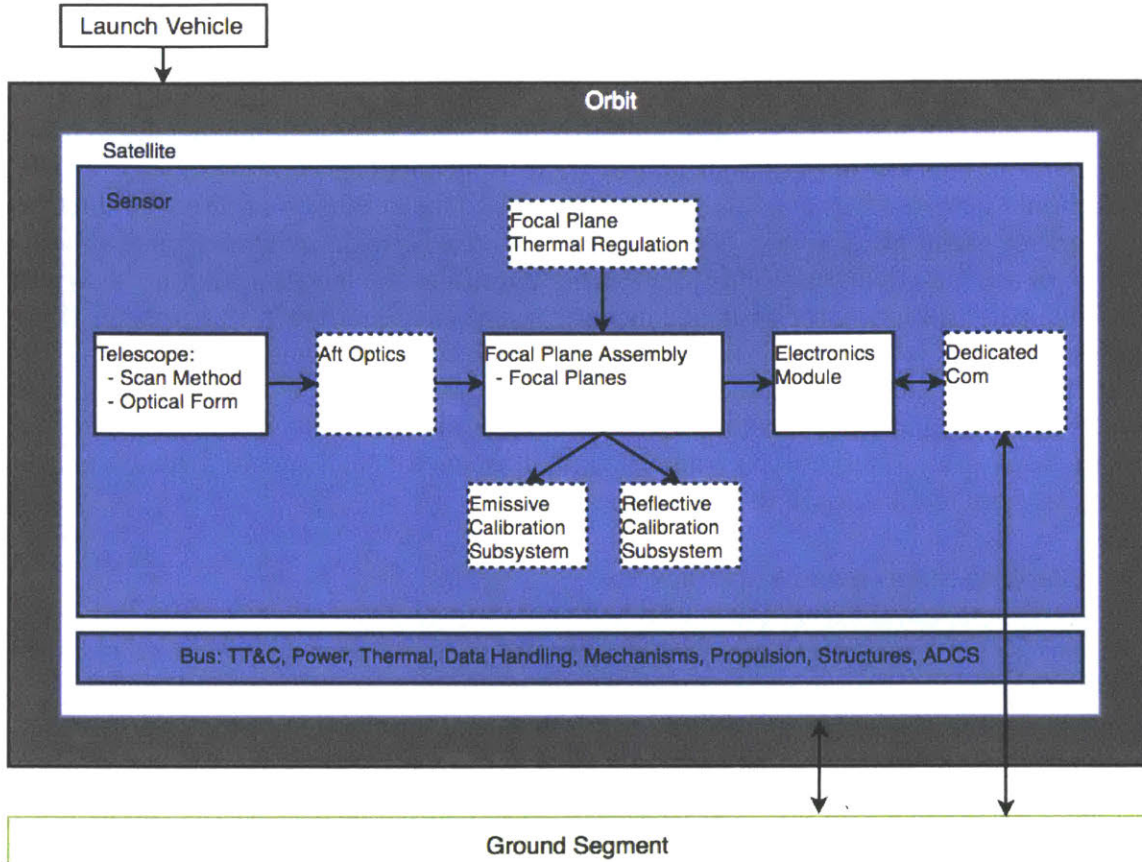


Figure 6: Block diagram of passive optical Earth observation variables.

3.5.1 Sensor Architecture Decision Variables

Sensor architecture variables affect the number of sensors required in orbit and the performance modeling of the sensors given available component technologies. They represent the higher-level sensor architecture decisions that will impose constraints upon the rest of the cyber-physical design. Thus, subsystem and component variables are dependent on the values of these architecture decision variables.

In passive optical Earth observation systems, there are three architecture decision variables: 1) orbit selection; 2) focal plane disaggregation scheme, and 3) sensor scan method. Together these attributes activate a set of constraints which are used to calculate or assign subsystem and component variable values which, when assigned, result in an initial sensor solution that satisfies stakeholder performance requirements. In line with an isoperformance analysis, the model (EO-AIM) enumerates through all possible architecture decision variables when creating solutions. Descriptions of the orbits, possible disaggregation schemes, and sensor scan methods are in Appendix C.

3.5.2 Critical Technology Subsystem Variables and Components

Critical technology subsystems and components are those which influence key sensor characteristics, like focal length and aperture diameter, and constrain the domains of non-critical “enabling subsystems”. In the EO-AIM, the critical technology subsystems are optical form and a subset of focal plane variables. As previously discussed, focal plane assemblies are comprised

of fixed-value parameters, such as pixel pitch, and variable parameters, such as array size. While a given focal plane technology's fixed values are defined during user input, and the technology is assigned to a state by the focal plane disaggregation scheme assignment, the focal plane's variable parameters aren't fully assigned until after the optical form assignment. Descriptions of the focal plane and optical form data structures are in Appendix C.

It should be noted that creating the model required the definition of three parametric relationships for each optical form. The first relationship is between maximum field of view and entrance pupil aperture diameter. This relationship constrains the model's solution search to only find realistic solutions which balance meeting sensitivity requirements against the desire to maximize a sensor's field of view. The second parametric equation calculates the optical form's largest element size, which is required for cost estimating, as a function of entrance pupil diameter. The third parametric relationship estimates mass as a function of aperture diameter and requires further work. The specific parametric relationships for each optical form are considered proprietary and are not discussed in this work.

3.5.3 Enabling Subsystems Variables

Enabling subsystem variables are those which are constrained given choices in critical technologies. For instance, the thermal regulation subsystem assignment will be constrained by whether the focal plane chosen requires a cryogenic operating temperature. The enabling subsystems are thermal regulation, calibration systems, and the electronics module. Further description of these subsystems is in Appendix C.

3.6 Chapter Summary

This chapter discussed the approach used for analyzing AAC's current processes for conducting passive optical EO sensor architecture tradespace exploration and concept selection, identified gaps within these processes, and introduced concepts which were ultimately implemented in a model (EO-AIM) to address process gaps. Key concepts introduced were isoperformance and constraint satisfaction problems. Finally, a typical Earth observation sensor architecture was defined as a constraint satisfaction problem with three distinct variable types: architecture, critical technology, and enabling subsystem decisions variables. The next chapter discusses the implementation of the EO-AIM

4 Earth Observation Architecture Isoperformance Model (EO-AIM) Implementation:

Once the relationships and constraints between the variables of passive optical sensor architectures were mapped out, algorithms were developed to read user inputs and perform architecture concept solution search. This included the development of sensitivity models for different types of focal planes and the creation of performance sensitivity algorithms which determine the "optimal" sensor parameter values, such as focal length and aperture diameter, necessary to meet requirements. The resulting Earth Observation Architecture Isoperformance Model (EO-AIM) ingests a set of user inputs and outputs a set of performance invariant architecture concepts.

User inputs include stakeholder requirements, model options, engineering constraints, and component information; all of which must be entered prior to executing the model. After reading all inputs, the EO-AIM assigns values to system variables as previously discussed and

shown in Figure 7. The order of operations discussed in this section represent the actions in going from one state (circles in Figure 7) to another as shown by the directional arrows in Figure 7. Assignments are made using one or more of the following techniques:

- System engineering use case “rules” determine candidates for assignment.
- Key parameters are calculated based on the “proposed” assignment.
- In some cases, constraint checks are calculated and, if violated, result in an alteration of the proposed assignment via system engineering rules (this is true for the sensitivity modeling).

At each depth in the solution search tree, branching occurs if there are multiple valid assignments. For instance, in the case of assigning a reflective calibrator subsystem to a state which already contains a reflective focal plane, the preceding state will be duplicated and two subsequent states will be formed: one with the vicarious calibration method and one with the solar diffuser. If the preceding state does not contain a reflective focal plane, then both the vicarious calibration and solar diffuser calibration methods will be eliminated from the reflective calibration variable domain, leaving only the “None” option. This domain pruning occurs via rules driven inference. For other variables, such as “orbit,” there are no rules and every orbit specified by the user constitutes a valid assignment. In this case, the branching factor is the size of the number of altitudes input as circular low earth orbit options.

The first variable assignments made by the model are the orbit, focal plane disaggregation scheme, and the sensor scan method. The first subsystem assigned is the optical form. After the optical form assignment, the model uses a sensitivity performance algorithm with systems engineering rules to find “initial” sensor solutions which satisfy sensitivity performance requirements and other constraints imposed by the architecture decision variables and optical form values. The performance algorithms for each scan method calculate the sensitivity performance for each wavelength as a function of the focal plane data structure variables, aperture diameter, and focal length. Additionally, the sensitivity model used is specific to the type of focal plane and how it produces a signal. This work includes models for both photocarrier focal planes and microbolometer focal planes, which can be found in Appendix D, Sections 5 and 6. The successful assignments to the orbit, focal plane disaggregation scheme, scan method, optical form, and focal plane variable parameters result in an “initial” sensor solution.

After creation of an initial solution, system engineering rules and constraints are used to assign values to enabling subsystem variables and their component variables with the end result referred to as a “final” sensor. In other words, the remaining subsystems, with their constituent components, are assigned to an initial sensor state resulting in an architecturally complete sensor.

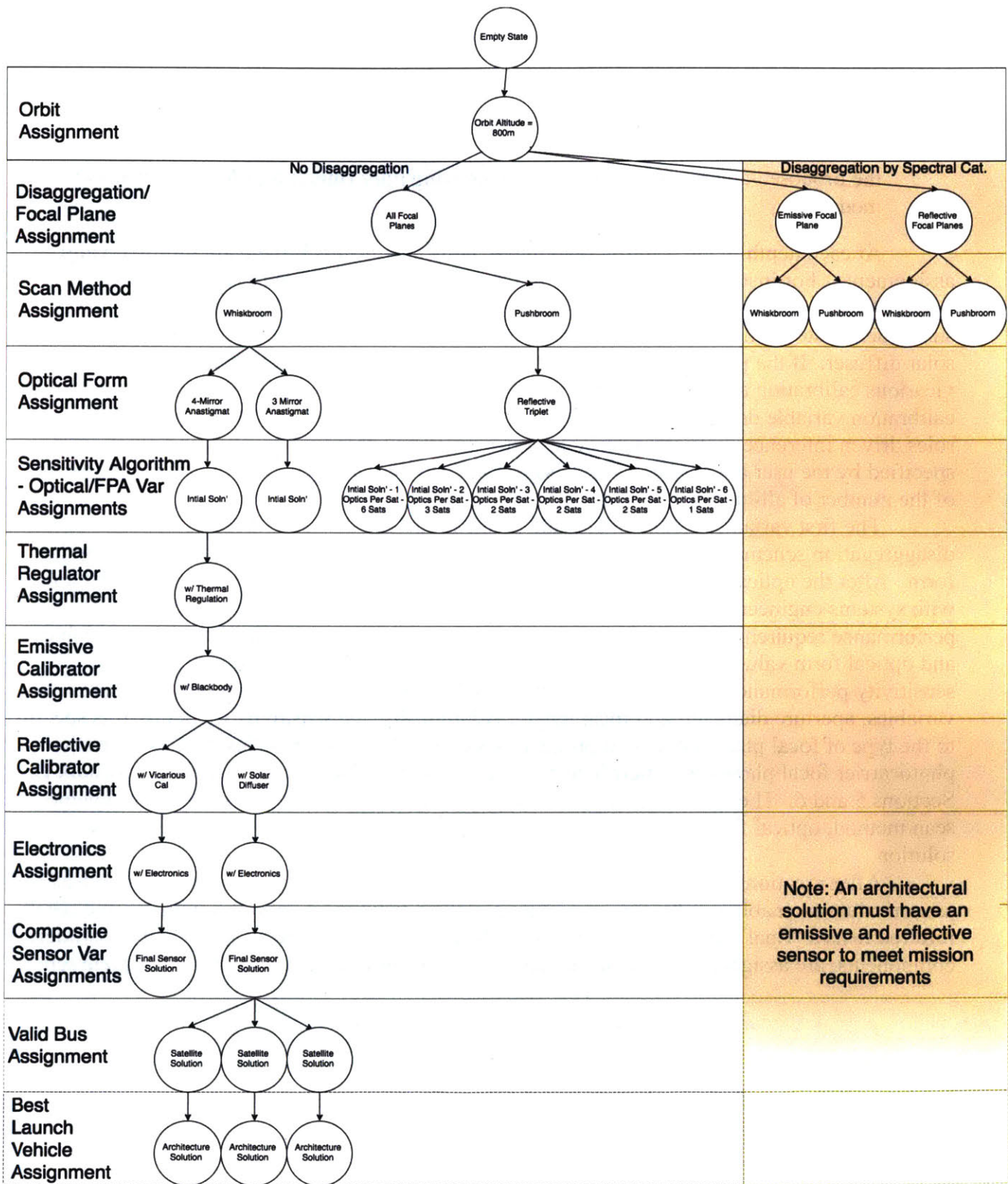


Figure 7: Flow of assignments in the model.

Once final sensors are created, all possible sensor architecture concepts are then formed by the model. A sensor architecture is defined as a subset of final sensors, in an orbital configuration, which meet mission requirements. For example, a sensor architecture might be comprised of a pushbroom visible/near infrared (VSNIR) sensor and whiskbroom thermal infrared sensor. This is followed by creation of mission architectures through bus and launch vehicle assignments to a sensor architecture using a simple algorithm. These assignments include all options for sensor satellite disaggregation. For instance, the VSNIR pushbroom and thermal whiskbroom sensors could reside on the same bus or be further disaggregated onto different busses which are launched separately. A more comprehensive alternative approach to creating a mission architecture using a mixed integer linear program is explored in Chapter 9 of this thesis.

The model’s final output is an array of mission architecture concepts which all meet stakeholder requirements. Each mission architecture is comprised of the data structures shown in Figure 8. The nested structure format enables evaluation criteria to be calculated from data queried at different levels and ensures some amount of configuration control. For instance, total architecture cost is a summation of the sensor costs, satellite bus costs, and launch vehicles costs each multiplied by the quantity required to fulfill mission requirements.

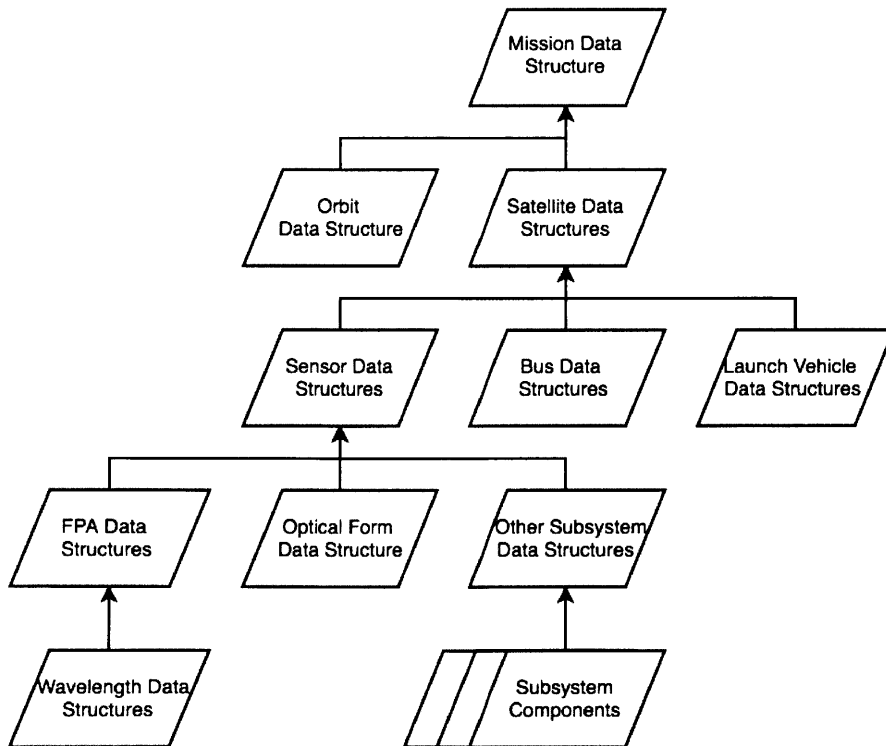


Figure 8: Model data structure hierarchy.

During the model’s execution, sensor cost estimates are created for initial and final sensor solutions using SEER-H. The reason for creating cost estimates for both the initial and final

sensor solutions is to reduce the overall run-time of the model. The initial sensor cost estimate file, containing work breakdown structure information for the optical form and focal plane characteristics, serves as a seed file for several final sensor estimates (the total depending on the number of search branches during enabling subsystem variable assignments). This method should be re-examined during future work for opportunities to increase the speed at which the EO-AIM runs.

In order to support automated generation of SEER-H cost estimations, each component input into the modeling tool requires an associated unique text file which provides the skeleton command lines for creating the component in SEER-H. These template files are edited using a “rules” file which contains use case checks or commands for changing parameters within the SEER-H command file to match those of the component. For instance, a focal plane’s SEER-H command file’s array size values would be edited to reflect the array size necessary to meet sensitivity requirements, found through execution of the sensitivity performance algorithm. The text files for all of the components are then combined into a single command text file followed by MATLAB execution of batch files to create the SEER-H cost estimation for each sensor. Appendix E contains more detail for each step of the model’s operations. An example of a SEER-H command file can be found in Section 4 of Appendix G.

5 Validation of the EO-AIM

The EO-AIM validation activities involved validating the sensitivity models and constraints used, validating the parametric equations used to characterize components, and validating of integration with SEER-H.

5.1 Validating the Sensitivity Models

Validating the sensitivity model for this project entailed using information from an ongoing special study to compare results from the automated MATLAB scripts to those generated via excel spreadsheet by senior systems engineers. In this sense, the model was validated against the status quo of wavelength sensitivity performance prediction for sensor system proposal concepts but not against actual fielded sensor performance. The MATLAB sensitivity model was shown to provide the same results as the Excel engineering models previously used by AAC for a variety of inputs.

5.2 Validating Component Parametric Equations

The parametric equations for the optical form’s aperture diameter versus field of view relationships were derived with the help of an experienced optical engineer at AAC. While a best effort was made, limited resources prevented an exhaustive effort from being undertaken to validate the relationships further; all of which were derived from at most a few fielded examples of each optical form. This limitation is partly due to the small sample size with which AAC can draw from internally since optical space systems up until recent years have been low volume, long time horizon efforts. A recommendation going forward is to analyze means of improving these parametric equations via the use of optical design software packages such as Zemax Optical Studio. Additionally, more complete mass parametric equations for optical forms can be derived from a compilation of past computer aided software designs.

5.3 Validating Sensor Cost Modeling

The integration with SEER-H was validated by examination. Going forward, it is incumbent upon AAC to calibrate SEER-H’s cost estimates with past designs. Auto-generating accurate cost estimates with SEER-H will prove enormously valuable for communicating with stakeholders.

6 Model Use, Results, and Discussion

This section demonstrates use of the model by providing a set of inputs and examining two output solutions. The input requirements, engineering constraints, subsystem options, model user options, and engineering constraints are presented before discussing the model’s results.

6.1 Mission Requirements

The mission requirements input into the demonstration use case, shown in Table 1, are similar to VIIRS requirements. The key difference between the test case requirements and VIIRS requirements (Appendix F) is the revisit rate, which is set to six hours in this example. Important engineering constraint values used in the test case are listed in Table 2. For reference, the model input window is shown in Figure 34 and Figure 35 of Appendix G. It can be seen that the demonstration is limited to an orbit of 800 kilometers, includes both whiskbroom and pushbroom scan method options, and creates options for both no disaggregation (‘combined’) and spectral category disaggregation. Disaggregation by required focal plane thermal regulation was not included because it results in the same disaggregated focal plane assignments as spectral category disaggregation. The wavelength requirement inputs are show in Figure 36 of Appendix G. They mirror the VIIRS requirements shown in Appendix F.

Requirement Description	Threshold
Data Required	The same wavelengths and sensitivities as the VIIRS moderate resolution bands (M15).
Coverage	Global
Revisit Rate	6 Hours
Spatial Resolution	375m

Table 1: Test case requirements.

Engineering Constraint	Value
Sun Synchronous Orbit Required?	Yes
Minimum Elevation Angle	20 degrees
Cross Track Overlap Factor	1.05
Along Track Overlap Factor	1.05
Assumed Heat Load	2W

Table 2: Test case engineering constraints.

6.2 Test Case Focal Planes

Characteristics of the focal planes used in this test run are shown in Table 3. These focal plane parameters are representative and are not based on any specific real-world focal planes. Quantum efficiency and dark noise look up tables were based on [32]. A screenshot of the input window for the focal planes can be found in Appendix G.

Field	VISNIR Focal Plane	SMWIR Focal Plane	LWIR Focal Plane
Lookup File	SiPIN_lookup.xlsx	HgCdTeMWIR_lookup.xlsx	HgCdTeLWIR_lookup.xlsx
Focal Plane Type	PhotocARRIER	PhotocARRIER	PhotocARRIER
Lower Bound Wavelength (meters)	4.00E-07	1.00E-06	7.50E-06
Upper Bound Wavelength (meters)	1.00E-06	7.50E-06	1.30E-05
Spectral Type	reflective	emissive	emissive
Material	Si PIN	HgCdTe on Silicon	Pv HgCdTe
COTS Configuration?	yes	yes	yes
X-Dim. Pixel Pitch (meters)	1.00E-05	1.80E-05	1.80E-05
Y-Dim. Pixel Pitch (meters)	1.00E-05	1.80E-05	1.80E-05
X-Dim. Number of Pixels	1.20E+03	1.50E+03	1.90E+03
Y-Dim. Number of Pixels	1.20E+03	1.50E+03	1.20E+03
Operational Temperature (K)	300	60	60
Thermal Regulation Required?	no	yes	yes
Bits Per Pixel	14	14	14
Operational Mode	N/A	N/A	N/A
Electron Well Capacity Per Pixel (e-)	2.00E+08	2.00E+08	2.00E+08
Readout Noise (e-)	80	80	80
Default Frame rate (Hz)	60	60	60
Max Readout Rate vs Frame Rate (Gbps)	2.72	2.72	2.72
Dark Current (A/m ²)	Lookup	Lookup	Lookup
Spatial Noise(e-)	50	50	50
Johnson Noise (e-)	0	0	0
NEP (Microbolometers)(W)	N/A	N/A	N/A
Quantum Efficiency	Lookup	Lookup	Lookup
Microns Between Bands	50	50	50

Table 3: Test case focal plane input characteristics.

6.3 Test Case Optical Form Options

For the test case, three optical forms were input: two supporting the whiskbroom scan method and one for the pushbroom scan method. Characteristics of the optical forms are shown in Table 4. It should be noted that all characteristics and equations have been modified and do not represent actual data. Additionally, the power relationship for the whiskbroom scanning mechanisms has been omitted.

Field	Optical Forms		
Reference String	RT REP	3MA RT	4MA
Name	Reflective Triplet (Real EP)	Three Mirror Anastigmat	Four Mirror Anastigmat
Use Case	Pushbroom	Rotating Telescope	Scanning Mirror
Real EP?	Yes	Yes	Yes
Real XP?	No	Yes	Yes
Number of Elements (Mirrors/Lenses)	3	3	4
Max FOV Constraint (degrees)	20	3	10
Max Element Size Constraint (m)	N/A	N/A	N/A
General Relationship Constraint (m*deg ²)	70	3	14
Largest Element Function	@(EP,FOV) EP*(1+0.08*FOV)	@(EP,FOV) EP	@(EP,FOV) EP*(1+0.15*FOV)
Minimum F Number Constraint	2.8	3.2	2.4
Wavelength Constraints	None	None	None
Along Track Constraint	3	3	3
Mass (kg)	@(LES) 240*LES ^{2.2}	@(LES) 240*LES ^{2.2}	@(LES) 300*LES ^{2.2}
Power (W)	0	0	0

Table 4: Test case optical form characteristics.

6.4 Test Case Thermal Regulation Options

Focal plane thermal regulation options for the test case, shown in Table 5, include an active cryocooler, cryoradiator, and no thermal regulation. A representative Ross look up table was used for the active cryocooler.

Field	Cryocooler		Cryoradiator	No Thermal
System Name	Cryocooler	Cryocooler	Cryoradiator	No Thermal Required
Component	Cooler	Cryocooler Electronics	Cryoradiator	No Thermal Required
Reference String	CRYOCOOLER	CRYO_ELECTRONICS	CRYORAD	NOTHERMAL
Type	Cooler	Electronics	Cooler	Cooler
Min Temp (K)	50	N/A	100	200
Max Temp (K)	150	N/A	200	N/A
Mass (kg)	5	4	20	0
Power (W)	Interpolate	3	0	0
Ross Data File	cryocoolerPower_lookup.xlsx		N/A	N/A

Table 5: Test case thermal regulation options.

6.5 Test Case Calibrator Options

Calibrator options include the blackbody source for emissive bands and a choice between vicarious calibration and solar diffuser calibration for the reflective bands. Characteristics are shown in Table 6.

Field	Blackbody	Solar Diffuser		Vicarious
System Name	Blackbody	Solar Diffuser	Solar Diffuser	Vicarious Calibrator
Component	Blackbody	Solar Diffuser	Solar Diffuser Monitor	Vicarious Calibrator
Reference String	BBS	SDS	SDSM	VICAL
Type	Calibrator	Calibrator	Monitor	Calibrator
Spectral Category	Emissive	Reflective	Reflective	Reflective
Mass (kg)	@(EP) 15*EP	@(EP) 10*EP	@(EP) 3	0
Power (W)	N/A	N/A		0

Table 6: Test case calibrator options.

6.6 Test Case Electronics Module Options

Finally, the tool input included one option for the sensor electronics subsystem with three components that vary depending on the sensor. Component definitions can be seen in Table 7. The mechanisms control component is dependent on whether the sensor uses a whiskbroom scanning method. The quantity of data processors is dependent on the number of pixels sampled per second by the sensor. The quantity of the digital focal plane interface electronics component is dependent on the number of focal planes in the sensor.

Field	Electronics Subsystem				
System Name	Electronics	Electronics	Electronics	Electronics	Electronics
Component	Power Component	Processing Component	FPA Interface Electronics	Thermal Power Component	Mechanisms Component
Reference String	ELECTRONICS_POWER	ELECTRONICS_DP	ELECTRONICS_DFPIE	ELECTRONICS_HEATER	ELECTRONICS_MECHANISMS
Type	Power Controller	Data Processor	DFPIE	Thermal Control	Mech Control
Mass (kg)	@(MC) 1	@(MC) 1	@(MC) 1	@(MC) 1	@(MC) 1
Power (W)	3	3	3	3	3
Quantity	@(sensorType,MC,systemPower,focalPlanes) 1	@(sensorType,MC,systemPower,focalPlanes) sum(ceil([focalPlanes,pixelsPerSecond]/400e3))	@(sensorType,MC,systemPower,focalPlanes) numel(focalPlanes)	@(sensorType,MC,systemPower,focalPlanes) 1	@(sensorType,MC,systemPower,focalPlanes) 1*(strcmp(sensorType,'whiskbroom'))

Table 7: Test case electronics subsystem option.

6.7 Results and Discussion

The model took a little under thirty minutes to complete. During this time, it created thirty-six unique sensor concept data structures; sixteen within the no disaggregation schema, six emissive sensor options, and twelve reflective sensor options. Thus, the total number of isoperformance sensor architecture concepts including combinations of reflective and emissive sensors is eighty-eight. Before enumerating through subsystem options, there was twenty-three initial sensors which resulted from orbit, disaggregation, sensor scan method, and optical form assignments only.

Each sensor is accompanied by its own SEER-H cost estimate that includes the total cost of producing the entire quantity required to populate a constellation. Each sensor also has sensitivity, size, weight, and power summary tables which can be used for analysis.

Two sensors solutions will be presented in this section as an example for showcasing the tool's results. The first is a whiskbroom sensor with all three focal planes (no disaggregation). The second is a constellation of pushbroom imagers with disaggregated focal planes by spectral category (reflective versus emissive). To make things interesting, the sensor solutions that require two satellites in an orbital plane were selected. The orbit selection of 800 kilometers altitude resulted in the viewing angles, orbital period, maximum range to target, and number of orbital planes required to meet revisit rate requirements shown below in Table 8.

Orbit Variable Parameter	Value
Orbit Altitude	800km
Inclination Angle for SSO	98.6 degrees
Period	6,052.2 seconds
Ground Velocity	6,621.4 m/s
Minimum Nadir Angle to Cover Longitudinal Shift	55.38 degrees
Elevation Angle	22.15 degrees
Maximum Range to Target	1,673km
Required FOV to Cover Node Change Along Track	110.77 degrees
Number of Planes to Meet Revisit Rate	3 (conservative estimate)

Table 8: Calculated Orbit variable parameters.

6.7.1 No Disaggregation Whiskbroom Solution

The assignments of no disaggregation, whiskbroom scan method, and a three mirror anastigmat optical form resulted in the whiskbroom imager shown in Table 9 with focal plane attributes found in Table 10. An aspect worth noting is the relatively long scan time which enables longer time delayed integration and thus allows for smaller pixel areas when meeting sensitivity requirements. With regard to pixel size, it can be seen that each focal plane requires different windowing dimensions to meet sensitivity requirements (see Figure 29 of Appendix E). Finally, the pixels required in the along-track and cross track dimensions are within the commercial off the shelf (COTS) focal plane array size limitations, which presumably translates to lower procurement costs of the focal planes. The total number of pixels in the cross-track direction (X-dimension) includes the 50 micrometers spacing between wavelength filter coatings on the focal plane. Mass, power, and data rate are not included because meaningful numbers cannot be calculated without the use of company and component specific data and parametric equations. Further development of these equations is also a key recommendation for future work. A screenshot of the SEER-H file work breakdown structure is in Figure 37 of Appendix G along with the cost break out in Table 37. Sensitivity data for each focal plane can be found in Table 38.

Field	Whiskbroom Sensor
Orbit Altitude	800km
Focal Planes	[Si PIN, HgCdTe SMWIR, HgCdTe LWIR]
Sensor Scan Method	Whiskbroom
Optical Form	Three Mirror Anastigmat – Rotating Telescope
Scan Time	6.0249s
Aperture Diameter	.0535m
Focal Length	.1785m
F-number	3.33
Cross Track Field of View w/ Scan	1.933 rad (110.77 degrees)
Along Track Field of View	0.0524 rad (3 degrees)
Thermal Regulator	Cryocooler
Emissive Calibrator	Blackbody
Reflective Calibrator	Solar Diffuser
Electronics	Electronics

Table 9: Test case whiskbroom imager characteristics.

Field	Si PIN	HgCdTe SMWIR	HgCdTe LWIR
Framerate	4643Hz	5170Hz	4653Hz
Windowing Dimension	4	2	1
X-Dim. Pixel Pitch w/ Windowing	40 μ m	36 μ m	18 μ m
Y-Dim. Pixel Pitch w/ Windowing	40 μ m	36 μ m	18 μ m
X-Dim IFOV / Windowing	2.2415e-4 rad	2.0173e-4 rad	1.0087e-4 rad
Y-Dim IFOV w/ Windowing	2.2415e-4 rad	2.0173e-4 rad	1.0087e-4 rad
X-Dim Sampling Distance at Nadir	179.32m	161.38m	80.7m
Total X-Pixels	288	52	13
Total Y-Pixels	935	520	520

Table 10: Whiskbroom imager focal plane characteristics.

6.7.2 Disaggregated Pushbroom Results

Attributes for the two pushbroom sensors are shown in Table 11. Most notable are the differences between the two sensors required focal lengths and aperture diameters needed to meet ground sampling distance requirements. This is expected since focal length proportionally increases with wavelength as ground sampling distance (GSD) is held constant. Thus, the emissive sensor will require a longer focal length to meet GSD requirements.

Sensitivity and SEER-H cost estimate work breakdown structures for the two pushbroom sensors can be found in Appendix G, Section 3. Comparing the pushbroom imagers to the previously created whiskbroom imager, it can be seen that all three sensors differ in cost by quite a bit. The most significant drivers of costs are the thermal regulation subsystems and the size of the HgCdTe focal plane arrays. Since the pushbroom imagers, by nature, require larger HgCdTe arrays and there are many more sensors in the pushbroom constellation than the whiskbroom solution, the cost is much higher. Again, the SEER-H estimate is not accurate and should not be trusted until proper component templates using internal cost data are constructed. However, the capability for automating SEER-H estimates based on dynamic system calculations is demonstrated. Screenshots of the SEER-H panes for each pushbroom sensor are presented in Figure 38 and Figure 39 of Appendix G.

It is important to note that these sensor cost figures do not account for the estimated cost of the sensors' host satellite busses and available launch options. As the VISWIR pushbroom

has a small aperture diameter, a shorter focal length, and does not require a physical calibration system, it could conceivably be integrated into a smaller, less expensive cubesat. While the emissive pushbroom sensor is also smaller, the cryocooler would likely require a small satellite host to provide enough power. On the other hand, the characteristics of the whiskbroom imager are larger and the sensor would likely require the largest bus and most expensive ride to orbit. In this case, it may be that the pushbroom sensor architecture concept results in the cheapest mission architecture when considering available options for host satellite bus and launch provider.

Finally, it may be worth exploring whether the mission could be accomplished without the medium infrared bands. This would allow for the potential use of a microbolometer which is more affordable and does not require a cryocooler. In this way, the model supports evaluating what specific requirements “cost.”

Field	Reflective Sensor	Emissive Sensor
Orbit Altitude	800km	800 km
Number Required	6 (2 Per plane)	6 (2 Per plane)
Focal Planes	[Si PIN]	[HgCdTe SMWIR, HgCdTe LWIR]
Sensor Scan Method	Pushbroom	Pushbroom
Optical Form	Reflective Triplet	Reflective Triplet
Aperture Diameter	.0021m	.0301
Focal Length	.0251m	.0843
F-number	11.6822	2.8
Cross Track Field of View Per Optic	.3455 rad	.3455 rad
Along Track Field of View	0.0283 rad	.0045rad
Optics Per Satellite	3	3
Cross Track Field of View Per Satellite	1.0366	1.0366
Satellites Per Plane	2	2
Thermal Regulator	None	Cryocooler
Emissive Calibrator	None	Blackbody
Reflective Calibrator	Vicarious	None
Electronics	Electronics	Electronics

Table 11: Test case pushbroom sensors attributes.

Sensor	Reflective Sensor	Emissive Sensor	
Focal Plane	Si PIN	HgCdTe SMWIR	HgCdTe LWIR
Framerate	20.76Hz	38.75Hz	38.75Hz
X-Dim IFOV	3.9868e-4 rad	2.1358e-4 rad	2.1358e-4 rad
Y-Dim IFOV	3.9868e-4 rad	2.1358e-4 rad	2.2415e-4 rad
X-Dim Sampling Distance at Nadir	319m	179.32m	179.32m
Total X-Pixels	726	1618	1618
Total Y-Pixels	71	21	9

Table 12: Test case pushbroom sensors' focal planes attributes

7 Summary of the Earth Observation Architecture Isoperformance Model (EO-AIM) and Discussion of Potential Benefits

Thus far, this thesis has detailed the creation and use of a sensor architecture concept generation and evaluation model, EO-AIM, for passive optical Earth observation systems using automation techniques and the concepts of isoperformance and constraint satisfaction problems. The EO-AIM resulted from analyzing AAC’s current sensor architecture concept generation and

evaluation processes and identifying capability gaps which could prevent AAC from exploiting a rapidly evolving architecture tradespace composed of changes in satellite bus, launch vehicle, and sensor technologies. The effort to create EO-AIM also entailed performing a systems engineering analysis to identify relationships between key stakeholder requirements, architectural decisions, and subsystem tradespaces.

Use the EO-AIM was demonstrated for a specific set of customer requirements and two sensor architecture concepts were examined. The EO-AIM produced over thirty possibilities in thirty minutes with little effort expended by an engineer. This is quite impressive given AAC's previous processes might only look at three such architecture concepts in the course of several weeks. The demonstration exhibits how the EO-AIM satisfies the findings from examination of AAC's current architecture concept generation and evaluation processes presented in Chapter 3: the EO-AIM uses knowledge bases that can easily be changed for all system components, it's options allow for the analysis of specific scenarios, it creates many concepts quickly with minimal expenditure of engineering resources, and it produces heuristics for examining an architecture concept's utility through integration with trusted third-party software. Furthermore, the EO-AIM includes in-line comments and references that make it easier for an engineer to understand what is happening. Overall, use of such a model is projected to provide the following benefits:

- Improved brainstorming through novel concept generation.
- Insight into the impact a specific decision has on the downstream trade space.
- Improved competitive analysis and strategic decision making through the use of model heuristics.
- Codification of corporate knowledge.
- Rapid assessment of architecture trades.
- Improved communication across teams and with stakeholders.

7.1 Improved Brainstorming

Using the EO-AIM provides a means of generating novel architectures that might otherwise be overlooked or not considered due to limited resource availability. This is because there is no tradespace bounding or elimination of ideas early on. Instead there is a systematic enumeration of all possible sensor architecture concepts. With the old process, bounding was necessary given the number of assignment variables and the number of possible solutions which, in the worst case, is the product of the domain sizes of all of the decision variables. While this is intractable when a person is required to do the calculations or analysis, calculations can be done quickly with the use of automation.

Furthermore, given constant advancements in technologies, there could be architectures, such as pushbroom constellations, that are more affordable than previously thought and should be seriously considered for the first time. As such, its advantageous to not arbitrarily bound the solution search space based on prior history and instead run the model to see what is possible. In this sense, the model serves as an aid to brainstorming: it may identify sensor architecture concepts as promising that were previously not considered or even thought of.

7.2 Total Mission Insight

The EO-AIM enables an engineer to trace the impact of assignments made upstream on the resulting downstream trade space. For instance, an engineer debating whether there is benefit in trying a low-heritage microbolometer for LWIR instead of a photocarrier focal plane need only

run two scenarios to see the differences the assignments have on subsequent architectures created. One solution output may include options that use new small satellites bus options while the other does not. Furthermore, integration with SEER-H provides the capability to analyze projected cost differences at the sensor level and between different Mission Class levels. Golarath provides guidance on which settings should be toggled to change a sensor from mission Class A to mission Class C. Instead of cost being a result of sensor concept selection, it becomes an input into selecting which ideas warrant in-depth analysis and further resource expenditure.

7.3 Competitive Analysis and Strategic Decision Making

The EO-AIM supports analyzing potential competitor architectures. If there is a reasonable amount of competitive intelligence on what technologies competitors have or systems decisions they tend to make, one need only to use this information as model input to evaluate their competition. As an example, Company A may be a new entrant in the commercial sector and has expertise in the use of small satellites with lower mission assurance standards. Company B might have a history of building high reliability sensors that have traditionally flown on bigger government satellite systems. In pursuit of growth, Company A starts looking at the government as a potential customer; especially since they've heard the government is becoming increasingly interested in exploiting small, "cheap" satellites. Company B should at least evaluate the threat of Company A by hypothesizing what their potential solutions could be and how receptive the target customer may be to them; this includes estimating what such solutions will cost.

7.4 Codification of Corporate Knowledge

The model codifies both systems engineering best practices and architecture options. The algorithms for creating sensor solutions follow the same steps a senior systems engineer takes in analyzing the system tradespace. As an example, the EO-AIM seeks to use time delayed integration to improve sensitivity before increasing a sensors entrance aperture. This relies on systems engineering rules of thumb that state existing focal plane real estate should be leveraged before making changes to the system which increase total mass and size characteristics. Furthermore, if best practices should ever change, one can change the logic in the model. To help with a newcomer's understanding, detailed comments are kept within the code base of the model such that intent and reasoning behind the rules and algorithms is readily transparent to new users.

Likewise, options which define the domain of possibilities for system variables are institutionalized through the use of databases or spreadsheets. Information gathered by previous studies is incorporated into the model. This allows for sharing of previous work and shifts an engineer's work from reinventing-the-wheel to maintaining and updating previously gathered information. Should a new focal plane be invented, it can be added to the focal planes definition input file. The same is true for optical forms, thermal regulation subsystems, etc. This knowledge sharing includes the institutionalization of SEER-H templates for various components, which bolsters sharing, analysis, and understanding of cost estimates. Overall, institutionalization of knowledge cuts down on the time needed for a new member of the team to "catch up" and provides a more concrete base on which to innovate new ideas.

7.5 Rapid Assessment of Architectures

One of the major benefits of this model is the automation and standardization of what can be considered redundant "spreadsheet" work traditionally used to create heuristics for determining

which architectures are most promising and merit further investment of resources for initial design. Instead of spending several hours setting up a single sensitivity model in Excel or a rough cost estimate in SEER-H, an engineer can spend their valuable time analyzing several sensor architecture concepts generated by the model and confirm the most promising solutions. These solutions can then be further validated via detailed computer aided design and structure-thermal-optical performance analysis, resulting in increased confidence of the final solution selected. Overall, resources are saved during upstream concept generation and rough order of magnitude cost estimation so that resources can be spent in detailed validation steps later on for several concepts instead of just one or two. While the model could conceivably take a large amount of time as more information and options are added (24+ hours); the only resource cost is a small amount of ample processing power that can run continuously day and night.

Additionally, the use of knowledge bases allows engineers to quickly analyze the potential impact a new development or technology has on the architecture concept tradespace. New satellite busses, launch vehicles, or subsystems technologies can be added to the input files and the simulation re-run at little cost. In this way, the model can be run throughout the trade study as new information is gathered.

7.6 Improved Communication

Finally, model results can be used to improve communication within the team and externally to stakeholders. Outputs provide justification for why decisions are made or substantiation for shaping a customer's expectations. For example, the cost impact of requiring the collection of medium infrared wavelengths can clearly be shown based on cost estimates for the requisite focal planes. Additionally, team members can refer to the model's documentation when trying to understand why certain trades are made. The model's results (with the model version) can be archived as decision rationale for how any final choice in architecture concept is made. In other words, early decision rationale is easily documented and supported in the form of model results.

8 Future Work

There are several areas where the EO-AIM can be improved through future work. First, it is necessary to continue developing mass, power, and data rate parametric equations for optical forms and other components. Having reasonable estimates enables further analysis of which satellite bus and launch vehicles options might be possible. This insight is necessary to truly understand the impact specific decisions have on an architecture's total cost. Mass, power, and data rate values also allow for the use of mixed integer linear programs for assignment and network optimization as described in Chapter 9.

Another recommendation is to develop a SEER-H template database which leverages the SEER-H models created for other projects. This allows systems engineers to simply pull a command file template for a common part and adjust it accordingly. Additionally, serious effort should be made to calibrate SEER-H files to reflect realities specific to AAC (such as the cost of an HgCdTe focal plane). However, it is important to not make the model overly optimistic. SEER-H incorporates data sets from multiple contractors and provides a good benchmark based on past data that AAC does not have access too.

The model requires further work for implementing a stop and stare sensor at non-circular orbits. Implementing stop and stare sensors at non-circular orbits is challenging because the sensor's ground area coverage and resolution change continuously as the satellite changes altitude along its orbital path. This difficulty is compounded by a comparatively more complex

decision trade space for meeting requirements. An algorithm would need to be developed which uses rules for tweaking the system design until sensitivity requirements are met.

In support of the above effort, it would be beneficial for systems engineers to document the order in which they change attributes of a system, or “rules”, as they seek to meet customer requirements. For instance, should the focal plane be operated at a colder temperature to meet sensitivity requirements or should the aperture diameter be increased? Such a question might depend on whether the “colder temperature” requires the introduction of a power intensive and expensive cryocooler. The model was developed such that new rules can be introduced, though the computational time required increases with each.

With some modification, the EO-AIM can accommodate analysis of both hyperspectral imagers and missions involving fixed-point targeting (fixed-point targeting missions are those which desire the capability for a satellite to observe a specific target area for as long as possible through the use of a gimbaled telescope or satellite attitude control). Using the model to create hyperspectral imagers requires the introduction of new subsystem variables, components, and constraints. Using the model for fixed-point targeting missions can be made possible through tighter integration with STK using the ActiveX COM automation port and minimal changes to the coverage analysis algorithm.

9 Using a Satellite Network Mixed Integer Linear Program for System Architecture Decision Analysis

Chapter 4 of this thesis discussed using simple algorithms for determining which satellite busses could host sensors generated from the Earth Observation Architecture Isoperformance Model (EO-AIM). This approach is relatively simple and provides a quick and rough order of magnitude heuristic for the total cost of populating an entire mission constellation required by the sensor architecture. However, such an approach fails to consider the need to send data generated by mission sensors to a ground segment. While these considerations are not as critical for (by today’s standards) low data volume multi-spectral applications such as VIIRS, which sends a maximum of 10.5 Mbit/s [33], they are non-trivial when considering future high data rate applications. Choices in the satellite bus communication technologies, bus data storage, and limitations in the pre-existing ground and space data network architecture may result in the most promising sensor architectures in terms of cost, being intractable or incredibly costly in a total mission architecture context. For instance, certain sensor solutions may require satellite busses that are capable of transmitting large amounts of data per second or an elaborate ground infrastructure in order for requirements to be met. Thus, while the previous approach was expedient, it is less useful when evaluating how a given sensor architecture design translates into, or integrates with, a larger space and ground network responsible for transferring mission data generated on orbit to the user. This section details how the previous sensor architecture concept enumeration model can integrate with a satellite network mixed integer linear program (MILP) to create more informative cost estimates and aid in architecture decision making. First, the need for such a program will be explained in more detail, followed by discussion of the program itself.

9.1 The Need for a Network Model for Evaluating Architectural Decisions

Given a mission which requires an instrument with high data volume generation, several decisions arise for how best to transfer collected data to a user via a network composed of ground stations, data relay satellites, and other means of transferring data. Such a network might

include traditional radiofrequency communications with accesses to ground only or novel technologies such as free space optical (FSO) communications for use in inter-satellite links. Additionally, when constructing the problem, mission architects must decide whether it is best to integrate with existing networks already serving ongoing missions, introduce a new network altogether (e.g. build dedicated ground stations), or implement a solution between the two extremes. Overall, decisions for the ground and space architecture must be made in complementary manner. Decisions include:

- Should the sensors be disaggregated onto separate satellite busses?
- How much data storage is required on each satellite?
- Can inter-satellite links be used to balance peak data transmission times to ground? If so what technologies should be used in order to achieve a specific data rate?
- Does the existing ground station network meet mission needs? If not, where should new stations be added?
- If existing ground assets will be used, what is the best way to schedule downlinks to optimize sharing of ground assets while meeting data availability requirements?
- What communication technologies, at what data rates, are required between the space and ground segment?

Not only are these decisions extremely interdependent, but they are many and must be made such that mission requirements like data availability (the time it takes data to get to the user) are satisfied. During the mission architecting phase, the objective when making these decisions is usually to minimize the overall cost of the system, maximize data throughput to the user, or maximize some utility function which balances both cost and throughput. Even with the aid of performance modelling software such as STK, it can be imagined that such detailed evaluations prove extremely time consuming for a human to perform...and this is just for one sensor architecture proposal. What happens when there is a desire to evaluate the tradeoffs of many sensor architectures like the thirty-six unique options produced by the simple example in presented Chapter 6?

Once more, the inputs to these decisions are changing every few years and it becomes necessary to re-evaluate options as communication technologies, bus technologies, launch vehicles, and mission needs change over time. For instance, as launch costs for small/microsatellites become cheaper, it may be desirable to disaggregate sensors onto multiple vehicles at the operational expense of decreased mass and power budgets. Such a system might entail a hyperspectral imaging (HSI) visible and near infrared (VSNIR) sensor on one bus and a multispectral thermal imager on the other. To balance power requirements between the two busses, the thermal infrared system, which is power intensive from cooling, sends its data via FSO inter-satellite link to the HSI VSNIR satellite which hosts a high powered, high data rate downlink capable of transmitting both the VSNIR and thermal data to ground. How can the potential value of such a novel architecture quickly be compared to a more traditional, heritage-based proposal where both sensors are on a single satellite? Such an evaluation would also provide information on the utility of novel technologies such as FSO inter-satellite links. While, the case for using FSO inter-satellite and downlink communications for high data producing cubesat applications has been studied; the benefit over radio frequency communications with respect to small and microsatellite satellite applications remains harder to quantify.

To better evaluate a mission's architecture tradespace, a satellite network mixed integer linear program is presented in the following section which outputs an optimal architecture configuration and data routing schedule given an input of network architecture options and a sensor architecture concept (sensors and orbit configuration) which already meets coverage, revisit, and sensitivity requirements. Such a program naturally complements the work previously covered in this thesis; sensor architectures with associated costs estimates are generated and used to populate the network model along with other key subsystems which integrate together to form a larger system in which data generated on orbit is ultimately transmitted to the end user. For each sensor architecture input, the network model results in a total mission architecture solution which is optimal for some utility function.

9.2 Satellite Network Mixed Integer Linear Program

At a high level, the model takes user inputs, determines potential architecture assignments based on constraints, calculates data communication accesses between architectural components using geometric propagation software, creates a network mixed integer program, and integrates with a commercial solver to solve the mixed integer linear program. Depending on the model's inputs and selected objective function for the solver, the program can determine:

- Whether the instruments should be disaggregated,
- What satellite busses should be used to host the sensors,
- Which communications technologies should be present on each satellite,
- How much data storage capacity is required on each satellite,
- Whether additional satellite relay nodes are required/useful,
- Whether additional ground stations are required/useful,
- What communication technologies should be present at the ground stations,
- The data routing schedule of the architecture,
- How much data is transmitted to ground at the end of the simulation, and
- How much data remains stored on the satellite busses at the end of the simulation.

The models order of operations are shown in Figure 9. Each operation will be discussed in detail and is presented with a conceptual example using NASA's HypIRI mission. The example serves to demonstrate what each operation represents in the context of setting up the satellite network mixed integer linear program; which is both an assignment problem and a network maximum flow problem.

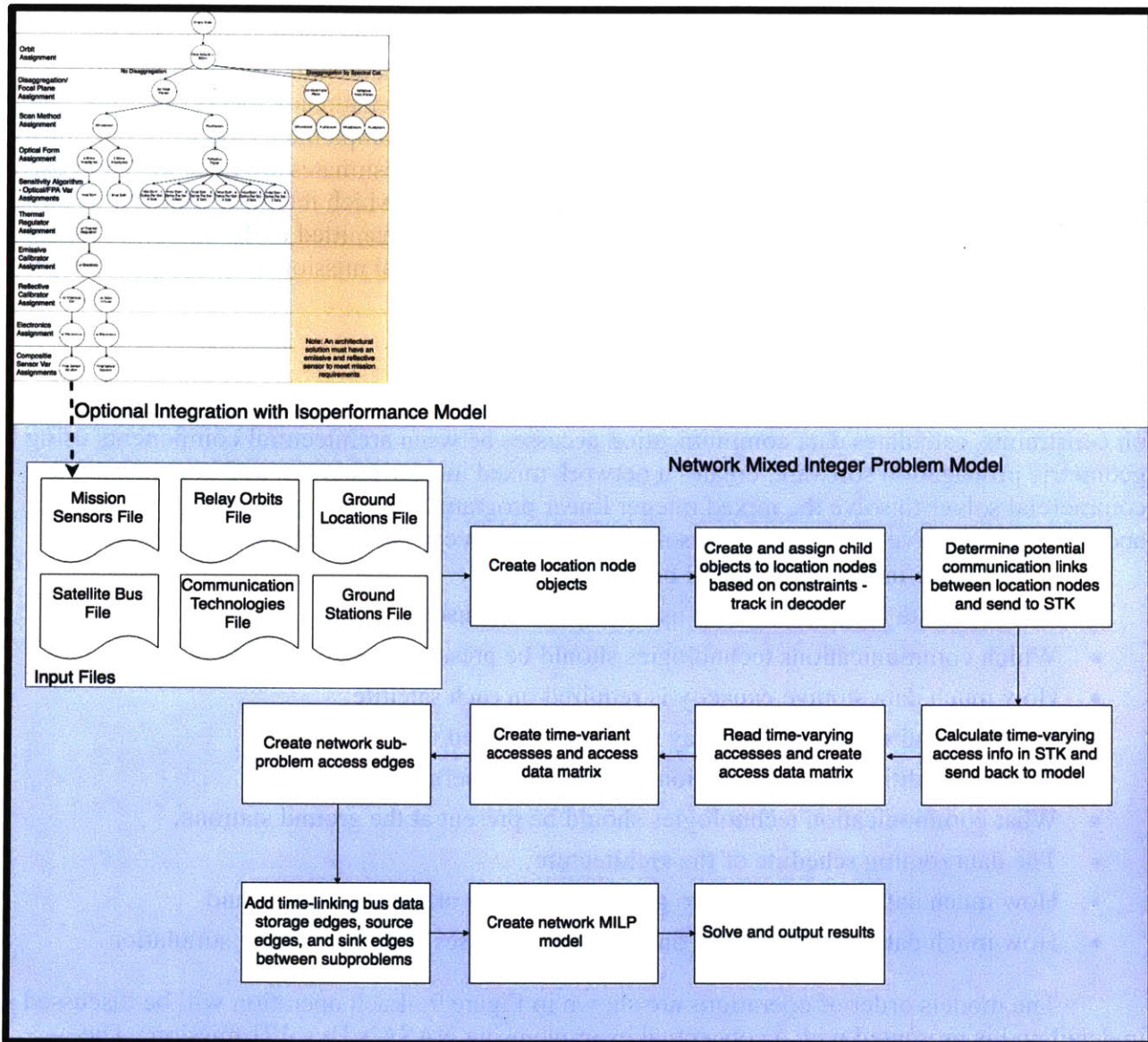


Figure 9: Satellite network mixed integer program flow

9.2.1 Satellite Network Model Input Files

First, it is important to discuss the model’s input files, which are structured in JavaScript Object Notation (JSON) format. An example of this format is shown in Appendix H. The files describe the mission needs (sensor architecture) and other architecture decisions which may include existing network infrastructure or new options for network expansion:

The mission’s sensors. Mission sensors are defined by their intended operating orbit and sensor characteristics. For the purposes of this thesis sensor characteristics include: the name of the sensor, the average power required for operating the sensor, the mass of the sensor, the data rate of the sensor, and the cost of the sensor. The orbit is described by altitude, eccentricity, inclination, right angle of ascending node, argument of perigee, and mean anomaly. Additionally, the file contains architecture constraints such as whether

the sensors (if multiple) can be disaggregated onto different satellites. In this thesis a sensor object will contain the following information:

$$s = (Name_s, W_s, M_s, DR_s, Cost_s, \{Constraints\})$$

Where:

- W_s is the estimated orbit average power of sensor s
- M_s is the mass of sensor s
- DR_s is the data rate of sensor s
- $Cost_s$ is the cost of sensor s

Satellite relay orbit locations. This file contains orbits at which the use of a satellite data relay, such as NASA's tracking and data relay satellites (TDRS), should be evaluated. A relay orbit is composed of orbit elements at the start of simulation and constraints that allow for the user to specify if a satellite relay is already present.

Ground locations. This file contains geodetic coordinates for ground locations at which a ground station is already present or should be considered.

Satellite bus options. This file contains an of array satellite bus options that can either host the mission sensors at the specified mission orbit or host the data relay communications technology at a specified relay orbit. In this thesis, a bus object will be represented using the following notation:

$$b = (Name_b, MP_b, M_b, W_b, DS_b, Cost_b, LCost_b, \{constraints\})$$

Where:

- MP_b is the mass allowance for the payload on the bus b
- M_b is the mass of bus b
- W_b is the estimated orbit average power generated by bus b
- DS_b is the data storage on bus b
- $Cost_b$ is the cost of bus b
- $LCost_b$ is the estimated launch cost of bus b

Ground station options. This file contains ground stations that can be present at a ground location.

$$g = (Name_g, DS_g, DP_g, Cost_g, \{constraints\})$$

Where:

- DS_g is the data storage at ground station g

- $DP_g = x \in \{0,1\}$ represents whether there is data processing at ground station g
- $Cost_g$ is the cost of building ground station g

Note: DS_g is included for expansion purposes where a ground station may be remotely located, is portable, and doesn't have a dedicated line to the data processing facility. All ground stations in this thesis are assumed to have a direct line to the data processing capability, which is indicated in a binary fashion.

Communication options. This file contains the communication technologies which can be present at orbit location nodes as part of a satellite bus, or at ground location nodes as part of a ground station. Each communication option is defined by: its name, the communications technology type, its transmit/receive configuration, its "specified" data rate, the half angle field of view in which it can establish a link with another communications terminal, the orientation of the option when placed on a ground location or orbit location, the orbit average power consumed by the technology, the mass of the technology, and the cost of the technology. Furthermore, constraints can be specified which limit a communication option's placement to being collocated with a particular satellite bus or ground station. A communications technology object is represented as

$$c = (T_c, X_c, DR_c, FOV_c, O_c, W_c, M_c, Cost_c, \{Constraints\}).$$

Where:

- T_c is the technology being used; ex. $T_c = x \in \{XBand, FSO, etc\}$
- $X_c = x \in \{Tx, Rx, TR\}$ is a variable that can be set to transmit, receive, or both transmit and receive. For the purposes of this paper we will omit this information and assume every communication system is bi-directional.
- DR_c is the data rate achievable by the technology
- FOV_c is the conical half angle field of view of the node with respect to the communication technologies orientation.
- $O_c \in (\theta, \phi)$ is the orientation of the communications system with respect to the host location node. In the case of a satellite, orientation will be referred to its directional along track movement. In the case of a terrestrial node, orientation will be defined from the longitudinal axis running north.
- W_c is the estimated power consumption of communication subsystem c
- M_c is the mass of communication subsystem c
- $Cost_c$ is the cost of communication subsystem c .

Note: Values such as data rate, can be assumed constant across all links for model simplicity at the cost of model fidelity; or can be dynamically computed for all

links once accesses between communication technologies assigned to nodes has been calculated.

9.3 Reading File Inputs and Creating Potential Child-Object/Location Node Assignments

The first two operations of the satellite network MILP program are reading the input files and creating location nodes and child objects. Location nodes are categorized as sensor, relay, or ground nodes and correspond to sensor or relay orbits and geodetic locations respectively. Child objects constitute mission sensors, satellite busses, ground stations, and communication technologies. Logic and constraints stated in the input files are used to determine which child objects can be assigned to each location node. In general, assignment constraints work in three ways. First, a child object type can be constrained to only be assigned to a specific location node type. For instance, mission sensor objects can only be assigned to sensor orbit location nodes. Second, a constraint can be imposed such that a location node “must have” a specific child object. In this case the location node is assigned that child object and no other child object of the same type (excluding mission sensor child objects). As an example, a relay node that “must have” a specific satellite bus represents a scenario where the satellite bus already exists in that orbit. Another satellite bus cannot be assigned to that location. Third, constraints can specify dependencies or mutual exclusions between child nodes. These constraints may represent physical limitations, use case rules, or reality. For instance, a communication technology can be “limited to” a specific satellite bus and mutually exclusive to the assignment of another communications technology. This constraint means that the communication technology can only be assigned to a location node if the requisite satellite bus is also assigned to that location node and if the other (mutually exclusive) communications technology is not. Another inter-child assignment constraint is between assignment of sensor child objects and assignment of a satellite bus object to a location. A sensor object can only be assigned to a location if there is also a bus assigned which provides a physical platform, can host the sensors mass, and can provide enough power.

All feasible assignment options are input into a “decoder” data frame with the columns specified in Table 13. The data frame provides a compact way of giving a unique “network node” index to each potential child object to location node object assignment. It also contains information such as how much power is supplied or required by the assignment of a child object. The “network node” index refers to the idea that we can translate each data frame entry into a mathematical decision variable which represents the decision of whether to assign a child object to a location node. These assignments, represented mathematically by Equation (1), constitute a set the potential nodes, or vertices, in a data network as will be shown with the HypsIRI example.

$$\forall v \in V: X_{Nv} = x \in \{0,1\} \quad (1)$$

Where:

V is the index set of all network nodes, or vertices, in the problem and is made up of the following subsets:

$B \subset V$ represents the subset of satellite bus child objects

$C \subset V$ represents the subset communication technologies child objects
 $S \subset V$ represents the subset of mission sensors child objects
 $G \subset V$ represents the subset of ground stations child objects

Data Frame Column	Description
Object	The child object
Object Type	The type of child object {'sensor', 'comm', 'bus', 'gs'}
Location Node	The location node the child can be assigned to
Child Index	The index of the child in the input set
Network Node	The corresponding network node index
Data Capacity	Data capacity of the child. Automatically zero for sensor and communication objects
Data Generated	Data generated by the network node. Zero for non-mission sensor objects
Power	Positive Number for Power Supply, Negative for Power Draw
Mass	Positive for Host Capacity, Negative for Passenger Mass
Cost	The cost of assigning the child to the location node
Fixed	Indicates whether the Child Object must be present at the location node.
Dependencies	Indicates constraint dependencies. This node can only be present in the network if it's dependencies are.
Mutual Exclusions	Indicates mutual exclusions of the child object. If a mutual exclusions is assigned the location node, this child object cannot be assigned

Table 13: Decoder data frame description.

9.3.1 HypsIRI Example Background

NASA states the planned capabilities of HypsIRI on their website:

The Hyperspectral Infrared Imager or HypsIRI mission will study the world's ecosystems and provide critical information on natural disasters such as volcanoes, wildfires and drought. HypsIRI will be able to identify the type of vegetation that is present and whether the vegetation is healthy. The mission will provide a benchmark on the state of the worlds ecosystems against which future changes can be assessed. The mission will also assess the pre-eruptive behavior of volcanoes and the likelihood of future eruptions as well as the carbon and other gases released from wildfires. [34]

The sensor architecture for HypsIRI, presented at the 2017 HypsIRI workshop, is shown in Figure 10 and Figure 11. This thesis selectively uses some of this information while constructing a visual representation to explain how the model works.

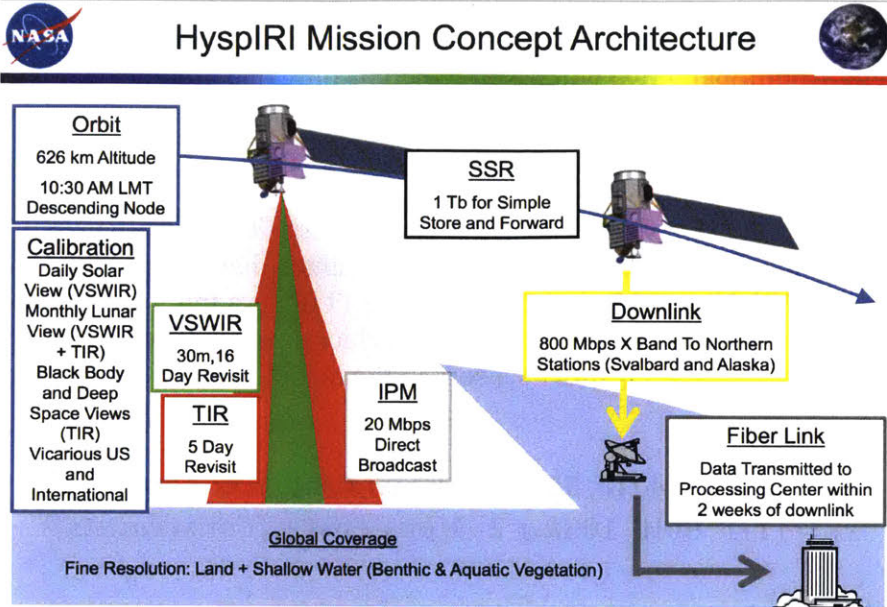


Figure 10: HyspIRI mission concept architecture. Source: [35]

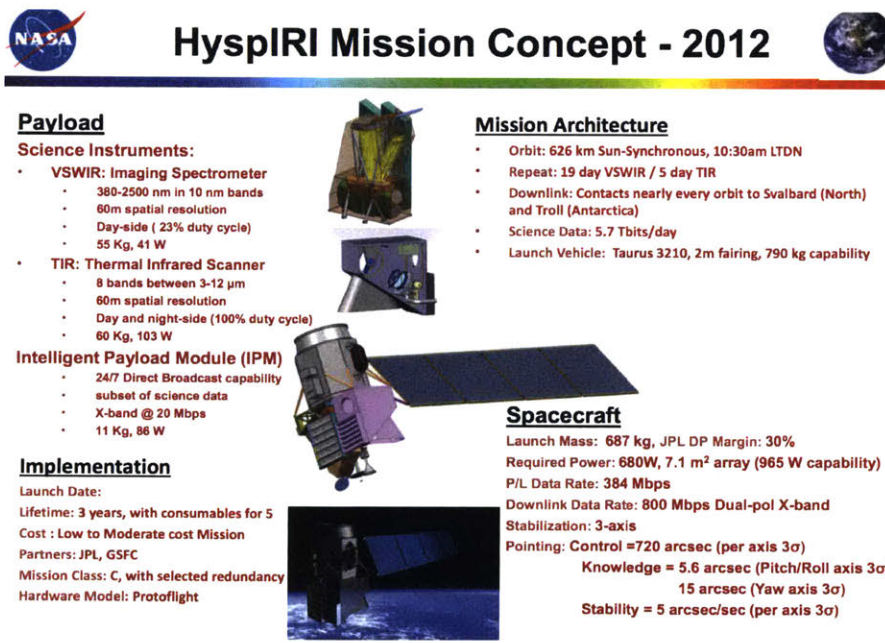


Figure 11: HyspIRI sensor architecture from 2012. Source: [35]

9.3.2 Location Nodes

It can be seen in the HyspIRI figures that sensors were designed for use in a particular orbit location which enables them to meet mission requirements. We will define this orbit without specific values as: $\mathbf{l}_0 = (e, a, i, \Omega, \omega, \theta, \{\text{Constraints}\})$. For the purposes of explaining features of the model, we will move away from official information and assume that a disaggregated system is feasible and the sensors can be operated in formation flight along the same orbit track. This assumption entails that there is sufficient capability for one of the sensors to be flown at an orbit, $\mathbf{l}_1 = (e, a, i, \Omega, \omega, \theta_\delta, \{\text{Constraints}\})$, which differs from \mathbf{l}_0 by a

small amount in true anomaly. This assumption also requires processing infrastructure for co-locating pixels etc. Additionally, we will assume there is only one ground station location: $l_2 = (\text{Lat}, \text{Lon}, \text{Alt}, \{\text{Constraints}\})$. The total set of location nodes is represented by L .

9.3.3 HypsIRI Mission Sensors

As is seen in Figure 11, the HypsIRI mission has two sensors. The HypsIRI VSWIR is a hyperspectral imager in the visible/near infrared band planned that will produce 300 Mbit/s when operational. The HypsIRI Thermal Infrared Radiometer (TIR) is a multispectral passive optical sensor operating over eight bands between medium and long wave infrared. It will produce approximately 24 Mbit/s. These HypsIRI sensors in the model's object notation are represented as:

$$s_0 = (\text{VSWIR}, 41W, 55k, 300Mbps, \text{Cost}_{s_0}, \{\text{Constraints}\})$$

$$s_1 = (\text{TIR}, 60W, 103kg, 24Mbps, \text{Cost}_{s_1}, \{\text{Constraints}\})$$

9.3.4 Satellite Busses

The busses we will consider in this explanatory scenario are the Surrey Satellite Technology US LLC (SSTL) -150ESPA and -300 platforms. Specifications as listed on the NASA Rapid Space Development Office (RSDO), are shown in Figure 12. They are represented in object notation as:

$$b_0 = (\text{SSTL150}, 150kg, 65kg, 85W, 16000Mb, 16500K, 5950K, \{\text{constraints}\})$$

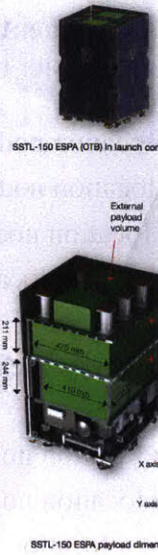
$$b_1 = (\text{SSTL300}, 218kg, 150kg, 140W, 16000Mb, 23500K, 7950K, \{\text{Constraints}\})$$

SSTL-150 ESPA

SSTL-300

SSTL-150 ESPA Baseline Specification

Indicative Mission Compatibility	
Orbit Average Payload Power	120 W / 85 W (peak/OAP) EOL
Maximum Payload Mass	85 kg
Bus Dry Mass	115 kg
Payload Data Downlink	2 Mbps, S-band
Payload Data Storage	16 Qb capacity, dual-redundant mass memory
Pointing Knowledge	0.8 deg (roll, yaw), 0.8 deg (pitch)
Pointing Control	2 deg all 3 axes
Pointing Stability (Jitter)	1.5 arcsec/sec
Slawrate	1 deg/sec
Position Knowledge	10 m
Mission Design Life	7 years
Compatible Launch Vehicles	ESPA rideshare on EELV vehicles and SpaceX vehicles
Types of Orbits Available	LEO 400 km to 1000 km, any inclination
Payload Volume (Green Volume)	476 x 647 x 244 mm (lower payload bay) 476 x 608 x 211 mm (upper payload bay) Additional accommodation available on Earth facing Facet up to 130 mm in Z direction with coordination
Bus Description	
Attitude Control System	3-axis control with reaction wheels and magnetorquers
Batteries	Li-Ion cells providing 15 Ah capacity
Solar Arrays	Triple junction GaAs cells
Main Bus Voltage Range	28-33 V range
C&DH Bus Architecture	Dual-redundant controller area network (CAN) bus
Communication Up/Downlink Band	S-band uplink/S-band downlink
Structure	Aluminum and aluminum-skinned honeycomb panels
Thermal Control	Primarily passive, plus limited use of heaters
Heritage and Programmatic Information	
Heritage Missions	OTB (2017), CFESat
Nominal Schedule from Order	24 months to payload integration, 31 months to launch

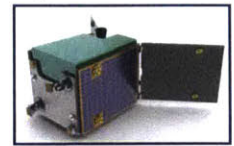


Baseline Performance Specification

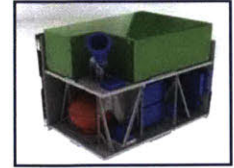
The SSTL-300 platform provides a capable and flexible baseline solution with the potential to enhance platform performance to meet customer-specific requirements.

Mission Compatibility	
Orbit Average Payload Power	140W (180W peak) EOL
Maximum Payload Mass	150kg
Bus Dry Mass	218 kg without payload
Science Data Downlink	105 Mbps, X-Band
Science Data Storage	16 Gbytes capacity, dual-redundant mass memory
Pointing Knowledge	72 arcsec (1 sigma) all 3 axes
Pointing Control	360 arcsec (1 sigma) all 3 axes
Pointing Stability (Jitter)	2 arcsec/sec
Slawrate	0.75 deg/sec
Position Knowledge	10m
Mission Design Life	7 years, Ps = 92%
Compatible Launch Vehicles	Falcon 1e, Atlas, Delta, Athena and other launchers
Types of Orbits Available	LEO 400km to 2000km, any inclination
External Payload Volume	730mm x 455mm x 1000mm
Internal Payload Volume	279.5mm x 231.5mm x 252.5mm
Bus Description	
Attitude Control System	3-axis control with reaction wheels and magnetorquers
Batteries	Li-Ion cells providing 15 Ah capacity
Solar Arrays	Triple-junction GaAs cells, total area 2.44m ²
Main Bus Voltage Range	28V-33V range
C&DH Bus Architecture	Dual-redundant Controller Area Network (CAN) bus
Communication Up/Downlink Band	S-Band uplink/S-Band downlink
Structure	Aluminum and aluminum-skinned honeycomb panels
Propulsion	Hot gas Xenon resistojet
Delta V	15m/s
Thermal Control	Primarily passive, plus limited use of heaters
Heritage & Programmatic Information	
Heritage missions	RapidEye, DMC + 4, TopSat, NigeriaSat-2
Nominal schedule from Order	31 months to payload integration, 37 months to launch

configuration



SSTL-300 in flight configuration



SSTL-300 baseline configuration



Surrey Mission Control

Figure 12: SSTL-150 ESPA [36] and SSTL-300 Information [37]

9.3.5 Ground Station

This example will include a default ground station which has no data storage but has a dedicated line to a data processing facility. The cost is zero since the ground station is already fielded.

$$g_0 = (\text{Default}, 0\text{Mbit}, 1, 0\text{K}, \{\text{constraints}\})$$

9.3.6 Communications Technology

To account for the communication capability already present on the SSTL platforms we create two communication technology data structures. Note that power, mass, and subsystem cost are set to zero because it is assumed that these quantities are accounted for within the satellite bus's specification.

$$c_0 = (\text{ESPA150_Comm}, \text{SBand}, \text{TR}, 2\text{Mbps}, 90, [0, 90], 0\text{W}, 0\text{kg}, 0\text{K}, \{\text{Constraints}\})$$

$$c_1 = (\text{ESPA300_Comm}, \text{XBand}, \text{TR}, 105\text{Mbps}, 90, [0, 90], 0\text{W}, 0\text{kg}, 0\text{K}, \{\text{Constraints}\})$$

For illustrative purposes, it will be supposed that additional communications technologies can be integrated onto a bus for a fixed cost. One technology is a hypothetical Free Space Optical (FSO) communication terminal that enables inter-satellite links between two satellites at 300Mbit/s. This requires the definition of two different communication objects, c_2 and c_3 , which only differ in orientation. Orientations are defined such that the communication terminals face each other while in orbit if c_2 is assigned to location node l_1 and c_3 is assigned to location node l_1 . Additionally, we will define an X-band upgrade option for the SSTL-300 that is capable of a 400Mbit/s downlink and available for a cost of \$700K. Finally, we define the communications capability of the ground segment as c_5 .

$$c_2 = (\text{FSO_Config1}, \text{FSO}, \text{TR}, 300\text{Mbps}, 5, [0, 0], 5\text{W}, 5\text{kg}, 200\text{K}, \{\text{Constraints}\})$$

$$c_3 = (\text{FSO_Config2}, \text{FSO}, \text{TR}, 300\text{Mbps}, 5, [180, 0], 5\text{W}, 5\text{kg}, 200\text{K}, \{\text{Constraints}\})$$

$$c_4 = (\text{XBand_Upgrade}, \text{TR}, 400\text{Mbps}, 90, [0, 90], 20\text{W}, 10\text{kg}, 700\text{K}, \{\text{Constraints}\})$$

$$c_5 = (\text{XBand_Ground}, \text{TR}, 800\text{Mbps}, 90, [0, 90], 0\text{W}, 0\text{kg}, 0\text{K}, \{\text{Constraints}\})$$

9.3.7 Constraints and Representation of the Assignment Data Frame

We will artificially constrain the problem such that it is easier to graphically represent and explain. Constraints include:

- b_0 , the SSTL-150, can only be assigned to location node l_0
- b_1 , the SSTL-300, can only be assigned to location node l_1
- g_1 , the ground station, must be assigned to location node l_2
- c_0 , the SSTL-150 comm, can only be assigned to a location node that also has b_0 , the SSTL-150 bus, assigned
- c_1 , the SSTL-300 comm, can only be assigned to a location node that has b_1 , the SSTL-300 bus, assigned
- c_2 , FSO config 1, can only be assigned to a location node that also has b_0 assigned
- c_3 , FSO config 2, can only be assigned to a location node that also has b_1 assigned
- c_4 , X-band upgrade, can only be assigned to a location node that also has b_1 assigned
- c_5 , ground X-band, must be assigned to a ground node l_2

The model's input files parser creates the decoder shown in Appendix H with each row in the decoder corresponding with a decision variable, $X_{Ni} = x \in \{0,1\}$ unless a value of "1" is in the fixed column which represents a constraint that the child object **must** be assigned to the location node. Per the HypsIRI example constraints, the ground station g_1 and the comm object c_5 must be present at the ground location l_3 ; thus $X_{N6} = X_{N5} = 1$. The visual representation of the resulting decision network is presented in Figure 13. The green dashed double arrows between communication objects represent potential communication links because they share a common communications technology and both child objects can transmit/receive.

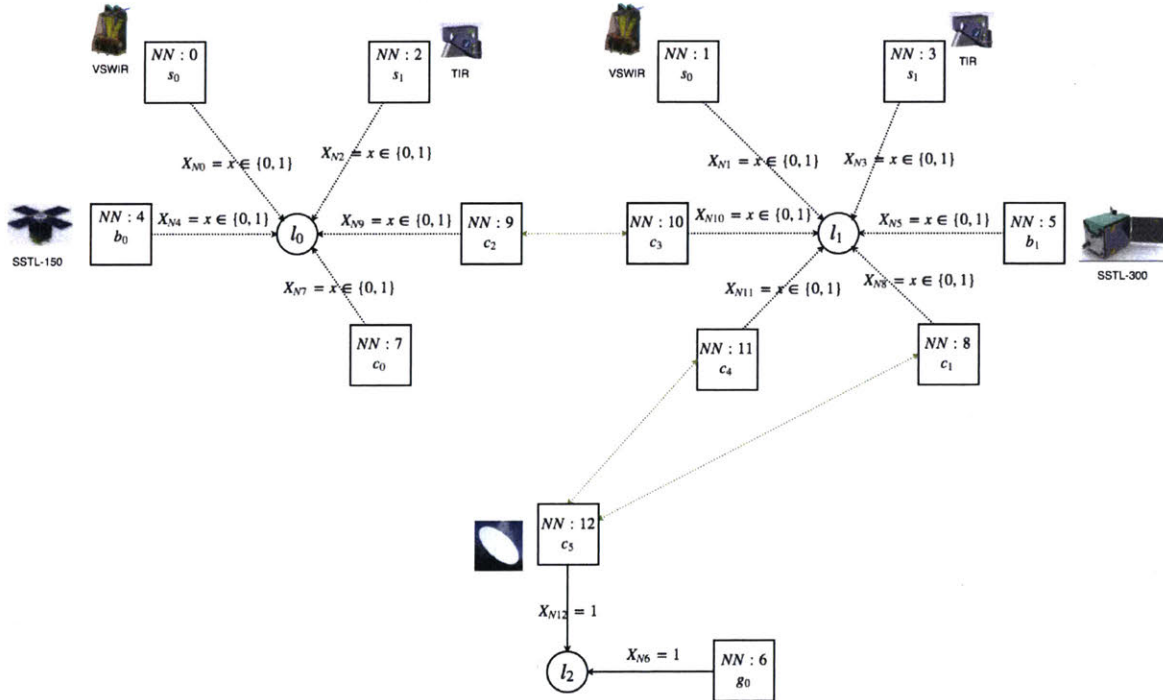


Figure 13: HypsIRI example network node representation.

9.4 Creating Accesses Between Network Nodes

To build the network problem it is necessary to determine when and how data can be transferred between any two nodes. For this purpose, an access is defined in this work as the time period in which data can be transferred between two network nodes (bus data storage, communications subsystems, sensor, sink, and source). The model initially constructs time-varying and time-invariant accesses which are both stored as a tuple:

$$\mathbf{a} = (v_i, v_j, t_{kstart}, t_{kend}, \bar{t}_k, \bar{r})$$

Where:

- v_i is the network vertex sending data, known as the predecessor vertex
- v_j is the network vertex receiving data, known as the successor vertex
- t_{kstart} is the start of the access
- t_{kend} is the end of the access
- \bar{t}_k is an array of sample times during the access which correspond with simulation time steps
- \bar{r} is an array of data rate values corresponding to \bar{t}_k

9.4.1 Calculating Time-Varying Accesses

The ability of the architecture to transfer data to the user is a function of the link accesses between the communication systems at space and ground locations. The data rate of a specific link is a function of the technologies providing the link as well as the geometric properties of azimuth, elevation, and range between the two locations. Because these accesses vary with time, we will refer to them as time-varying accesses in the context of this work.

A geometric propagation model, in this case STK, is used to determine time-varying accesses between communication technologies assigned to two different location nodes with link potential. Link potential is determined by a function that parses through all location nodes and determines whether communication network nodes at two different locations have the same technology (e.g. X-band) and satisfy transmit/receive constraints (e.g. v_i can transmit and v_j can receive). Potential links are then passed to a function which automates STK to compute accesses for all potential links (l_i, c_u, l_j, c_v) over the simulation time interval (T_{start}, T_{end}). The number of accesses between any two location nodes with compatible communication technology assignments depends on the time interval used in the propagation model along with the location node's initial conditions.

Computing accesses results in a set of time-varying accesses, A_V , which can be mapped to a data rate versus time matrix, D_V . D_V is an $m \times T$ matrix, where m is the number of accesses in A_V and T is the number of time steps between simulation start and stop. Thus, $d_{kl} \in D_V$ corresponds to the maximum data rate of access $a_k \in A_V$ at time $T_{start} + \delta l$ where δ is the time step. Data volume of the time-varying access over the time step is simply δd_{kl} .

9.4.2 Time-Invariant Accesses

Child objects assigned to the same location node form a time-invariant network. That is, a child object at a location node will always have the ability to send data to a subset of other children co-located at the same location node via the host system. For example, a sensor on a satellite bus

will always have a unidirectional access to the satellite bus's data storage. The maximum data transferred per second between co-located child objects using time-invariant accesses is limited by the following constraints:

- The maximum data rate between communication child objects at the same location node is the minimum data rate between them.
- The maximum data rate from a sensor to any co-located child object is the minimum data rate between the two.
- The maximum data rate between any non-bus child object and the bus child object is the maximum data rate of the non-bus object.

Given a set of time-invariant accesses between network nodes, A_I , we can create a data rate versus time matrix, D_I . The key difference between D_V and D_I is that the values in D_I are invariant with respect to time. That is the data rates are constant throughout the simulation.

$$\forall k \in m, \forall l \in T: d_{kl} \in D_I = d_{k(l+1)} \in D_I = \text{constant}$$

9.4.3 Combining Accesses

Accesses are combined into a single set, $A_C = A_I \cup A_V$, and the data generation matrices are concatenated such that each row $k \in m$ of D_C , represents data generation over time for the corresponding access $a_k \in A_C$. Additionally, a set of access start times and end times is created and chronologically ordered from T_{start} through T_{end} . This set is referred to as "Times of Interest", TOI , and represents points in time in which there is an overall change in the network problem, as will be discussed in the next section.

9.4.4 HypsIRI Example: Accesses

For the HypsIRI example, Figure 14 illustrates time-invariant accesses as black lines and time-varying accesses as red lines. In this work, a major simplification is treating sensor data generation as time-invariant. Sensor data generation can be changed to time-varying given logic for determining when the sensor generates data as a function of orbit. For instance, a VSWIR sensor will not generate data when it is in night since there is minimal light reflected from the Earth at this time. This, along with other modifications which increase the fidelity of the model, will be discussed in the recommendations for future work section.

Hypothetical access data for a 200 second simulation corresponding to Figure 14 is shown in Appendix H. Looking at the time-varying accesses it can be seen that the access between the two FSO terminals lasts the entire simulation. However, downlink accesses v_{11} to v_{12} and v_8 to v_{12} are only available from $t=100$ through $t=200$ seconds.

NN : Object

- 0 : s_0
- 1 : s_0
- 2 : s_1
- 3 : s_1
- 4 : b_0
- 5 : b_1
- 6 : g_0
- 7 : c_0
- 8 : c_1
- 9 : c_2
- 10 : c_3
- 11 : c_4
- 12 : c_5

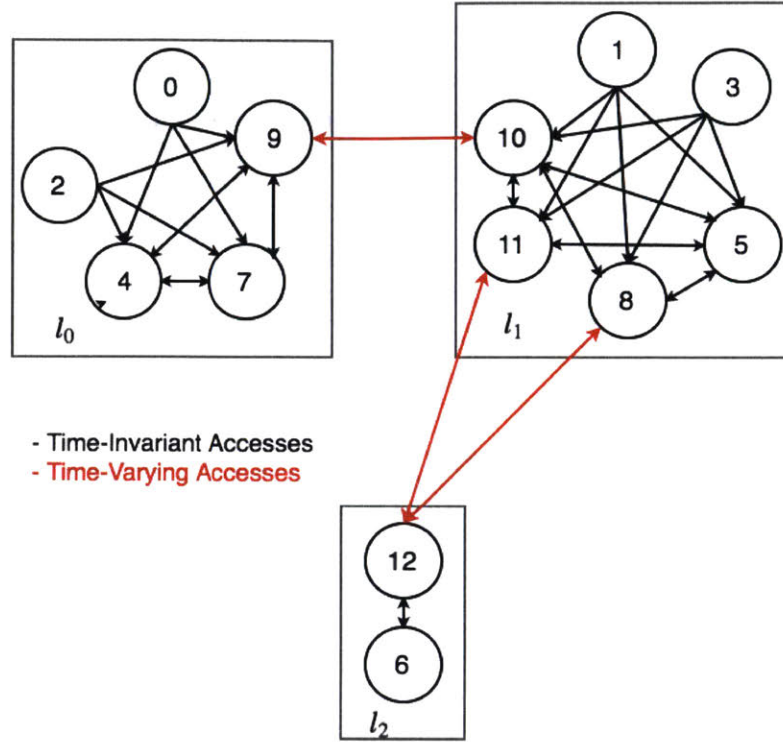


Figure 14: Time-varying and time-invariant accesses.

9.4.5 Creating a Network Problem via Network Sub-Problems

A network sub-problem, in the context of this thesis, refers to a time period in which the accesses in the network problem remain static. In other words, there is no start or finish of a time-variant access within the sub-problem. Instead the start or finish of an access results in the creation of a new sub-problem. Given a sorted array of access start and stop times, TOI , it can be inferred that the amount of sub-problems will be equal to the length of TOI minus one with each sub-problem representing a time period, $TOI[i]$ through $TOI[i + 1]$, of the larger simulation. P is the set of the sub-problem indexes with $t_{ps} \forall p \in P$ representing the start of a sub-problem, and $t_{pe} \forall p \in P$ representing the end of a sub-problem.

Referring to Table 44 of Appendix H, the times of interest in the HypsIRI example are $TOI = \{0,100,200\}$. Thus, there are two sub-problems: the first representing time $t = 0$ through $t = 100$ and the second representing $t = 100$ through $t = 200$.

Access pairs which have non-zero values for data rate during these sub-problems are mapped to edges between network nodes for each sub-problem with each edge defined as $e = (p, v_k, p, v_l) \in E_p \forall p \in P$. Thus, each sub-problem has its own set of internal edges. In addition to these edges, there are three additional edge categories which operate between sub-

problems and must be added. These edges allow for the sub-problems to be aggregated into one network problem that spans over the entire simulation. They are:

Inter-problem bus data storage (or time-linking) edges. There is an edge that connects the data storage on a bus in one sub-problem to the data storage on the bus in the subsequent problem. The data transferred via this edge represents the data stored on the bus at the time of sub-problem transition. These edges are represented as

$$e = (p_i, v_k \in B, p_{i+1}, v_k \in B) \in E_B .$$

Source to sensor edges. There is an edge from a “source” node to all sensor nodes across sub-problems with data volume equal to the sensor data generation during the sub-problem’s period. $e = (0, source, p_i \in P, v_l \in S) \in E_{source}$. Note that the source is placed in sub-problem zero for convenience.

Ground station to sink edges. There is an edge from all ground stations to a “sink” node. $e = (p_i \in P, v_i \in V, 0, sink) \in E_{sink}$. Note that the sink is placed in sub-problem for convenience.

During construction of each edge $e \in E_p$, the model calculates a corresponding maximum data volume $d_e \in D_{vol}$, using the data matrix D_C and the time period of the sub-problem. These maximum data volume constraints are necessary for imposing maximum flow constraints on each edge. For each source to sensor edge, $e \in E_{source}$, the model calculates the data volume, $d_e \in D_{vol}$, that **must** be transmitted due to data generation of the sensor over the sub-problem time period.

A visual representation of the network problem constructed from the HyspIRI access data in Appendix H is shown in Figure 15. The black arcs represent time-invariant edges, the red arcs represent time-variant edges, the blue arcs represent the inter-problem bus data storage edges, and the green arcs represent the source and sink edges. It can be seen how the black and red arcs do not cross sub-problem boundaries. Example notation is shown for each type of edge along with the maximum data it can transmit or, in the case source to sensor edges, the data it must transmit. For inter-problem data storage edges (or time-linking edges) and sink edges there are no flow constraints other than it must be a positive real number.

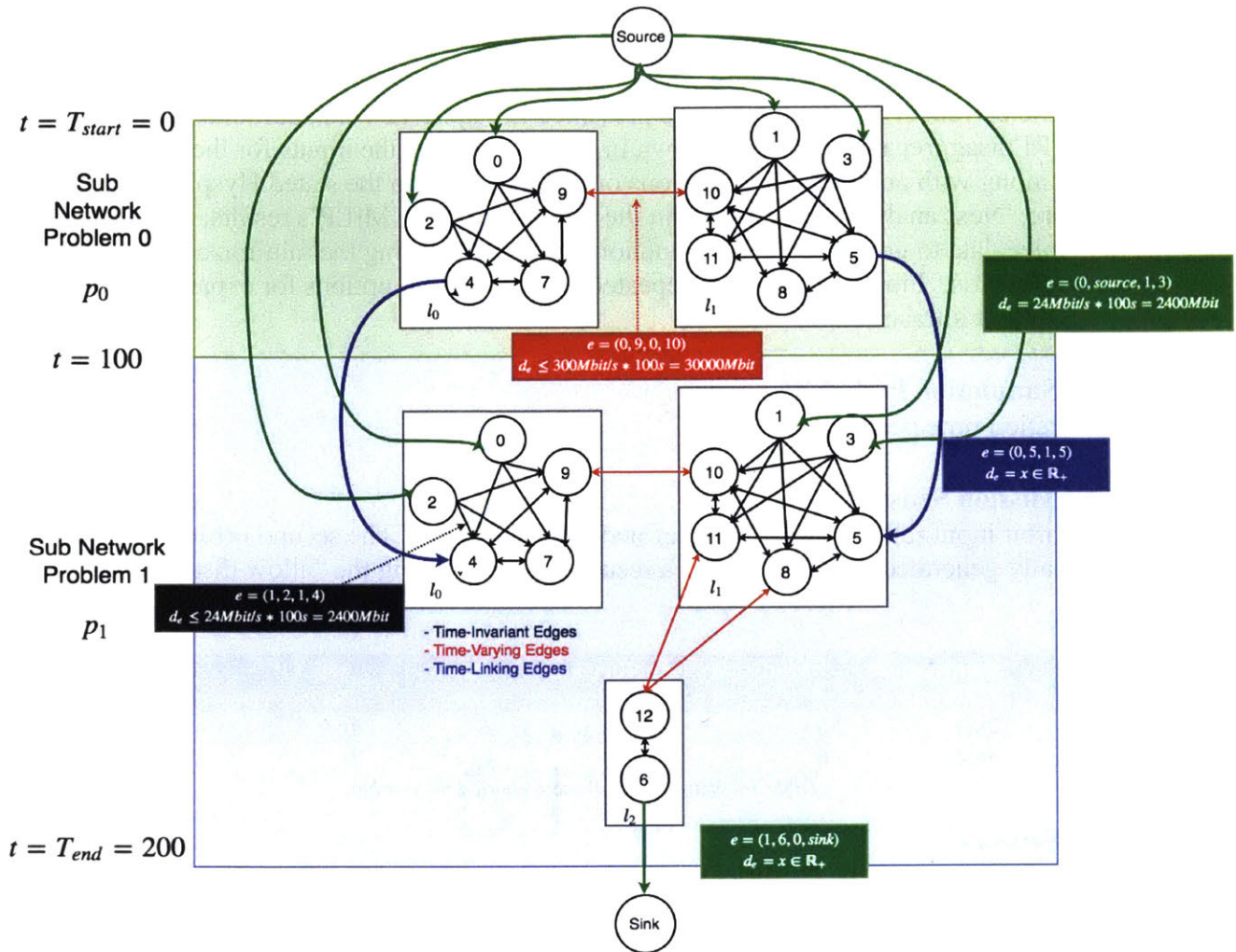


Figure 15: HysplRI example network problem.

9.5 Generalizing the Network Problem Notation and Solving It

In general, the network graph constructed from user inputs can be described in terms of the sub-problems, location nodes, network nodes, and edges: $G = (P, L, V, E)$. Once the graph is populated, the model imposes constraints and defines the user selected objective function, resulting in a network maximum flow problem. The model then integrates with a solver, in this case Gurobi, to optimize the objective function of the network problem by assigning values to the various decision variables. Objective functions examined in this thesis include (1) maximizing the data transferred to ground during the simulation and (2) minimizing the cost of the architectures.

Once solved, the model outputs the values of the decision variables that result in the optimal solution. Thus, the user is informed of the architecture's optimal composition based on the values of the assignment decision variables. By using the maximize data to ground objective, the user is also informed of the optimal schedule for transmitting data to ground throughout the simulation via the data transfer edge decision variables. The network problem's general form is in Appendix I.

9.6 Use and Results

A scenario is presented in this section which demonstrates the use of the model. The scenario is based on the current HypsIRI architecture presented in Figure 11 with additional options from the HypsIRI disaggregation example shown in Figure 13. First the inputs for the model are described along with any major assumptions or differences from the stated HypsIRI mission architecture. Next analysis is presented on the satellite network MILP’s results; first from using the maximize data to ground objective function, followed by using the minimize architecture cost objection function. Finally, analysis is repeated with additional options for expanding the ground segment infrastructure.

9.6.1 Simulation Period

The simulation time period is 24 hours.

9.6.2 Mission Sensor Orbit Inputs

The first orbit input reflects HypsIRI’s defined operating orbit. The second orbit listed is automatically generated by the model as a result of implementing the “allow disaggregation” option.

Location Node	Semi-major axis (km)	Eccentricity	Inclination (Deg.)	RAAN (Deg.)	Arg. Of Perigee (Deg.)	Mean Anomaly (Deg.)
l_0	7004	0	98	0	0	0
l_1	7004	0	98	0	0	.1

Table 14: Network MILP use scenario sensor orbits.

9.6.3 Sensors

The sensor inputs reflect the HypsIRI architecture information with three major assumptions. First, the data generation from the sensors is considered time-invariant, so average data rates are used. For the VSWIR sensor, a duty cycle of 50% is assumed resulting in average data generation of 150Mbit/s. This assumption accounts for the VSWIR sensor only taking images during day. The TIR duty cycle is assumed to be 100% since emissive bands are collected throughout orbit. The second major assumption is that the power draws for both sensors are constant. The final assumption is that the cost of each sensor is \$50M (since costs are not published).

Child Object	Name	Power Required (W)	Mass (kg)	Data Rate (Mbit/s)	Cost (K)
s_0	HypsIRI VSWIR	41	55	150	50000
s_1	HypsIRI TIR	60	103	24	50000

Table 15: Network MILP use scenario sensors.

9.6.4 Ground Location

Per Figure 10, the HypsIRI’s mission architecture uses NASA’s Fairbanks and Svalbard ground station locations. It is assumed these locations already have ground stations and communications relays installed, as represented by the “must-have” constraints.

Location Node	Location Name	Lat	Long	Alt	Fixed – Must Have
l_2	Svalbard	78.229772	15.407786	0	g_0 , Standard GS c_6 , X-Band Ground 1 c_7 , X-Band Ground 2
l_3	Fairbanks	64.976666	212.48305	316	g_0 , Standard GS c_6 , X-Band Ground c_7 , X-Band Ground 2

Table 16: Network MILP use scenario ground station locations.

9.6.5 Satellite Busses

The HypsIRI mission is currently slotted to use the Orbital ATK’s A200 bus [38]. However, as there is no publicly available cost figure for the A200, the SSTL-600 bus is used since it has comparable mass support and power supply capabilities [39]. The SSTL-150 ESPA and SSTL-300 are also included in the input files. Launch costs for each bus are estimated using data from Spaceflight Industries [40].

Child Object	Name	Dry Mass	Max Payload Mass	Orbit Average Power	Data Storage	Cost	Launch Cost
b_0	SSTL-150 ESPA	115	65	85	16	16500	5950
b_1	SSTL-300	218	150	140	16	23500	7950
b_2	SSTL-600	429	200	386	128	36000	20000

Table 17: Network MILP use scenario bus inputs.

9.6.6 Ground Station

The ground station facility in this scenario is generic without constraints. There is no cost associated with this ground station because it is assumed this type of ground station has already been fielded at both Svalbard and Fairbanks.

Child Object	Name	Data Storage	Data Processing	Cost	Constraints
g_0	Standard GS	0	Yes	0	None

Table 18: Network MILP use scenario ground station input.

9.6.7 Communication Technologies

Communications technologies include each bus's “off-the-shelf” or “stock” capability, free-space optical inter-satellite link options, satellite X-Band downlink upgrade options, and the X-Band ground station capabilities. The 400Mbit X-band transceiver data comes from a published Surrey Satellite Technology US brochure on the XTx-400 X-Band Transmitter [41] and a U.S. General Services Administration price agreement valid through March 2017 [42]. Data for existing free space optical inter-satellite link subsystems is not available. Instead, the data used is completely hypothetical. The ground station’s X-Band capability includes two 400Mbit X-band links so that the station can communicate with multiple downlinks at any given time.

Child Object	Name	Comm. Tech.	Data Rate (Gbit/s)	Trans/Receive	Field of View	Orien.	Orbit Avg. Power (W)	Mass (kg)	Cost (K)	Limited To	Mutual Exclusions
c_0	SSTL-150 Comm	S-Band	0.002	T	90	[0,90]	0	0	0	b_0 , SSTL-150 ESPA	c_5
c_1	SSTL-300 Comm	X-Band	.105	T	90	[0,90]	0	0	0	b_1 , SSTL-300	c_5
c_2	SSTL-600 Comm	X-Band	.105	T	90	[0,90]	0	0	0	b_2 , SSTL-600	c_5
c_3	FSO Config-1	FSO	.3	TR	5	[0,0]	5	2	1000	any-bus none-ground	c_4
c_4	FSO Config-2	FSO	.3	TR	5	[180,0]	5	2	1000	any-bus none-ground	c_3
c_5	X-Band Upgrade 1	X-Band	.4	T	90	[0,90]	20	4	700	Any-bus None-ground	
c_5	X-Band Upgrade 2	X-Band	.4	T	90	[0,90]	20	4	700	b_2 , SSTL-600	
c_6	X-Band Ground 1	X-Band	.8	TR	90	[0,90]	0	0	0	None-bus Any-ground	
c_7	X-Band Ground 1	X-Band	.4	TR	90	[0,90]	0	0	0	None-bus Any-ground	

Table 19: Network MILP use scenario communication technologies input.

9.6.8 Scenario Analysis

9.6.8.1 Minimizing Data Stored on Bus – Maximizing Data to Ground

The network model can be used in a variety of different modes and altered in each successive run, using different constraints or objective functions, to better understand architecture options. To start, this scenario is executed using the objective function of minimizing the maximum data stored on satellite busses without a constraint on bus data storage capacity. Minimizing the maximum data stored on satellite busses inherently minimizes the end bus data storage and thus maximizes the data throughput to ground for the entire simulation. This allows the user to develop an idea of the maximum throughput possible if cost is not an issue.

Figure 16 is a screenshot of the solver results. It is notable that the model was able to solve a network problem with 9259 continuous decision variables and 30 binary integers in 1.29 seconds using Gurobi. The maximum amount of data stored on the busses at any given time during the simulation is 2.164 terabits (which is quite a lot). This indicates that the architecture will not have enough throughput even with unlimited budget; though, it is also important to remember the time-invariant assumptions for the sensor data generation and link data rates when making this assertion. The total amount data sent to ground is shown in Figure 17.

Model assignments are shown in Figure 18. The optimal solution has the two sensors disaggregated onto two SSTL-600 satellite busses with both having two 400Mbit/s X-Band transmitters to ground. Additionally, inter-satellite links are established between the two satellites using the FSO communication technology. This inter-satellite link allows the bus hosting the high data VSWIR instrument to send data to the other bus hosting the low-data TIR instrument and take advantage of two more high throughput X-band transmitters. The total cost of all of these assignments is \$216.8M as shown in Figure 19. Optimal link scheduling is also an output from this model and is shown in Figure 42 of Appendix J.

```

Optimize a model with 29946 rows, 9289 columns and 71331 nonzeros
Variable types: 9259 continuous, 30 integer (30 binary)
Coefficient statistics:
  Matrix range [1e-08, 1e+09]
  Objective range [1e+00, 1e+00]
  Bounds range [1e+00, 1e+00]
  RHS range [9e-02, 2e+03]
Warning: Model contains large matrix coefficient range
        Consider reformulating model or setting NumericFocus parameter
        to avoid numerical issues.
Presolve removed 18788 rows and 4070 columns
Presolve time: 0.24s
Presolved: 11158 rows, 5219 columns, 33218 nonzeros
Variable types: 5197 continuous, 22 integer (22 binary)
Found heuristic solution: objective 5228.5450000

Root relaxation: objective 2.164226e+03, 3678 iterations, 0.77 seconds

  Nodes | Current Node | Objective Bounds | Work
  Expl Unexpl | Obj Depth IntInf | Incumbent BestBd Gap | It/Node Time
*  0    0          0  2164.2260000 2164.22600 0.00% - 1s

Explored 0 nodes (4283 simplex iterations) in 1.29 seconds
Thread count was 4 (of 4 available processors)

Solution count 2: 2164.23 5228.54

Optimal solution found (tolerance 1.00e-04)
Best objective 2.164225999999e+03, best bound 2.164225999999e+03, gap 0.0000%

```

Figure 16: Base scenario network MILP solver readout.

```

The total data generated by sensors in this scenario is: 15033.426
The total data sent to ground is:12869.2
The maximum amount of data stored on SSTL-600 at location 0 is: 579.552
The end amount of data stored on SSTL-600 at location 0 is: 579.552
The maximum amount of data stored on SSTL-600 at location 1 is: 1584.674
The end amount of data stored on SSTL-600 at location 1 is: 1584.674

```

Figure 17: Base scenario network MILP data volume.

```

HYSPIRI VSWIR is assigned to location node 1
HYSPIRI TIR is assigned to location node 1
SSTL-600 is assigned to location node 0
SSTL-600 is assigned to location node 1
Standard GS is assigned to location node 2
Standard GS is assigned to location node 3
FSO Crosslink Config 1 is assigned to location node 0
X-Band Upgrade-1 is assigned to location node 0
X-Band Upgrade-2 is assigned to location node 0
FSO Crosslink Config 2 is assigned to location node 1
X-Band Upgrade-1 is assigned to location node 1
X-Band Upgrade-2 is assigned to location node 1
X-Band Ground 1 is assigned to location node 2
X-Band Ground 2 is assigned to location node 2
X-Band Ground 1 is assigned to location node 3
X-Band Ground 2 is assigned to location node 3

```

Figure 18: Base scenario network MILP object assignments.

```

The total cost of this architecture is: 216800.0

```

Figure 19: Base scenario network MILP architecture total cost.

9.6.8.2 Minimizing Cost

Now that some of the limiting factors of the model are understood, we can attempt to see what changes might occur if the objective function is changed to minimize cost. In addition to changing the objective function, a constraint will be reactivated which states the maximum bus storage is three terabits. Additionally, the cost of each megabit above the stock bus data storage capacity is \$10K. The objective function and additional constraint are shown below.

$$\min \sum_{v \in V} c_v X_{Nv} + \sum_{v \in B} 10K(Z_{Mv} - cap_v) \text{ s. t.}$$

$$\forall v \in B: Z_{Mv} \leq 3000Mbit$$

This time, the solver took 9.83 seconds to solve as seen in Figure 40 of Appendix J. The data throughput information is shown in Figure 20 and the assignments information in Figure 21. Looking at the assignments reveals a very interesting solution in which the sensors are disaggregated between a SSTL-150 with a single 400 Mbit downlink upgrade and a SSTL-300 with its stock option. This can be expected since the SSTL-150 is hosting the high data rate VSWIR sensor. Both satellites can communicate via the FSO inter-satellite link allowing them to share data and maximize data throughput to ground. While this architecture looks promising in terms of cost at 156.6M, it performs rather poorly compared to the results of NASA's published HypsIRI architecture, shown in Section 6 of Appendix J, of a single SSTL-600 satellite bus with both instruments and two 400 Mbit downlink X-Band transceivers. NASA's architecture for HypsIRI only costs an additional 0.8M (157.4M total) and results in two-thirds the overflow data at the end of the simulation as compared to this disaggregated structure (~3,800 Gbit vs. 6000 Gbit total between two satellite busses).

```
The total data generated by sensors in this scenario is: 15033.426
The total data sent to ground is:9033.425999996623
The maximum amount of data stored on SSTL-300 ESPA at location 0 is: 3000.0
The end amount of data stored on SSTL-300 ESPA at location 0 is: 3000.0
The maximum amount of data stored on SSTL-150 ESPA at location 1 is: 3000.0
The end amount of data stored on SSTL-150 ESPA at location 1 is: 3000.0
```

Figure 20: Base scenario network MILP, minimizing cost, data throughput.


```

HYSPIRI TIR is assigned to location node 0
HYSPIRI VSWIR is assigned to location node 1
SSTL-300 ESPA is assigned to location node 0
SSTL-150 ESPA is assigned to location node 1
Standard GS is assigned to location node 2
Standard GS is assigned to location node 3
SSTL-300 ESPA Comm is assigned to location node 0
FS0 Crosslink Config 1 is assigned to location node 0
FS0 Crosslink Config 2 is assigned to location node 1
X-Band Upgrade-1 is assigned to location node 1
X-Band Ground 1 is assigned to location node 2
X-Band Ground 2 is assigned to location node 2
X-Band Ground 1 is assigned to location node 3
X-Band Ground 2 is assigned to location node 3

```

Figure 21: Base scenario network MILP, minimizing cost, object assignments.

9.6.8.3 Changing Model Inputs – Adding Ground Stations

Building on the previous example, let's assume that more than one terabit of data storage on a satellite bus is infeasible. Thus, both NASA's published architecture and the previously discussed disaggregated architecture must discard mission data at the end of the scenario. If discarding data is unacceptable, then it is necessary to change inputs of the model such that more trades can be conducted and a solution found. In this case, a satellite relay could be introduced, payload satellites could be equipped with even higher data rate transmitters, or more ground stations could be added. Here, the third option will be explored; though the model can certainly handle adjusting the inputs to examine all three options.

To start, two more ground stations with no cost are introduced. These ground stations represent NASA's McMurdo and Troll ground stations; both in Antarctica. Additionally, four optional, "portable" ground station locations will be introduced at coordinates corresponding to NASA related locations in Santiago, Chile; Wallops Island, Virginia; near South Point, Hawaii; and Dongara, Western Australia. These sites can only be populated by a hypothetical portable X-Band ground station. The information for the locations, facility, and receiver can be found in Section 3 of Appendix J.

Again, the model is first run using the minimize maximum data on the bus objective function. The resulting solver information, data throughput of the system, assignments, and total cost are in Section 4 of Appendix J. The results show a marked improvement in the data throughput of the system, with only approximately 550 Gbit of data remaining across both of the two satellite busses at the end of the twenty-four-hour simulation. Additionally, it is apparent that a single well-placed ground station at one of the poles is much more effective than a ground station placed in the mid latitudes. This is expected given the system's sun-synchronous orbit. Since cost was not an issue, all ground station options were utilized resulting in a total cost of \$220.8M.

The model was run a second time using the minimize cost objective function with a constraint that the bus data storage on each satellite can be upgraded to a maximum of 512 Gbit. Like before, the cost of each additional Gbit of capacity is 10K. Screenshots of the program's results are in Section 5 of Appendix J. The solver took 34.95 seconds to find a solution to this problem and resulted in another disaggregated architecture. This time however, only one

satellite X-band upgrade was “purchased” for the SSTL-150 hosting the VSWIR sensor. The SSTL-300 hosting the TIR sensor is assigned stock communications technology with a downlink of 105Mbit/s. There is also no inter-satellite link in this result. Instead the solution chose to “build” two optional ground stations at location nodes 7 and 9 which correspond to Wallops Island, Virginia and Santiago. The total cost of the architecture is \$156.6M; tied for lowest cost with the previously disaggregated architecture. However, this architecture only has 976 Gbit remaining on both satellites at the end of the simulation. Not only is this architecture cheaper than NASA’s published architecture by \$0.8M but it outperforms in total data throughput by 125% (14058 Gbit to 11266 Gbit) and decreases end of scenario data storage by 74% (3800Gbit vs 976 Gbit). Thus, this system has higher data throughput at a lower total cost than NASA’s planned architecture shown in Figure 10.

9.7 Conclusion and Future Work

This Chapter demonstrated how a satellite network mixed integer linear program is a powerful tool for analyzing system architecture options given a sensor architecture concept. Using the MILP to examine alternative network options for NASA’s HypsIRI mission resulted in system architecture with 20% higher data throughput and cost savings of \$0.8M. Furthermore, automating such a program provides the ability to rapidly analyze the tradeoffs amongst multiple sensor architecture concepts. In this regard, the work discussed in this section complements the work previously covered in Chapters 3 through 8 and provides a more comprehensive means of evaluating the utility of total mission architectures: outputs from the Earth Observation Architecture Isoperformance Model (EO-AIM) can be formatted and input into the network MILP which creates more meaningful heuristics than the simple algorithms previously used.

Additionally, the program as built is incredibly flexible. Constraints can be added or removed, the objective function can be changed, and new architectural options can be added to the input files. And not only can the solver provide the optimal assignments to the architecture, but it can also calculate an optimal link schedule. For instance, an existing space system architecture can be input into the model as “fixed” with the objective of finding an optimal link schedule only. If there are multiple assets, the user can even specify constraints for how much data from each asset must reach ground by the end of the scenario. For either use case, as long as the problem is solvable, the integrated solver will find an optimal solution amongst more options than an engineer can analyze through more traditional methods.

While substantial progress was made in creating the network model, there are a few areas of future work that will add to the fidelity of the results and increase the model’s overall utility. The model was built such that these upgrades in functionality can be implemented with minimal changes.

The first area of future work is improving the model’s accuracy in the generation of sensor data as a function of time based on a given mission’s concept of operations. The model currently applies average data generation rates across the duration of the simulation but a more accurate method would encompass further integrating with orbit propagation software and creating generation rules such as “if observation area is in day: generate data; else: no generation.” This could be accomplished with some effort using STK.

A second area of work is creating more accurate data rates for the time-varying accesses. Currently, the model just applies the communication system’s specified data rate across the entire interval of access. This can be improved by integrating the use of time-varying azimuth, elevation, and range (AER) data for each link to estimate the time-varying data rate via a

parametric equation or more comprehensive model. AER data is already computed during the step for computing time-varying accesses with STK. The work that remains to be done is implementing use of the data and leveraging research on data-rate versus AER for several technologies, including FSO.

A third major improvement would to incorporate dynamic power constraints that balance bus power generation, sensor power draw, and communication subsystem power draw. This could be done with modest effort by constructing a power versus time matrix similar to the data rate versus time matrix. Power generation would a be function of the satellites geometry relative to the sun over time, which can be computed using the orbit propagation model. Power consumed is largely a function of whether the subsystems (communications and the sensor) are being used at any given time.

Appendix A: NASA Systems Engineering Figures

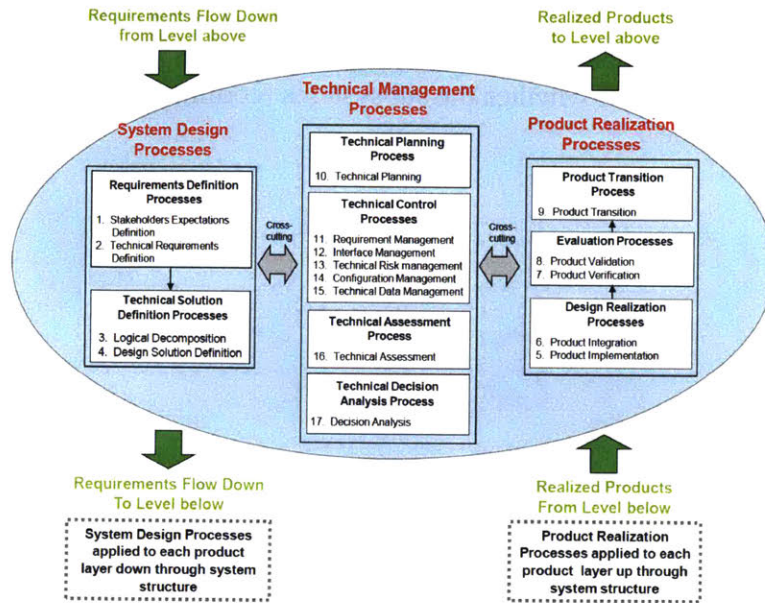


Figure 22: The NASA systems engineering engine. Reprinted from NASA Systems Engineering Handbook, NASA, 2016, p. 6.

	Phase	Purpose	Typical Outcomes
Pre-Formulation	Pre-Phase A Concept Studies	To produce a broad spectrum of ideas and alternatives for missions from which new programs/projects can be selected. Determine feasibility of desired system, develop mission concepts, draft system-level requirements, assess performance, cost, and schedule feasibility; identify potential technology needs, and scope.	Feasible system concepts in the form of simulations, analysis, study reports, models, and mockups
Formulation	Phase A Concept and Technology Development	To determine the feasibility and desirability of a suggested new system and establish an initial baseline compatibility with NASA's strategic plans. Develop final mission concept, system-level requirements, needed system technology developments, and program/project technical management plans.	System concept definition in the form of simulations, analysis, engineering models and mockups, and trade study definition
	Phase B Preliminary Design and Technology Completion	To define the project in enough detail to establish an initial baseline capable of meeting mission needs. Develop system structure end product (and enabling product) requirements and generate a preliminary design for each system structure end product.	End products in the form of mockups, trade study results, specification and interface documents, and prototypes
Implementation	Phase C Final Design and Fabrication	To complete the detailed design of the system (and its associated subsystems, including its operations systems), fabricate hardware, and code software. Generate final designs for each system structure end product.	End product detailed designs, end product component fabrication, and software development
	Phase D System Assembly, Integration and Test, Launch	To assemble and integrate the system (hardware, software, and humans), meanwhile developing confidence that it is able to meet the system requirements. Launch and prepare for operations. Perform system end product implementation, assembly, integration and test, and transition to use.	Operations-ready system end product with supporting related enabling products
	Phase E Operations and Sustainment	To conduct the mission and meet the initially identified need and maintain support for that need. Implement the mission operations plan.	Desired system
	Phase F Closeout	To implement the systems decommissioning/disposal plan developed in Phase E and perform analyses of the returned data and any returned samples.	Product closeout

Table 20: The systems engineering project phase descriptions. Reprinted from NASA Systems Engineering Handbook, NASA, 2016, p9.

Appendix B: Satellite Buses and Launch Vehicles

1 Selected Small Satellite Bus Information

Vendor	Bus Model	Satellite Dry Mass (kg)	Max Payload Mass (kg)	Orbit Average Power (W)	Downlink Data Rate (Mbps)
Blue Canyon	Microsat [43]	~50	70	Up to 200	150
Millennium Space Systems	Altair [44]	50	50	90	N/A
York	S-CLASS [45]	65	85	100	5, 50
Surrey Satellite US	SSTL-100 [46]	83	15	24	80
Surrey Satellite US	SSTL-150 [47]	103	50	40	80
Surrey Satellite US	SSTL-150 ESPA [36]	115	65	160	2
Lockheed Martin	LM100 [48]	95	152	155	N/A
Sierra Nevada	SN-50L [49]	80	50	100	N/A
Sierra Nevada	SN-100L [49]	103	100	200	N/A
Ball Aerospace	BCP 300 [50]	139	250	200	2
QinetiQ Space	P200 [51]	<130	70	70	N/A
Orbital ATK	ESPASat [52]	70	30	30	N/A

Table 21: Small satellite bus information.

2 Emerging Commercial Small Satellite Launch Providers

Launch Vehicle/ Service	Max Payload Mass	First Launch
Firefly Alpha [53]	600kg (SSO, 500km)	3Q2019
Rocket Lab Electron [54]	225kg (SSO, 500km)	21 Jan 2018
Virgin Orbit Launcher One [55]	300kg (SSO, 500km)	2018
Spaceflight Industries Ride Share [40]	50-1000kg (Varies)	N/A
Gilmour Space Eris [56]	400kg (LEO)	Q42020

Table 22: Small satellite launch vehicles launch capacities compared to small satellite busses and payloads.

Appendix C: Architecture Variables and Model Data Structure Descriptions

1 The Orbit

The orbit of a single satellite is described by the six classical orbit elements shown, in Figure 25, eccentricity, semi-major axis, inclination angle, longitude of ascending node, argument of periapsis, and true anomaly. When selecting orbit, major tradeoffs occur between

- Launch costs for achieving a specific altitude
- Viewing geometry for a specific altitude
- And required aperture to meet ground sampling distance requirements

Table 23 describes user input parameters which form the orbit data structure. Table 24 describes parameters that are calculated during execution of the model.

For this thesis, the domain of the orbit variable is limited to circular orbits between 400km and 1200km above the Earth’s surface. The model was designed so that other orbits can be used but analysis in this thesis is limited to circular low earth orbits. Non-circular low Earth orbits rely more heavily on modeling software such as STK for determining the number of satellites required to satisfy revisit rate and coverage requirements.

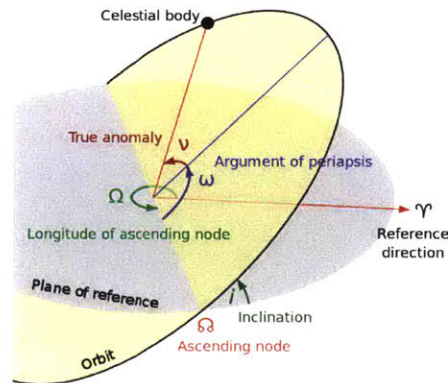


Figure 23: Classical orbit elements of a celestial body. Reprinted from Wikipedia Commons.

Field	Description	Domain, Equation, or Assignment
Average Altitude, h	The average altitude of the satellite from the surface of the Earth. For a circular orbit the altitude at a given time is approximately the average altitude.	{400, 450, ..1200}
Semi Major Axis, a	The sum of the periapsis and apoapsis distances divided by two. For the case of a circular orbit around Earth, it is the altitude at apogee plus the radius of the Earth.	$a = h + R_e$
Eccentricity, e	The shape of the orbital ellipse, describing how much it is elongated compared to a circle. For a circular orbit, $e = 1$.	{1}
Inclination Angle, i	The vertical tilt of the ellipse with respect to the reference plane, measured at the ascending node.	Equation (15)

Argument of Periapsis, ω	The orientation of the ellipse in the orbital plane, as an angle measured from the ascending node to the periapsis. Specified by user for STK modeling.	$\{[0,360]\}$
Right Angle of Ascending Node, Ω	Horizontally orients the ascending node of the ellipse (where the orbit passes upward through the reference plane) with respect to the reference frame's vernal point. Specified by user for STK modeling.	$\{[0,360]\}$
Initial True Anomaly, ν	The position of the orbiting body along the ellipse at the time of simulation start. Specified by user for STK modeling.	$\{[0,360]\}$
Apogee Altitude, h_a	Altitude from Earth of the satellite at apogee. Used for STK automation.	
Perigee Altitude, h_p	Altitude from Earth of the satellite at perigee. Used for STK automation.	

Table 23: User inputs for the orbit assignment.

Field	Description	Domain, Equation, or Assignment
Orbital Period, P	The amount of time for a satellite in a given orbit to complete one full rotation around the central body.	Equation (6)
Average Ground Velocity, V_g	The average velocity of a satellites orbit projected onto the ground of the central body.	Equation (7)
Minimum Central Angle, λ_{min}	Minimum Central Angle of Satellite Coverage to cover the required swath width as a function of inclination angle and the longitudinal shift of the earth. This is requisite for global coverage.	Equation (10)
Minimum Nadir Angle, η_{min}	The minimum nadir angle from the satellite corresponding to the minimum central angle for coverage.	Equation (11)
Elevation Angle, ϵ	The elevation angle from the outermost point of the satellites ground coverage swath.	Equation (4)
Slant Range, R	The range in kilometers from the satellite to the outermost edge of its ground swath.	Equation (5)
Required Field of View, FOV_{req}	The required field of view from the satellite in the case that the required swath is covered a single satellite/single instrument.	Equation (14)
Planes Required, $\#_p$	The number of planes required as a function of the longitudinal shift covered per orbit and the required revisit rate.	Equation (13)
Coverage Number/Slots Per Plane, $\#_c$	The number of satellites required to cover the longitudinal shift per orbit as a result of the minimum elevation angle constraint being breached. Starts at 1.	Any real positive integer.

Table 24: Variable parameters in the orbit data structure calculated during model execution.

2 Focal Plane Disaggregation

A focal plane disaggregation scheme refers to creating tailored sensors to meet different data requirements by separating the focal planes based on disaggregation criteria. This allows designers to specify different aperture sizes, take advantage of satellite disaggregation options, etc. The disaggregation options implemented in the model (which can be changed) include:

- No disaggregation
- Disaggregation of focal plane by calibration method required (reflective band vs emissive band calibration)

- Disaggregation based on whether focal plane thermal regulation is required (disaggregation of power/mass requirements). One such case may include having a longwave uncooled microbolometer and VISNIR focal plane as part of the same instrument while a cooled medium wave focal plane is its own instrument.

3 Sensor Scan Method

The sensor scan method describes the mode in which the remote sensor observes the Earth as it travels along its projected track on the Earth's surface, referred to as the along track direction. Cross-track refers to the direction perpendicular to the satellites along-track movement. There are three dominant scan methods for passive optical Earth observation systems:

Whiskbroom Scan Method. The whiskbroom sensor scan method, shown in Figure 24, describes the mode in which an array of pixels is oriented in the along-track direction and a mechanism is used to sweep the array in the cross-track direction such that a swath of Earth is observed at a specified scan rate.

Pushbroom Scan Method. - A pushbroom sensor, shown in Figure 24, is one in which an array of n by m pixels are oriented such that it covers the entire cross track swath passively as it moves in the along-track direction. The design has the advantage of less moving parts. However, it is challenged by the optical form tradeoff between field of view and aperture diameter, which is limited by wavefront error effects on receiver sensitivity.

Step and Stare. Step and stare scan method refers to a sensor with a gimbaled telescope which "stares" at a patch of the Earth for a fixed amount of time needed to meet sensitivity requirements before "stepping" to the next patch on the grid. These types of sensors are typically located in Geostationary Orbit. Modeling of step and stare sensors is not included in this thesis, though the capability can be added to the model.

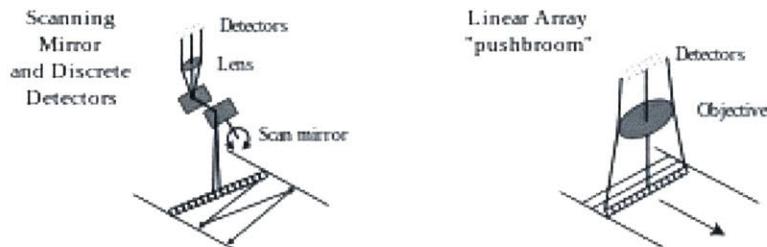


Figure 24: A whiskbroom scanning mirror sensor is show on the left. A pushbroom sensor is show on the right. Reprinted from Wikimedia Commons.

4 The Focal Plane Assembly

The Focal Plane Assembly subsystem is composed of one or more focal plane detector arrays integrated with read out integrated circuits. The model's focal plane data structure is composed of user inputs parameters described in Table 25 and those dynamically computed during model execution described in Table 26. The fixed-value parameters provide the necessary information for constraining the model such that sensitivity figure of merit calculations can be made. The major fixed parameters of the focal plane include its type (e.g. photocarrier), the material it's made out of, and pixel pitch. The domain of focal plane assemblies includes all applicable focal planes available for purchase.

Each focal plane has wavelength data structures assigned to its active wavelengths field. A wavelength data structure contains values which are user defined such as the sensitivity requirement or are dynamically calculated during execution of the model such as the number of samples required to meet the sensitivity requirements. Sensitivity is either specified in terms of cloud top radiance and signal to noise ratio or observation temperature and noise equivalent difference in temperature (NEdT). Description of the wavelength data structure can be found in Table 27 and Table 28.

Parameter	Description	Domain, Equation, or Assignment
Focal Plane Name	A string representing the name of the focal plane.	All strings
Reference String	A unique identifier for the focal plane used throughout the model to match different lookup files and use case rules.	All strings
Lookup File	The filename of the lookup file to be used for looking up values of specific parameters specified below file is an .xls with the name '{Ref. String} lookup.xls'. It consists of tabs 'Operating Mode', 'Dark Current', 'Quantum Efficiency', and 'Readout Rate'	All valid excel file names *.xls
Type/ Sensitivity Model	The type refers to the class of focal plane and as a result the applicable sensitivity. The two types covered in this research include photocarrier and microbolometer focal planes.	{'photocarrier', 'microbolometer'}
Lower Bound Wavelength (m)	The lower bound wavelength for the band of wavelengths that can be assigned to the focal plane. For the purposes of the model described herein, wavelength bounds between focal planes must be mutually exclusive and completely exhaustive in their coverage of the required wavelengths.	All real positive numbers
Upper Bound Wavelength (m)	The upper bound wavelength for the band of wavelengths that can be assigned to the focal plane. For the purposes of the model described herein, wavelength bounds between focal planes must be mutually exclusive and completely exhaustive in their coverage of the required wavelengths.	All real positive numbers
Spectral Type	Spectral type refers to the 'type' of wavelengths that can be assigned to a focal plane of the domain {'reflective', 'emissive'} where 'reflective' refers to the primary source radiance coming from solar reflection (4-2.5 μ m) whereas 'emissive' refers to the primary source of radiance coming from the Earth itself. This field is used for disaggregation purposes.	{'reflective', 'emissive'}
Material	Material of the focal plane.	All strings
X-Dimension of Pixel Pitch, d_x	The width of a focal planes pixel in the cross-track oriented direction.	All real positive numbers

Y-Dimension of Pixel Pitch, d_y	The width of a focal planes pixel in the along-track direction.	All real positive numbers
X-Dimension Number of Pixels COTS, n_{xCOTS}	The number of specified pixels in the cross-track direction given the focal plane is has an “off-the-shelf” configuration	All real positive numbers
Y-Dimension Number of Pixels, n_{yCOTS}	The number of specified pixels in the along-track direction given the focal plane is has an “off-the-shelf” configuration	All real positive numbers
Operational Temperature, T_{op}	The required operational temperature of the focal plane.	All real positive numbers
Thermal Regulation Required	A Boolean declaration of whether thermal regulation of the focal plane is required as a function of the operational temperature specified above. This field is used for disaggregation purposes.	{‘yes’, ‘no’}
Bits Per Pixel/Quantization, Q_e	The number of bits used in quantization of the analog pixel signal.	All real positive numbers
Operational Mode	The operational mode of the focal plane array which is used for determining the electron well capacity per pixel and readout noise.	All operating modes defined by focal plane specification.
Electron Well Capacity Per Pixel, W_e	The number of electrons a detector well can hold. Can be a specified integer or ‘lookup’. If ‘lookup’, the lookup table is used to determine the well capacity based on the operational mode of the focal plane.	{‘lookup’, All real positive numbers}
Frame Rate (Hz), ν	A default read out framerate setting for the focal plane can be specified during user input. This supports the future work of including step and stare sensor creation. The default value is superseded by the framerate required by physical constraints imposed by whiskbroom or pushbroom scan methods.	All real positive numbers
Max Readout Rate vs Framerate (Gbps), RO_{max}	This value is used to constrain the problem during the sensitivity-performance algorithm. It specifies the maximum number of bits that can be readout in a second. Thus Framerate*Number of Pixels \leq Max Readout Rate	All real positive numbers
Readout Noise, $\sigma_{n.read}$	The readout noise of the focal plane. It can be a specified integer or the string ‘lookup’ which signal the model to use a lookup table to set the value based on the operational mode of the focal plane.	{‘lookup’, All real positive numbers}
Dark Current, J_d	This value represents the dark current $J(\frac{A}{(cm)^2})$ of a focal plane operating at operating temperature, T_{op} . It can be specified as an integer or the string ‘lookup’. If ‘lookup’ is specified, a lookup table is used based on T_{op} .	{‘lookup’, All real positive numbers}
Spatial Noise, $\sigma_{n,spatial}$	Spatial noise represents the effect of residual non-uniformity and fixed pattern noise on the FPA [25, p. 292] This quantity should be specified by the focal plane manufacturer.	All real positive numbers
Johnson Noise, $\sigma_{n,Johnson}$	The noise associated with thermal carrier motion in resistive circuit elements. [25, p. 284] This quantity should be specified by the focal plane manufacturer	All real positive numbers
Microns Between Bands	Given a multispectral sensor with multiple wavelengths being collected on the same focal plane, Microns between Bands represents the minimum number of microns between two active pixel rows collecting different wavelengths. This represents “dead” pixels as result of applying wavelength filter coating etc. and is shown in Figure 28.	
Quantum Efficiency, QE_{FPA}	The ratio of expected photocarriers available from incident photons. Applicable only for photocarrier focal planes. If ‘lookup’ is specified, a lookup table is retrieved from the focal	{‘N/A’, ‘lookup’, [0,1]}

plane look up file. A number between 0 and 1 can be specified if the engineer wishes to apply a constant QE to all wavelengths assigned to the focal plane.

Table 25: User inputs for focal plane component definition.

Parameter	Description	Domain, Equation, or Assignment
Active Wavelengths	Wavelength data structures that are assigned to the focal plane data structure based on its minimum and maximum wavelength values.	Wavelengths structures
Frame Rate (Hz), ν	The default value is superseded by the framerate required by physical constraints imposed by whiskbroom or pushbroom sensors.	Equation (46) or Equation (53)
X-Dimension Instantaneous Field of View (IFOV), $IFOV_x$	The calculated instantaneous field of view for a pixel in the cross-track direction.	Equation ()
Y-Dimension IFOV, $IFOV_y$	The calculated instantaneous field of view for a pixel in the along-track direction.	Equation ()
Window Dimension, W_e	Windowing refers to aggregating pixels to form a larger pixel. A window dimension of 2 refers to two by two array of actual pixels forming a single composite pixel.	All real positive numbers greater than 0.
Total X-Dim. Pixels, n_x	The total number of pixels in the cross track required by the focal plane to meet requirements, including “dead” pixels between wavelengths as applicable.	Equation ().
Total Y-Dim Pixels, n_y	The total number of pixels in the along-track required by the focal plane to meet requirements, including “dead” pixels between wavelengths as applicable.	Equation ().
Maximum Rows Readout per Second	The maximum number of pixel rows in the along-track direction that can be read out in the case of a whiskbroom sensor.	All real positive integers
X-Dim FPA Count	The number of FPAs of “off-the-shelf” configuration in mosaic formation in the cross-track dimension to provide the total number of X-dim. Pixels.	All real positive integers
Y-Dim FPA Count	The number of FPAs of “off-the-shelf” configuration in mosaic formation in the along-track direction to provide the total number of Y-dim. Pixels.	All real positive integers

Table 26: Focal plane variable parameters in the orbit data structure calculated during model execution.

5 Wavelengths

Parameter	Description	Domain, Equation, or Assignment
Wavelength, λ	The center wavelength specified by the requirement.	All real positive numbers
Bandwidth, $\lambda_u - \lambda_l$	The bandwidth of the wavelength from which radiance is measured. It is specified by the upper and lower cutoffs.	All real positive numbers
Typical Radiance, L_{typ}	Typical cloud top radiance for a given wavelength as specified by the requirements.	All real positive number
Maximum Radiance, L_{max}	The maximum radiance for a given wavelength as specified by the requirements	All real positive numbers
Signal to Noise Ratio Required, SNR_{req}	The required signal to noise ratio for a given wavelength.	All real positive numbers

Typical temperature, T_{typ}	The typical temperature of the target as specified by the requirements.	All real positive numbers
Maximum temperature, T_{max}	The maximum temperature of the target as specified by the requirements.	All real positive numbers
Noise Equivalent Difference in Temperature Required, $NEdT_{req}$	The maximum noise equivalent difference in temperature for a given wavelength.	All real positive numbers
Frame Rate, ν	The framerate of the focal plane. Inherited from the focal plane data structure.	All real positive numbers
Dark Noise, $\sigma_{n,dark}$	The noise generated from the detector even when no signal is detected.	Equation (36)
Readout Noise, $\sigma_{n,read}$	The readout noise of the focal plane.	Inherited from focal plane.
Spatial Noise, $\sigma_{n,spatial}$	Spatial noise represents the effect of residual non-uniformity and fixed pattern noise on the FPA [25, p. 292] This quantity should be specified by the focal plane manufacturer.	Inherited from focal plane.
ADC Noise, $\sigma_{n,ADC}$	The noise generated from converting the analog signal into a digital one.	Equation (37)
Noise Equivalent Power, NEP_{bol}	A sensitivity figure of merit associated with microbolometers. NEP can be specified by the user based on the microbolometers calculations or calculated based on specified $NEdT$ at a sample wavelength.	Specified or Equation (44)

Table 27: User input fields for wavelength data structures.

Parameter	Description	Domain, Equation, or Assignment
Quantum Efficiency, QE	The quantum efficiency of the focal plane the wavelength is assigned to. Lookup upon assignment.	{{0,1}}
Transmission Efficiency, τ_o	The transmission efficiency of the optical form. Transmission efficiency can be specified by the user or dynamically calculated based the spectral information of optics components, such as absorption.	{{0,1}}
Max Integration Time, τ_{max}	The maximum amount of time a sample can be integrated over given limitations in electron well capacity and framerate.	Equation (32)
Single Frame Electrons, N_{SF}	Applied to photocarriers only. The number of photocarriers generated during a single frame.	Equation (33)
Shot Noise, $\sigma_{n,shot}$	Noise resulting from a Poisson process and dependent on receiver signal.	Equation (35)
Calculated Single Frame, SNR_{SF}	The sensitivity of a single sampling frame.	Equation (38)
Calculated Number of Sample Required to Meet Sensitivity Requirement	The number of sample frames required to meet the sensitivity required.	All real positive integers

Table 28: Wavelength variable parameters in the orbit data structure calculated during model execution

6 Optical Form

The optical form subsystem, interchangeably referred to as the telescope of the system, is composed of a series reflective (mirrors) or refractive (lens) elements which focus radiance collected at the entrance pupil onto the focal plane array. Initial parameters of the optical form

which must be entered into the model are shown in Table 29 in the Appendix. The domain of the optical form includes, but is not limited to, Three Mirror Anastigmat, Reflective Triplets, Three Lens Refractive, Four Mirror Anastigmat, and Five Mirror Anastigmat Telescopes.

Parameter	Description	Domain, Equation, or Assignment
Optical Form Name	The name of the optical form being described via the other parameters.	All strings
Reference String	A unique identifier for the optical form used throughout the model to match different lookup files and use case rules.	All strings
Use Case	The sensor scan method for which the optical form can be used. Scanning mirror and rotating telescope are two different instantiations of a whiskbroom scan method.	{‘Pushbroom’, ‘Scanning Mirror’, ‘Rotating Telescope’}
Real Entrance Pupil	A descriptor of whether the optical form has a real entrance pupil.	{‘yes’, ‘no’}
Real Exit Pupil	A descriptor of whether the optical form has a real exit pupil.	{‘yes’, ‘no’}
Number of Elements (Mirrors/Lenses)	An integer which describes the number of elements in the optical form.	Any real positive integer
Max Field of View Constraint {degrees}	Describes the maximum field of view achievable by the system.	Any real positive number
Maximum Element Size Constraint {m}	Describes the maximum diameter of any element within the optical form.	Any real positive number
General Relationship Constraint {m*deg ² }	Describes the relationship between achievable field of view versus aperture size up to the maximum field of view constraint. This is a parameterized function used to constrain the model. Any combination of aperture diameter or Field of View with values greater than this constraint are expected to result in too big of wavefront error.	Any real positive number
Largest Element Function	The expected largest element size given an entrance pupil and field of view.	A function specified in the MATLAB function handle format ‘@(Var)(Function(Var))’
Minimum F Number Constraint	The minimum/shortest F number possible for a given optical form.	Any real positive number.
Wavelength Constraints	Any constraints relating to wavelengths. For example, many lens systems are localized in performance to only one of VISNIR, MWIR, or LWIR band.	
Along Track Constraint	The along track field of view constraint if different from the General relationship constraint. It must be smaller than the allowable cross track field of view calculated via the general relationship.	Any real positive number.
Mass	The expected mass of the system given an entrance pupil and a field of view.	A function specified in the MATLAB function handle format ‘@(Var)(Function(Var))’

Table 29: Optical form user inputs.

7 Thermal Regulation Subsystem

Thermal regulation requirements are largely influenced by the focal planes being used for infrared wavelengths. Current photocarrier technology requires a combination of components such as an active or passive cooler, cold stops, and dewars to limit noise from self-emission. In contrast, certain focal plane technologies like microbolometers designed for the long wave infrared spectrum do not require cooling. (Though, a microbolometer's sampling framerate is limited by the thermal conductance of the material.)

Thermal regulation subsystem options discussed in this thesis include a cooler and dewar configurations where the thermal regulator is either an active cryocooler accompanied by control electronics or a passive cryoradiator such as the one used on VIIRS. The dewar is a mechanical enclosure which houses the focal planes requiring cryogenic temperatures and is integrated with the thermal regulator via conductive thermal straps.

8 Electronics Subsystem

The electronics subsystem is responsible for processing the received signal, managing the active components of the system, and communicating with the host for data transfer. Information on the electronics subsystem is largely proprietary and is not discussed in this thesis. In general, increases in data volume collected and/or sensor complexity leads to an increase in the number of components required in the electronics subsystem. The model uses this information and use constraints to determine the necessary components of the electronics. A simple example rule present in this model is using total samples per second to calculate how many data processing components are needed.

9 Calibration Subsystem

The calibration subsystems are used for on-board calibration of the sensor data. There are different subsystem variable domains for the reflective and emissive bands. Only subsystem options which are included in the model will be described in this section. However, a systems engineer can easily expand the model to include more options.

There are two methods for calibration of reflective wavelengths included in the model: a solar diffuser assembly and vicarious calibration. A description of how a solar diffuser and solar diffuser stability monitor work can be found in t [4]. Vicarious calibration is the technique of using Earth's, or even the Moon's, natural features for reflective calibration. The reader is referred to [57] for a brief summary on vicarious calibration.

The only emissive calibration subsystem included in the model is a black body calibration source which is used for calibration of the emissive bands. This type of calibrator is a plate with high emissivity (close to 1), which emits at a known temperature. As such, the expected radiance is known to some acceptable uncertainty and can be used to calibrate observations from the Earth.

10 Sensor Data Structure Description

The sensor data structure contains information on sensor level characteristics such as the altitude the sensor must operate at, the aperture diameter, and the focal length. Descriptions for all of the fields of the sensor data structure is shown in Table 30.

Field	Description	Domain, Equation, or Assignment
Orbit Altitude, h	Average altitude of the orbit inherited from the orbit data structure.	Orbit altitude.
Sensor Count	The number of sensors required in an orbital plane to cover the required swath. Initialized at 1.	All real positive numbers.
Focal Planes	The focal planes in the sensor.	Focal plane data structures.
Sensor Scan Method	The sensor scan method described in Section 3 of this appendix.	{‘whiskbroom’, ‘pushbroom’}
Optical Form	The optical form assigned to the sensor.	Applicable optical form data structures
Aperture Diameter, D	The aperture diameter of the optical form/telescope.	See algorithms in Appendix E.
Focal Length, f	The focal length of the system.	Equation (17)
F Number	The F number of the systems.	f/D
Cross Track Field of View Per Sensor, FOV_N	The cross-track field of view per sensor.	Equation (22)
Mass	Mass of the sensor.	Equation (56)
Power Required	Power required by the sensor	Equation (57)
Sensor Electronics	The sensor electronics subsystem assigned to the sensor.	Sensor electronics data structure.
Emissive Calibrator	The emissive calibrator assigned to the sensor.	Emissive calibrator data structure
Reflective Calibrator	The reflective calibrator assigned to the sensor.	Reflective calibrator data structure.
Sensor Data Rate	The sensors data rate.	Equation (50) and Equation (54)
Optics Per Satellite	In the case of a pushbroom sensor, the number of optical form telescopes in a single sensor. Otherwise one.	Any real positive number
Scan Time, τ_{scan}	In the case of a whiskbroom, the time between scans. Otherwise, N/A.	Equation (51)
Cost	A data structure resulting from a SEER cost estimate of the file. The data structure includes: Sensor ID, A file path to the estimate file, Previous files used to build the most current estimate, an export of the SEER WBS cost table including Monte Carlo analysis figures.	SEER Cost table.

Table 30: Sensor data structure fields.

Appendix D: Useful Equations

1 Helpful Circular Orbit Equations

First, it helps to recall a few key equations for the geometry of a circular orbit around a central body. For a more detailed explanation the reader is referred to Wertz et al. chapters 8 through 10.

The semi-major axis of a circular orbit at altitude h is the sum the altitude and the Earth's radius R_E .

$$a = R_E + h \quad (2)$$

The Earth angular radius ρ can be calculated from the radius of the Earth and the orbit altitude.

$$\sin \rho = \frac{R_E}{R_E + h} \quad (3)$$

Furthermore, we have the following relationship between central angle, elevation angle, and nadir angle.

$$\varepsilon = \frac{\pi}{2} - \lambda - \eta \quad (4)$$

Slant range is the maximum range from satellite to the edge of its field of view on the target is from the satellites viewing geometry and altitude.

$$R = R_E \left(\frac{\sin(\lambda)}{\sin(\eta)} \right) \quad (5)$$

The orbital period of a satellite can be calculated from the semi-major axis, a , and the specific gravity of Earth, μ .

$$P = 2\pi\sqrt{a^3/\mu} \quad (6)$$

The orbital ground velocity is given as:

$$V_g = \frac{2\pi R_e}{P} \quad (7)$$

2 Calculating the Number of Satellite Planes to meet Global Coverage Requirements and Revisit Rate Requirements

To calculate the number of satellite planes to meet global coverage and revisit rate requirements for a circular orbit we first approximate the change in longitude at the equator. This equation can be found with more explanation in Fortescue et al., page 124. $\tau_E = 61,164.1$ is the Earth sidereal day in seconds.

$$\Delta L = -\left(\frac{2\pi P}{\tau_E} + \frac{3\pi J_2 R_E^2 \cos i}{a^2(1-e^2)^2}\right) rad/robit = \quad (8)$$

The change in longitude can be converted to angular change of the ground per circular orbit perpendicular to the satellites track.

$$\Delta T = \Delta L \cos(90 - i) \quad (9)$$

Thus, we can calculate the minimum Earth central angle with respect to the satellite that must be covered in order to meet coverage and revisit rate requirements. We introduce a variable dubbed coverage number, $\#_c$, which is initially set to one and represented the number of satellites required to cover this track given the minimum elevation angle constraint.

$$\lambda_{min} = \frac{\Delta T}{2\#_c} \quad (10)$$

The minimum Earth central angle is used to calculate the minimum nadir angle, from the satellites perspective, to meet coverage requirements.

$$\eta_{min} = \tan^{-1}\left(\frac{\sin \rho \sin \lambda_{min}}{1 - \sin \rho \cos \lambda_{min}}\right) \quad (11)$$

Equation (4) is then used to calculate elevation angle, ε , at the edge of a single satellites track given η_{min} and λ_{min} . It is necessary to check that the minimum elevation angle constraint is met.

$$\varepsilon \leq \varepsilon_{min} \quad (12)$$

If not, we increment $\#_c$ and compute Equations (10) through (12) once more.

Finally, we can calculate the minimum number of satellites planes necessary to meet coverage and the revisit rate requirement, R_{req} .

$$\#_p = \frac{180}{2 * \lambda_{min}} * \frac{P}{R_{req}} * \#_c \quad (13)$$

At this point we can also define the required field of view per orbital plane.

$$FOV_{req} = 2\eta_{min} \quad (14)$$

If there is a requirement for the sensor to be sun synchronous then the inclination angle is selected such that (Fortescue 124) then the inclination angle must assume a value such that Equation (15) is satisfied.

$$\frac{-3\pi J_2 R_E^2 \cos i}{a^2(1-e^2)^2} = 2\pi \frac{\tau_E}{\tau_{ES}} \frac{P}{\tau_E} \text{ rad/orbit} \quad (15)$$

Where:

τ_E is the sidereal period

$\tau_{ES} = 3.155815e7$ is the orbital period of Earth round the sun

J_2 is the second gravitational term

a is the semimajor axis

e is the eccentricity of the orbit

i is the inclination angle of the orbit

If sun-synchronous is not specified, then the inclination angle is specified there is maximum track width allowing for better coverage with (hopefully) less satellites.

3 Common Geometry Relationships for All Sensor Scan Methods

First, its important recall the viewing geometry fundamentals of any optical system with discrete detectors. The relationship between detector size in one dimension and the projection on the target is given by magnification:

$$M = \frac{f}{h} = \frac{d}{X} \quad (16)$$

Where:

f is the focal length measured from entrance aperture to detector

h is the altitude at entrance aperture above the target

d is the detector pitch in one dimension

X is the sampling distance width in one dimension

Thus, given a ground sample distance requirement, required focal length is calculated as a function of detector pitch and orbit altitude:

$$f = \frac{dh}{X} \quad (17)$$

Instantaneous field of view (IFOV) for a single dimension is then given as:

$$IFOV = \frac{d}{f} = \frac{X}{h} \quad (18)$$

Thus, a relationship between field of view and the number of detectors, n , to cover a field can be formed:

$$n = \frac{FOV}{IFOV} \quad (19)$$

Equation (18) can be substituted into Equation (19) to produce Equation (20) which helpful for determining the number of pixels in an array required to meet a specific field of view in one dimension as a function of pixel size and focal length.

$$n = \frac{f}{d} FOV \quad (20)$$

4 Pushbroom Scan Method Constraints

To begin, a constraining relationship between Field of View (FOV) and aperture diameter (D) can be described as:

$$FOV \leq \frac{O}{D} \quad (21)$$

Where O is a constant of units $meter * degrees$ or $meter * degrees^2$, depending on the optical form.

FOV_N is defined here as the field of view of one of N optical forms on a satellite where all N satellites have the same field of view. FOV_N is calculated as a function of nadir angle η and N .

$$FOV_N = \frac{2}{N} * \eta \quad (22)$$

Substituting Equation (21) into Equation (22) yields:

$$D \leq \frac{N * O}{2 * \eta} \quad (23)$$

Equation (23) forms an upper bound on the entrance pupil diameter as a function of satellite coverage nadir angle and the number of telescopes/sensors used to cover the nadir angle. This is illustrated in Figure 25 below. We can also calculate the number of satellites required in the same orbit at different true anomalies based on FOV_{req} and FOV_N . Recall that FOV_{req} is the field of view required to cover the longitudinal shift and meet coverage requirements.

$$\#_{sv} = \frac{FOV_{req}}{FOV_N} \quad (24)$$

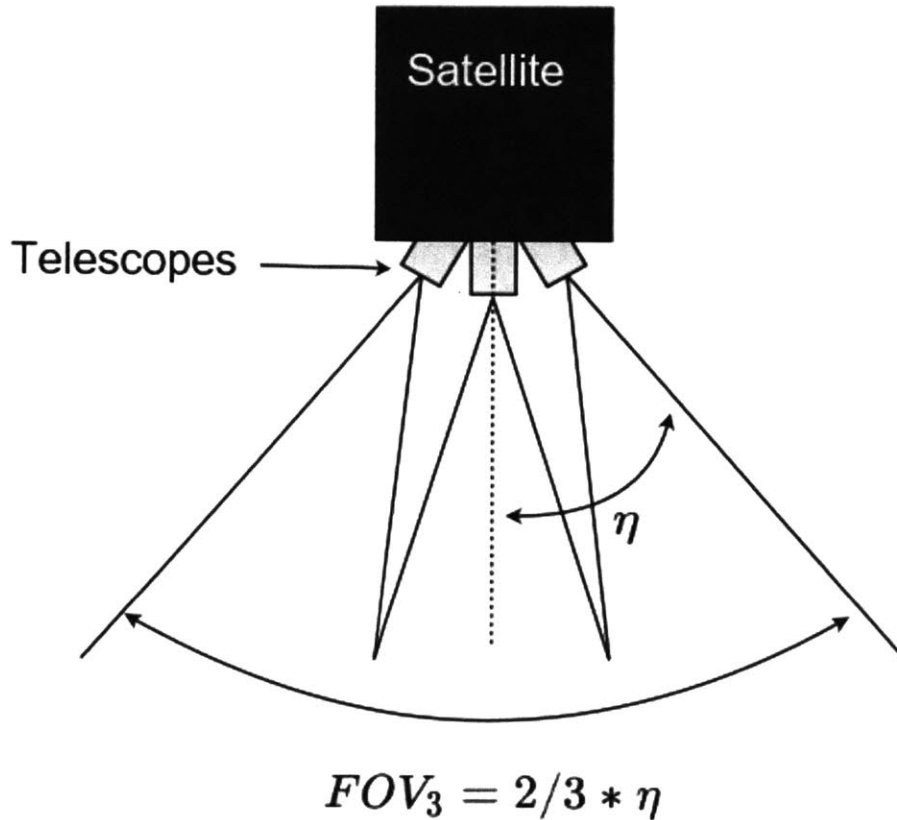


Figure 25: Sensor field of view as a function of number of optical forms and nadir angle.

Given there is a requirement for at least one wavelength to be diffraction limited, based on the ground spatial resolution requirement, a lower bound constraint on the aperture diameter can be formed.

$$D \geq \frac{R * \lambda_d}{X_{req}} \quad (25)$$

Where R is the slant range, λ_d is the diffraction limited wavelength, and X_{req} is the required ground spatial resolution. Writing the slant range as a function of central angle, nadir angle, and the Earth's radius, Equation (25) can be rewritten as:

$$D \geq R_e * \frac{\sin(CA)}{\sin(\eta)} * \frac{\lambda_l}{X_{req}} \quad (26)$$

where $CA = \frac{\pi}{2} - \varepsilon - \eta = \frac{\pi}{2} - \arccos\left(\frac{\sin(\eta)}{\sin(\rho)}\right) - \eta$ with $\sin(\rho) = \frac{R_e}{R_e+h}$

Thus, we can constrain the aperture diameter by both Equations (23) and (23)

$$R_e * \frac{\sin(CA)}{\sin(\eta)} * \frac{\lambda_l}{X_{req}} \leq D \leq \frac{N * O}{2 * \eta} \tag{27}$$

To illustrate this concept, Figure 26 represents an example scenario where:

$$\begin{aligned} O &= 3m * deg \\ \lambda_l &= 12e^{-6}m \\ X_{req} &= 375m \end{aligned}$$

$h = 800km$ (nadir altitude)

The dotted line which trends up is the aperture diameter's lower bound. The solid lines which trend down towards a constant represent the aperture diameter's upper bound for different values of optical telescopes per satellite, N . The intersection of a solid line and the dotted line is the maximum nadir angle versus aperture size achievable by N of the same optical form telescopes on the same satellite. To determine the field of view of the satellite, one must multiply the maximum nadir angle of the satellite by two. The field of view of a single optical form telescope is the satellite field of view divided by N .

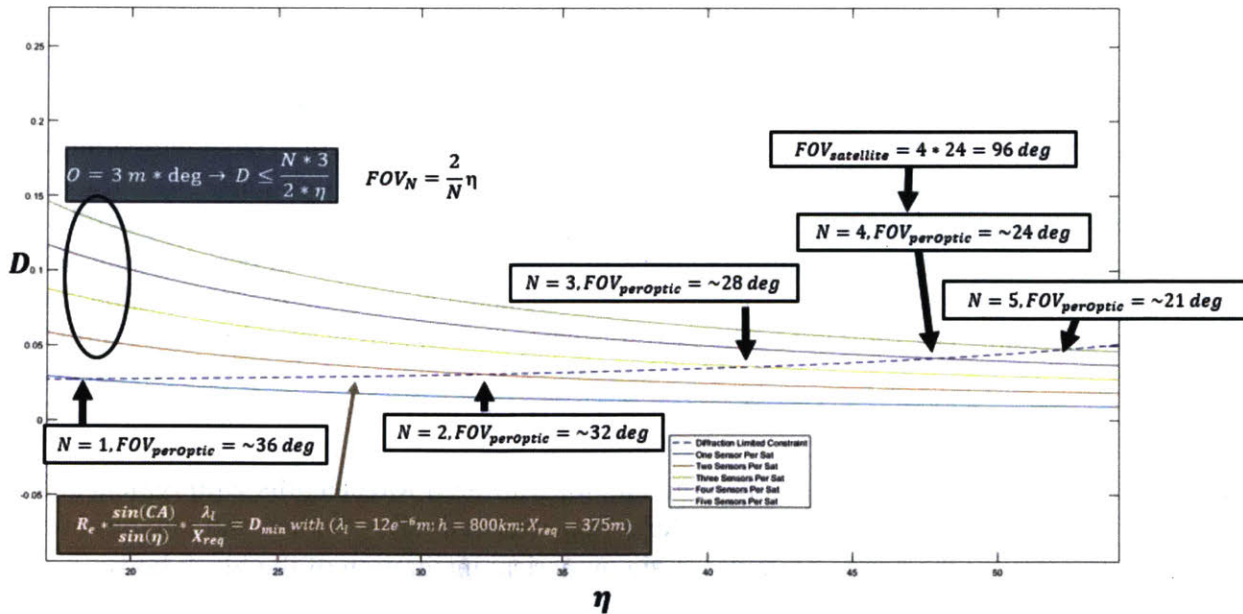


Figure 26: Satellite field of view for varying numbers of optics.

Figure 27 shows how many satellites are required to satisfy a required field of view, FOV_{req} , equal to the VIIRS field of view of approximately 112 degrees. It can be seen that the optimal number of optical forms per satellite versus satellites required is when $N = 2$ with $FOV_2 = 32deg$ and $FOV_{sat} = 64deg$. Finding aperture size is a function of optics per sat allows for the

creation of a general algorithm for creating LEO pushbroom options which meet the sensitivity requirements of the mission.

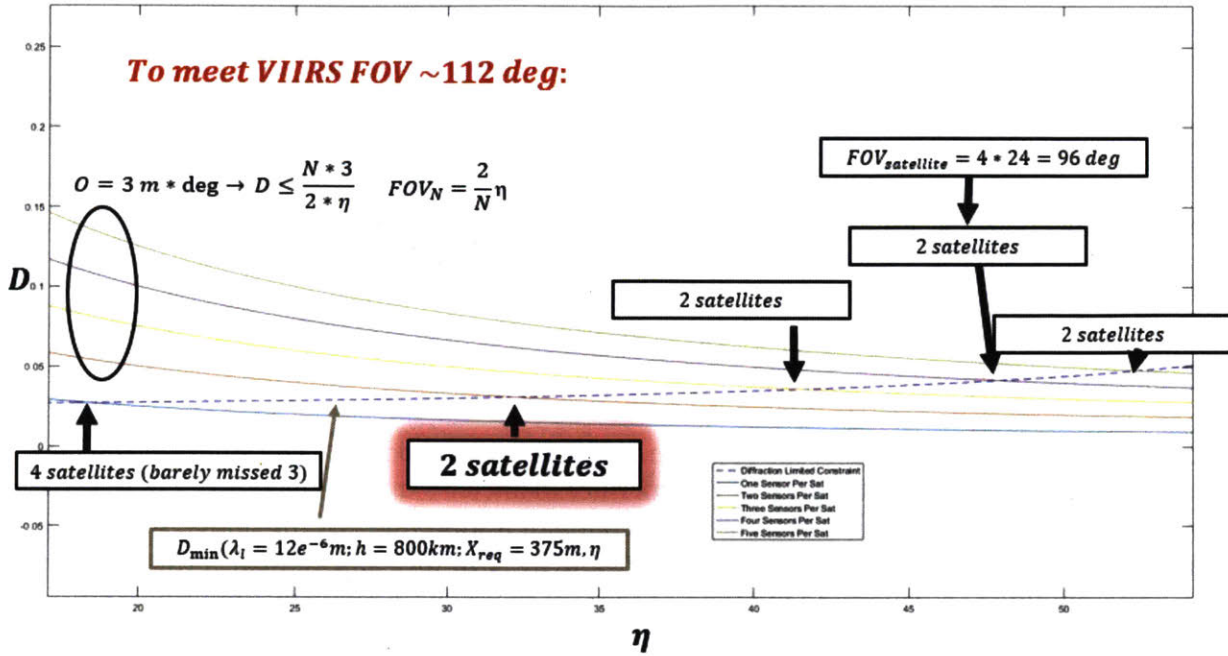


Figure 27: Number of satellites required to meet required field of view as a function of number of optics on a satellite.

5 Photocarrier Sensitivity Model

As seen in Appendix F, either typical radiance and maximum radiance or typical temperature and maximum temperature are specified for each band. In practice visible through short infrared wavelengths are specified in terms of typical spectral radiance L_{typ} and maximum spectral radiance L_{max} at the center wavelengths in units of watts per meter squared per steradian per micron ($W/m^2/sr/\mu m$). Radiance for is band are calculated by multiplying these values by the specified bandwidth. The figure of merit used for these wavelengths is signal to noise ratio SNR.

$$L_{\lambda} = L_{typ}(\lambda_u - \lambda_l) [W/m^2/sr] \quad (28)$$

Temperature values are specified for medium and long infrared wavelengths with typical and maximum temperatures represented by T_{typ} and T_{max} respectively. Typical and maximum spectral radiance over the band is calculated via numerical integration of the black body equation.

$$L_{\lambda} = \int_{\lambda_l}^{\lambda_u} \frac{2\pi hc^2 \epsilon}{\lambda^5 (e^{\frac{ch}{kT\lambda}} - 1)} d\lambda [W/m^2/sr] \quad (29)$$

Where:

λ_l is the lower bound of the wavelength band
 λ_u is the upper bound of the wavelength band
 h is Planck's constant
 c is the speed of light
 $\varepsilon = 1$ is the emissivity of Earth
 k is Boltzmann's constant
 $T = \{T_{typ}, T_{max}\}$ is the temperature

Power from the ground sample reaching the entrance pupil is calculated based on the focal planes pixel pitch and the sensors F-number.

$$P_\lambda = L_\lambda d_x d_y \left(\frac{D}{f}\right)^2 \left(\frac{\pi}{4}\right) [W] \quad (30)$$

The calculated radiance values for each band are then used to calculate the number of electrons, N_e , being produced per second as a function of:

*(Power Entering the Instrument) * (Conversion of Power to Photons) * (Efficiencies)*

$$N_{e\lambda} = L_\lambda d_x d_y \left(\frac{D}{f}\right)^2 \left(\frac{\pi}{4}\right) * \frac{\lambda}{hc} * \tau * QE \quad (31)$$

Where:

L_λ is the radiance over the band with center wavelength λ
 d_x, d_y are the dimensions of a pixel on the focal plane in meters
 D is the aperture diameter
 f is the focal length
 λ is the center wavelength
 h is Planck's constant
 c is the speed of light
 QE is the focal planes quantum efficiency at Lambda
 τ is the transmission efficiency of the optics.

Transmission efficiency is the efficiency at which photons are transmitted from the telescope entrance pupil to the focal plane. It accounts for losses from material absorption, misalignment, etc. It can either be set at a constant value (typically .75) or calculated dynamically based on the optical form and aft optic components such as mirrors, filters, etc. Specific values used to calculate the transmission efficiency are not discussed in this thesis.

Quantum efficiency is conversion factor ranging from 0 to 1 and is specified by the focal plane manufacturer. It is a ratio of the expected photocarriers available from incident photons over the number of incident photons. The model uses look up tables to set the value for quantum efficiency.

Next, the sensitivity model calculates the maximum integration time as a function of electrons per second during periods of maximum radiance. The maximum integration time represents the minimum between the amount of time a pixel can integrate over, as limited by its electron well capacity W_e , or the framerate of the sensor.

$$\tau_{max} = \min \left(\left(\frac{W_e}{N_{e-max}} + J_d \frac{d_x d_y}{e} \right), \frac{1}{f} \right) \quad (32)$$

Where:

W_e is the full well electron capacity of the pixel

N_{e-max} is the calculated number of electrons per second using L_{max}

J_d is the dark current density

d_x, d_y are the dimensions of a pixel on the focal plane in meters

e is the charge of an electron

f is the readout frame rate of the sensor in hertz

Thus, the electrons resulting from a single frame are the product of the maximum integration time and the electrons generated per second.

$$N_{SF} = N_e * T_{max} \quad (33)$$

System noise includes shot noise, dark-current noise, Johnson noise, readout noise, spatial noise, and analog-to-digital conversion noise. [25, p. 292]

$$\sigma_n = \sqrt{\sigma_{n,shot}^2 + \sigma_{n,dark}^2 + \sigma_{n,Johnson}^2 + \sigma_{n,read}^2 + \sigma_{n,spatial}^2 + \sigma_{n,ADC}^2} \quad (34)$$

Shot noise, dark noise, and ADC noise are dynamically calculated for each sensor option by the model. Johnson, readout, and spatial noise are specified by focal plane manufacturer.

$$\sigma_{n,shot} = \sqrt{N_{SF}} \quad (35)$$

$$\sigma_{n,dark} = \sqrt{\frac{J_d d_x d_y T_{max}}{e}} \quad (36)$$

$$\sigma_{n,ADC} = \frac{2^{-Q} N_{max}}{\sqrt{12}} \quad (37)$$

Recall Q is the number of bits used to represent a data sample.

SNR of a single frame is calculated as the number of electrons generated from a single frame over the noise of a single frame.

$$SNR_{SF} = \frac{N_{SF}}{\sigma_n} \quad (38)$$

The number of frames required to meet SNR requirements can then be calculated for a background limited sensor:

$$n_{\lambda_i sample} = \left(\frac{SNR_{req}}{SNR_{SF}} \right)^2 \quad (39)$$

If performance is specified in terms of NEdT, a few more steps are required. First, the change in radiance per change in temperature is calculated:

$$\frac{dL}{dT} = \int_l^u \frac{2h^2c^3\varepsilon}{k\lambda^6} * \frac{e^{\frac{ch}{kT\lambda}}}{T^2 \left(e^{\frac{ch}{kT\lambda}} - 1 \right)^2} d\lambda \quad (40)$$

Second, the net equivalent radiance is calculated from the SNR of a single frame:

$$NER = \frac{L}{SNR_{SF}} \quad (41)$$

Finally, we can calculate NEdT:

$$NEdT = \frac{NER}{\frac{dL}{dt}} \quad (42)$$

The number of samples required is calculated by:

$$n_{samples} = \left(\frac{NEdT}{NEdT_{req}} \right)^2 \quad (43)$$

6 Microbolometer Sensitivity Model

A microbolometer is a focal plane in which infrared radiance on the material results in a change of the materials resistance is measured as a signal. The sensitivity model used for a microbolometer uses the specified noise equivalent temperature difference, $NEdT_{spec}$, for a specified wavelength to calculate a noise-equivalent power which can then be used to calculate sensitivity for all wavelengths assigned to the microbolometer. Noise-equivalent power (NEP) is defined as the required signal power to achieve a signal to noise ratio of one. Calculating NEP is also dependent on the F-number the manufacturer used to measure $NEdT_{spec}$

$$NEP_{bol} = \frac{\pi NER_{bol} d_x d_y}{4(F\#_{bol})^2} \quad (44)$$

Where:

$$NER_{bol} = \frac{NEdT_{spec}}{\frac{dL}{dT}}$$

Equation (30) multiplied by the transmission efficiency is then used to calculate the single frame power for each required wavelength which enables calculation of the single frame SNR by dividing power at the pixel by the bolometers noise equivalent power.

$$SNR_{SF} = \frac{Ld_x d_y \left(\frac{D}{f}\right)^2 \left(\frac{\pi}{4}\right) \tau}{NEP_{bol}}$$

Once SNR_{SF} is known, Equations (39) through (43) are used to calculate the number of samples needed to meet sensitivity requirements.

Appendix E: Model Operations and Constraints

1 Reading Input Data and Creating Model Data Structures

The first operation of the model is ingesting data inputs required from the user. Inputs include mission requirements, engineering constraints, model options, available focal plane technologies, optical forms, other subsystems, satellites bus options, and launch vehicles. The model then creates data structures for each component technology and maps wavelength data structures to the applicable focal plane data structures. Figures of the Excel based model input windows can be seen in Appendix 1. To simplify the model, focal planes must be input such that they cover the required wavelengths in a mutually exclusive and completely exhaustive manner. In other words, the systems engineer who enters the wavelength requirements into the model, must also input the focal planes to which those wavelengths will be assigned. This is required because the fixed values of the focal plane are critical parameters for constraining the problem. Analyzing options composed of different combinations of focal planes is as simple as running the model multiple times with different focal plane inputs. Once the user's inputs are entered, the model can be executed.

2 Assignment of the Orbit and Creation of Orbit Data Structures

For the circular LEO scenario, the first assignment is the altitude of the orbit which is input by the user. Non-LEO orbits may be entered but require more information than just altitude. Engineering constraints for orbit parameter assignments include the minimum allowable elevation angle the Earth may be viewed at as well as whether the system is required to operate in a sun-synchronous orbit.

Together, the altitude, revisit rate requirement, minimum elevation angle, and sun-synchronous options are used to calculate the orbit data structure parameters described in Table 8 of the Appendix. Equations for calculating the parameters can be found in Appendix D.

3 Assigning Disaggregation Scheme, Scan Method, and Optical Forms to Create Initial Sensor Data Structures

After orbit structures are created, the model proceeds to create sensors for each orbit via assignment of focal plane disaggregation scheme, sensor scan method, and optical form. For each disaggregation scheme, including no disaggregation, sensor options are created for each sensor scan method with the exception of when the Orbits Coverage Number, $\#_C$, is greater than one. This constraint is implemented because it is unlikely a whiskbroom sensor scan method would be pursued in orbits which require multiple satellites to cover the longitudinal shift due to of the minimum elevation angle constraint.

A sensor data structure is created via a function that takes in the mission requirements, orbit data structure, sensor scan method, and focal planes. The disaggregation assignment determines which focal planes are assigned to the sensor structure. The sensor scan method determines the sensitivity performance algorithm used to calculate the telescope's aperture diameter size, focal length, and array size based on the scan methods unique physical constraints, the mission's sensitivity requirements, and the appropriate focal plane sensitivity model. The sensitivity performance algorithm for the scan method is called after a valid optical form is assigned to the sensor, which allows the problem to be adequately constrained. The sensor data structure is initialized with the empty fields show in Appendix C, which are populated as the model propagates.

4 Assigning the Pushbroom Scan Method - Creating LEO Pushbroom Imagers

A sensor is a pushbroom once the sensor scan method variable is assigned the value "Pushbroom." This assignment limits which optical forms can be assigned to subsequent search states along the branch as was previously shown in Figure 3. Following the solution search branching for each optical form, another round of branching occurs in search of solutions which vary in the number of telescopes per satellites and the number of satellites per plane for satisfying coverage requirements. Each telescope on the satellite covers a different window of the satellite's overall field of view as shown in Figure 25 of Appendix D. Additionally, if the field of view for the entire satellite is less than the field of view required for the orbital plane in order to meet coverage requirements, than multiple satellites which are spaced along the same track are required. Overall, the solution search branching occurs using the following algorithm:

1. The number of telescopes per satellite is initialized at 1.
2. An initial sensor solution is found using the pushbroom sensitivity algorithm. A solution might require multiple satellites in a "string of pearls" configuration such that the longitudinal shift is covered.
3. The number of telescopes per satellite is incremented by one and steps one through three are repeated until only one satellite is required to cover the longitudinal shift.

The reader is referred to Appendix D for more information on how the number of telescopes and satellites are calculate using the aperture diameter versus maximum field of view parametric relationship. This particular constraint is notable because the relationships were created at AAC and allow the problem to be constrained such that solution search is automated. It may also be

helpful for the reader to first review the primer on optical geometry fundamentals in Appendix D, Section 3.

5 Pushbroom Sensitivity Performance Algorithm for Assigning Component Variables

The pushbroom sensitivity performance algorithm calculates values for several key pushbroom sensor characteristics required to satisfy sensitivity requirements. First, the initial aperture and focal length are calculated based on the wavelength required to be diffraction limited. Next the sensitivity per sample is calculated for each wavelength as a function of F-number and sensor framerate. From the sensitivity per sample, the number of time delayed integration samples per wavelength, $n_{\lambda, samples}$, required to meet sensitivity requirements is calculated. The number of required sampler per wavelength is then used to calculate the number of pixels in the along-track dimension, n_y , using Equation (45), where Sp_{λ} is the number of “dead” pixel rows in between two different wavelength areas of a focal plane. “Dead” pixels are a result of the applying wavelength filters directly to the focal plane and can be seen in Figure 28. The number of pixels in the cross track, n_x , are calculated by substituting the field of view of the telescope, FOV_N , the calculated aperture size, and focal length into Equation (20). Additionally, the framerate of the focal plane is calculated as a function of the sampling distance and the ground track velocity as shown in Equation (46).

$$n_y = \sum_i (n_{\lambda, samples} + Sp_{\lambda}) \quad (45)$$

$$v = V_g \frac{f}{dh} \quad (46)$$

A possible “initial” sensor solution state is formed when the aperture diameter, focal length, focal plane array size(s), and frame rate(s) have been calculated. The following constraints must be satisfied for the “initial” sensor solution to be valid:

- The physical F-number limitation of the optical form assigned to the sensor. This represents the complexity of making optics with short focal length to aperture diameter ratios.

$$\frac{f}{D} \leq F\#_{min} \quad (47)$$

- The field of view per telescope is less than the maximum cross track field of view for the optical form, if specified. This value represents the field of view where the aperture versus field of view relationship is no longer valid.

$$FOV_N \leq FOV_{Omax} \quad (48)$$

- The number of samples, which correlates to the number of pixels used for time delayed integration, to meet sensitivity requirements does not cause the maximum along track field of view constraint to be breached. If a maximum along track field of view constraint is not specified, it is assumed the $FOV_{Ymax} = FOV_N$.

$$n_y IFOV_y \leq FOV_{Ymax} \quad (49)$$

The readout data rate calculated is less than the maximum readout possible by the focal plane integrated electronics.

$$DR = vQn_x \sum_i n_{\lambda_i samples} \leq RO_{max} \quad (50)$$

Where:

v is the readout rate in Hz

Q is the quantization of a sample (generally 10-12 bits per samples)

n_x is the number of pixels required to cover the FOV_N given by Equation (20)

$\sum_i n_{\lambda_i samples}$, is the number of active pixels in the along-track direction

RO_{max} is the specified maximum readout rate of the focal plane

If any of these constraints are breached, the algorithm takes the below steps based on systems engineering best practices to meet requirements. The steps are taken in order of increasing system complexity and cost. Additionally, a subsequent step is taken only if requirements and constraints are still not being met.

- 1) First, the aperture diameter is increased if the minimum F-number constraint is not active. Increasing aperture decreases the number of samples required to meet sensitivity requirements. It is the least costly option for adjusting the system to meet requirements.
- 2) The aperture and focal length are increased. This decreases the instantaneous field of view for and allows for more samples to be taken while meeting field of view constraints.

The algorithm exits as invalid if the maximum readout rate constraint is exceeded and the maximum F-number constraint is active. If the readout rate constraint is violated then there is no means for meeting requirements because the result of taking step 2) repeatedly is a monotonic increase in the number of pixels to meet requirements.

After ensuring that no constraints are violated and the solution is valid, the algorithm increments the number of optics per satellites as previously discussed. A major presupposition to this algorithm is that the only constraint for the number rows used for time delayed integration, for each wavelength, is the optical form's along-track field of view constraint. However, more constraints can be incorporated into this algorithm if the user desires. Finally, an additional area of expansion concerning this algorithm is treating operating temperature as a variable which can be determined via the sensitivity algorithm, rather than as a user input.

6 Whiskbroom Sensitivity Performance Algorithm for Assigning Component Variables

The model uses the whiskbroom sensitivity performance algorithm to create whiskbroom imager solutions if the whiskbroom scan method and a compatible optical form are assigned. First, the model calculates initial values for focal length and aperture diameter using the diffraction limited wavelength requirement and required spatial resolution. Next, the required scan time is calculated based on the maximum field of view allowed by the optical form, orbit altitude, required along-track overlap, and ground velocity. Scan time is defined here as the required amount of time between whiskbroom scans for coverage requirements to be satisfied and is used to calculate the required framerate of the focal plane(s) via Equation (53). Also, in the whiskbroom case, the maximum field of view in the along-track direction is limited either by the optical form constraint or the shortest field of view amongst all of the focal planes resident on the sensor.

$$\tau_{scan} = FOV_{Ymax} * \frac{h}{O_{AT}V_g} \quad (51)$$

$$FOV_{Ymax} = \min(FOV_{Omax}, n_{y1}IFOV_{Y1} \cdot n_{yn}IFOV_{Yn}) \quad (52)$$

$$v = \frac{2\pi}{\tau_{scan}IFOV_y} \quad (53)$$

Once frame rate is known, the number of rows required to meet sensitivity requirements, $n_{\lambda_i samples}$ is calculated for each wavelength followed by a check to see if whiskbroom constraints are breached. The algorithm terminates when the following constraints are satisfied:

- The total number of rows, or samples, to meet sensitivity requirements must be less than the maximum number of rows that can be read out. The maximum number of rows is calculated based on the frame rate, the number of pixels per row, and the max readout rate of each focal plane. This equation is similar to Equation (50) except that rows are now in the long-track direction, n_y , and the result is multiplied effective duty cycle: required field of view (FOV_{req}) divided by a full scan (2π).

$$DR = \frac{FOV_{req}}{2\pi} v Q n_y \sum_i n_{\lambda_i samples} \leq RO_{max} \quad (54)$$

- The field of view required by the number of rows, or samples, to meet sensitivity requirements must be less maximum cross track field of view constraint.

$$\sum_i (n_{\lambda_i samples} + Sp_\lambda) * IFOV_x \leq FOV_{Xmax} \quad (55)$$

If any of these constraints are breached by any focal plane, the algorithm takes the following steps until a solution satisfying all constraints is reached.

- 1) If the minimum F-number constraint is not active, increase aperture diameter to increase power and thus decrease the required number of samples.
- 2) If the minimum F-number constraint is active, increase the window dimension of the focal plane. Windowing is the aggregation of adjacent pixels to form a pixel of larger size. As can be intuited from Equation (30), this increases power received at the focal plane by the square of the window dimension, if F-number is constant, for each pixel on the ground at the expense of doubling the focal length to meet spatial resolution requirements.

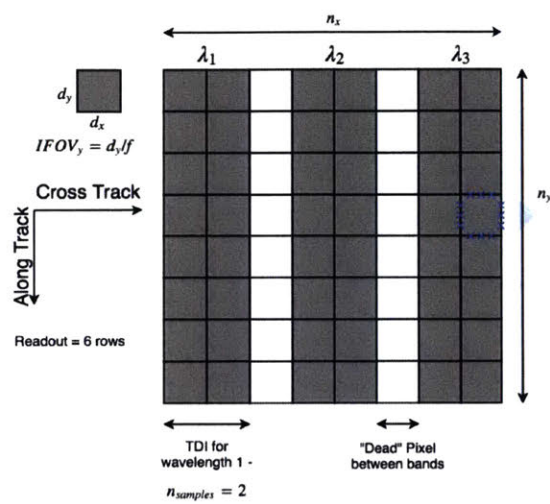


Figure 28: Focal plane on a whiskbroom sensor with two rows of the same wavelength for multiple samples.

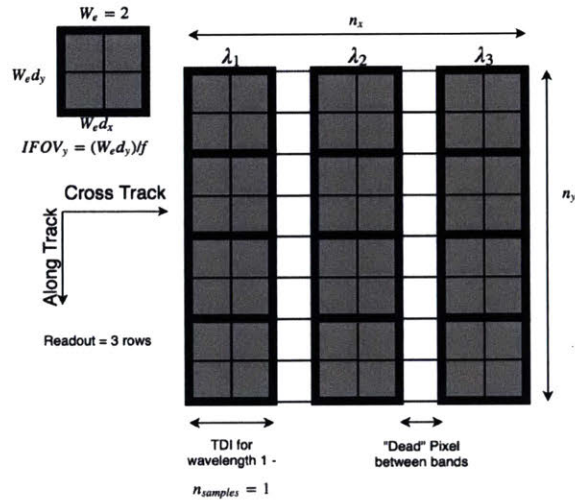


Figure 29: Focal plane on a whiskbroom sensor with a window dimension of two thus forming 2x2 aggregate pixels.

7 Creating an Initial Cost Estimate

Once the model creates an array of initial solution sensor states for all disaggregation assignments of a given orbit, SEER-H is used to create a cost estimate for each initial sensor solution use batch files and a template SEER-H command files for every component assigned to the initial solution (optical form, focal planes, etc.).

After constructing the sensor, the model reads which components have been used to build the sensor. The model retrieves the associated template file and edits the file to match the values of variable parameters calculated by the model. For instance, if the model found a viable solution using a three mirror anastigmat telescope with the largest element diameter 15 cm, then the SEER-H three mirror anastigmat command template's largest element field is edited to 15cm. In the case of a focal plane template, the array dimensions fields are edited to match the values, n_x and n_y , previously calculated. An example command file for a Si PIN focal plane can be found in Table 41 of Appendix 4.

After editing the command files for the focal planes and optical form, the model aggregates them and executes batch files to create a SEER-H cost estimate for the initial sensor. The cost estimate is organized, via the aggregate command file, into a standard work break down structure within the SEER-H program. Furthermore, the path for the cost estimation file is added to a parameter of the sensor data structure for easy traceability. A block diagram of the actions and data at each step is show in Figure 30. Screenshots of the SEER GUI can be seen in Appendix G.

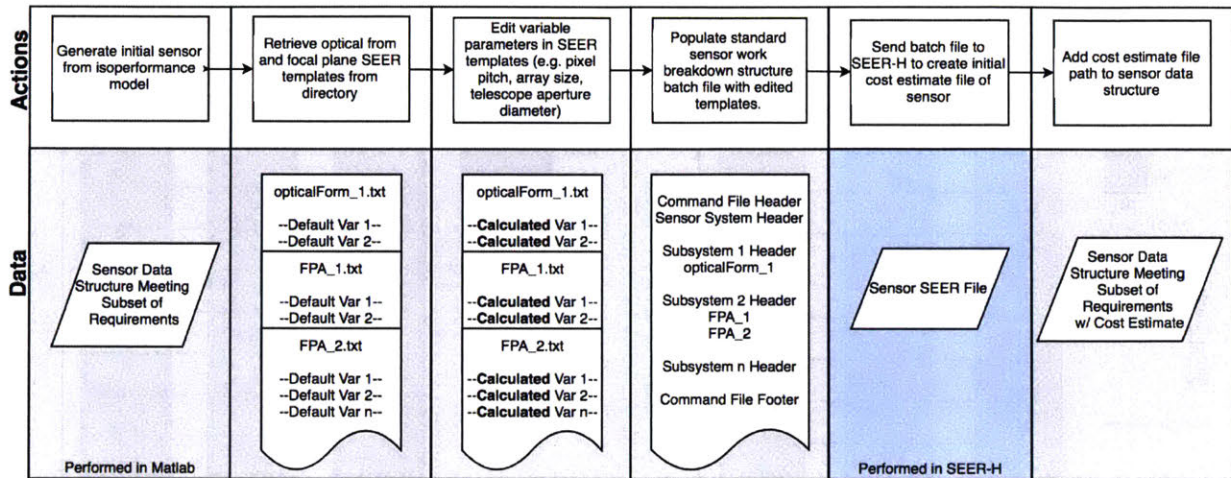


Figure 30: Initial SEER-H cost estimate process.

8 Assigning Enabling Subsystems

After an initial cost estimate is created, the remaining subsystems required for a sensor are assigned following rules and constraints. In some cases, assignment to the subsystem variable is straight forward via constraint propagation and no branching is required. For example, rules and constraint are used to prune the domain of thermal regulation subsystem variable based on the operating temperature(s) of the focal plane(s). This mirrors reality; a systems engineer would not arbitrarily design a system with an active cryocooler when focal planes operate at 300K. In cases where there is no definite rule for choosing one subsystem option over another, branching occurs such that the model can search for multiple solutions using different subsystem assignments for a specific subsystem variable (see Figure 5). For instance, when assigning a reflective wavelength calibration source, solution branching is required since there are multiple options with different cost, risk, and performance tradeoffs. Vicarious calibration is software intensive with no substantial hardware required for implementation, thus removing mass and hardware complexity from the system. On the other hand, vicarious calibration performance has little heritage for high reliability mission class A missions and may be considered “riskier”. In contrast, a solar diffuser assembly is relatively complicated when factoring in the calibrating mechanism and its monitor, the SDSM, but has heritage in multiple missions, such as VIIRS. In this case, it is valuable to create a sensor solution for both options so that downstream cost impacts can be presented to stakeholders who make the ultimate decision.

Subsystem components are defined in the model’s user input excel file as a series of parameters, parametric functions, and use case rules/constraints. An input parser file converts

the component information into Matlab data structures and creates subsystems options as a composite of the components structures. Some component values are pre-defined, such as mass for an active cryocooler control electronics circuit. Other component values, like the power required by the active cryocooler to maintain the focal plane assemblies operating temperature, are dynamically calculated once the component is added to the sensor. Defining a new subsystem option is relatively easy and just requires adding components to the input file. Once a state, stemming from the initial sensor state, has valid assignments for all subsystems variables, a final SEER cost estimate for the sensor is created.

Specific subsystem options, rules, and constraint details will not be presented in this thesis. Instead, general examples based on publicly available information will be used.

8.1 Focal Plane Thermal Regulation Subsystem

The domain for the focal plane thermal regulation subsystem includes:

- None, no thermal regulation required
- Passive cryoradiator
- Active cryocooler

In the case of assigning a thermal regulation subsystem to a sensor, a combination of system engineering rules and solution branching is used. The system engineering rules are a result of interviewing one of AAC's experts with a Ph.D. in thermal space systems.

The rule is to assign a thermal regulation subsystem to a sensor based on the lowest operating temperature of the resident focal planes. As seen in Figure 31, rather than have a discrete cutoff from the assignment of one category to the next, a smaller in-between range exists where the preceding solution state is branched to create two solutions with the different thermal regulation options. The motivation for pursuing a no cryocooler solution is simple: the resulting sensor has less mass and requires less power than one with a cooling mechanism. Switching from cryoradiator to cryocooler as temperature declines is a result of the complexity and size required from a cryoradiator to maintain the proper operating temperature. Creating a parametric for cryoradiator mass as a function of focal plane operating temperature is recommended future work.

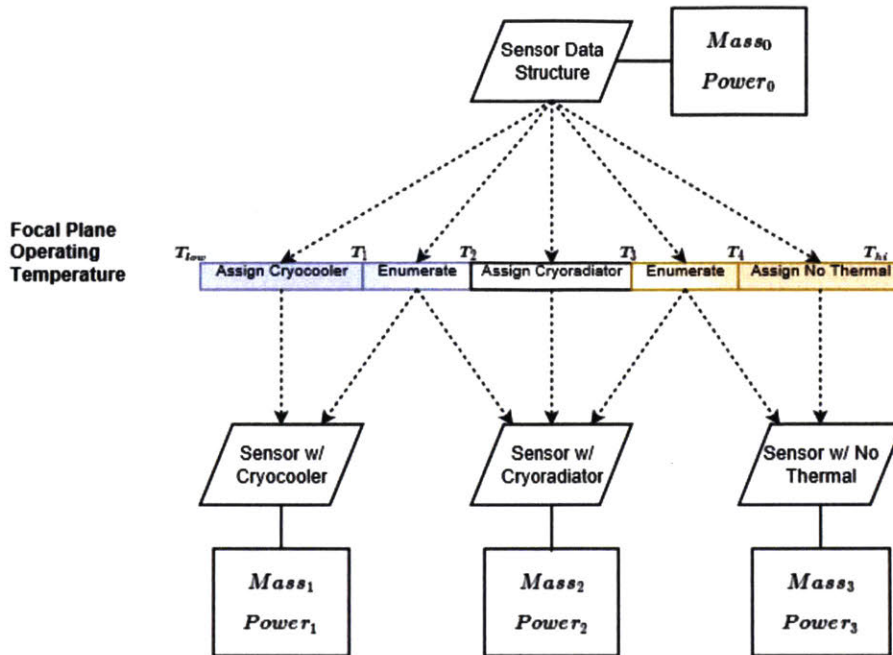


Figure 31: Assigning thermal regulation to sensor based on focal plane operating temperature rule.

8.2 Calibration Subsystem

Calibration subsystems are assigned using a constraint which states that a sensor must have a means for calibrating the spectral categories it operates over. This is the motivation behind defining a spectral category field for each focal plane, which must assume one two values, “reflective” or “emissive.”

For categories with multiple means of calibration, solution branching is used. For example, given two means of emissive calibration, two means of reflective calibration, and an initial sensor solution with focal planes operating in the reflective and emissive spectrum, there will be four potential sensor solutions created. Each solution corresponding to a unique combination of the calibration options. Final sensor size, weight, power, complexity, and cost characteristics will be unique for each combination.

8.3 Electronics Subsystem

The model assumes the electronics subsystem is a modular architecture with components, namely field programmable gate arrays (FPGA) and circuit cards, added to the subsystem based on characteristics of the sensor. A AAC sensor electronics expert was consulted in creating general rules for determining the quantity of each component. An example rule is using the number of pixels collected per second to determine the required quantity of FPGAs used for data processing. Another straightforward rule is that the number of focal plane interface electronic cards must be equal to the number of focal planes present in a given sensor. Detailed review of the electronics subsystem is not included in this thesis for proprietary reasons. Instead, general components such as “power component”, “focal plane interface component”, “mechanisms

control component”, etc. are used. Recommended future work includes conducting a more exhaustive rules-based systems engineering analysis for the sensor electronics.

9 Final Sensor Characteristics

After all subsystems are assigned, final sensor mass and power is calculated. Mass is a summation of all of the component masses which make up the sensor. Likewise, average power required is the summation of all of the average power required by the sensor. Sensor data rate is a function of how many ground samples are captured per second multiplied by the quantization number.

$$m_{total} = \sum_i m_{c_i} \quad (56)$$

$$p_{total} = \sum_i p_{c_i} \quad (57)$$

10 Final SEER Cost Estimate of the Sensor

After sensor options have been modified or created based on subsystem assignment rules, constraints, and solution branching, a final sensor SEER-H estimate is created for each option, as illustrated in Figure 32. Similar to the previous operation, SEER-H templates for subsystem components are retrieved and parameter variables are adjusted before an aggregated SEER-H command file is created. SEER-H is executed using batch files and commands to first open the initial SEER-H cost estimate of the initial sensor and then add subsystem components to the appropriate work break down structure levels. SEER-H’s built in Monte Carlo capability is used and the mean cost, standard deviation, and new sensor SEER-H cost estimate file path is added to the final sensor data structure.

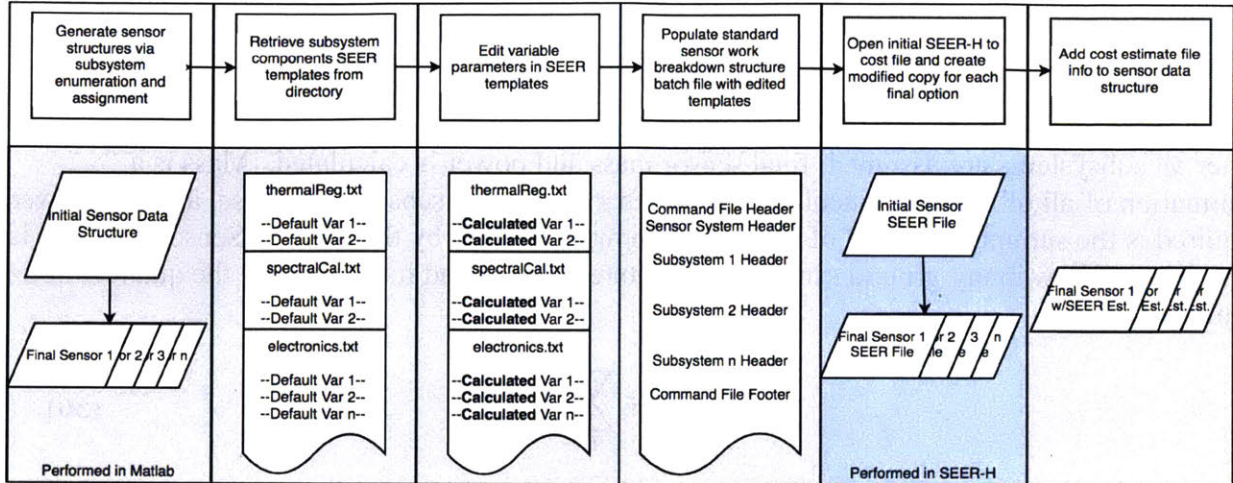


Figure 32: Steps in final sensor SEER-H estimate.

11 Assigning Compatible Satellite Busses and Launch Vehicles to Create Architectures

Following creation of the final sensor solutions, data on potential satellite bus options and compatible launch vehicles can be used to create a heuristic for evaluating the entire mission architecture. For each sensor (and combination of sensors within disaggregation schemes), solution branching occurs with each compatible bus assignment, forming a satellite. Subsequently, the “best” launch option is identified for each satellite. An alternative to this method of assignment is to use the mixed integer program discussed in Section 9.

11.1 Combining Sensors and Compatible Busses to Form Satellites

For simplicity, this model uses an algorithm to create all sensor(s)/bus combinations that are feasible, as determined by the mass and power constraints seen in Equation (58) and Equation (59) respectively. A sensor(s)/bus combination forms a satellite data structure.

$$\sum_{i \in N} (1 - \varepsilon_S) m_i \leq M_j \quad (58)$$

$$\sum_{i \in N} (1 - \varepsilon_P) p_i \leq P_j \quad (59)$$

Where:

N is the number of sensors

m_i is the mass corresponding to sensor i

ε_S is the factor of error in mass estimates from the model

M_j is the maximum allowed payload mass for bus j

p_i is the power requirement corresponding to sensor i

ε_P is the factor of error in power estimates from the model

P_j is the on-orbit average power available from bus j

Satellite bus data structures are composed of the information found Table 31. More information and constraints can be added to the model, as desired, with some additional programming.

Field	Description
Vendor	The vendor of the satellite bus.
Model	The model of the satellite bus.
Dry Mass	The mass of the satellite bus without the payload.
Max Payload Mass, M_j	The maximum payload mass the satellite can support.
Orbit Average Power, P_j	The average power generated for any given orbit.
Compatible Launch Vehicles	Launch vehicles the satellite can be launched on.
Cost	The cost of the satellite bus without sensor integration.

Table 31: Satellite bus data structure contents.

To understand how the assignment algorithm works, it is important to recall how the model generates sensors. For each orbit, the model generates sensor options which uniformly meet requirements along a focal plane disaggregation schema and sensor scan method. The solution branching that occurs for each optical form and other subsystems results in many different viable sensors. However, when there is disaggregation, at least one sensor solution from each disaggregation category is required to meet the mission's requirements. For instance, a valid solution that is complete in the thermal disaggregation scheme is an infrared collecting whiskbroom sensor (requires thermal regulation category) and VSNIR collecting pushbroom sensor (does not require thermal regulation category).

The assignment algorithm uses the disaggregation category constraint above to create the following satellite options for sensor combinations within each disaggregation schema.

1. Mass and power constraints determine which sensor(s)/bus combinations are feasible such that all sensors reside on one bus. This option is only available in cases where the disaggregated sensors (or a subset of them) fly in the same orbit(s). This logic results from the constraint that the sensors' collections of a single spot on the ground occur within some reasonable amount of time.
2. Constraints determine which sensor(s)/bus combinations are available such that combinations of sensors are disaggregated onto different busses.

The result of this algorithm is an array of satellites (sensor(s)/bus) solutions which meet mission requirements. Here a satellite solution is defined as set of satellites which meet mission requirements when flown in the proper constellation configuration. Figure 33 illustrates an example of three sensor sets which form a sensor solution. Two sets are from the thermal disaggregation scheme with one set comprised of two different pushbroom imagers (IR and VSWIR) and the other a pushbroom/whiskbroom combination. The third set is from the no-disaggregation scheme, thus only one sensor is necessary. Potential bus/sensor pairings are shown via the dotted arrows. As shown, Bus 1 can host both sensors from the first set or the sensors can be disaggregated onto Busses 2 and 3. The third sensor set is compatible with Bus options 4 and 5. Each combination of bus with sensor(s) represents a satellite solution set.

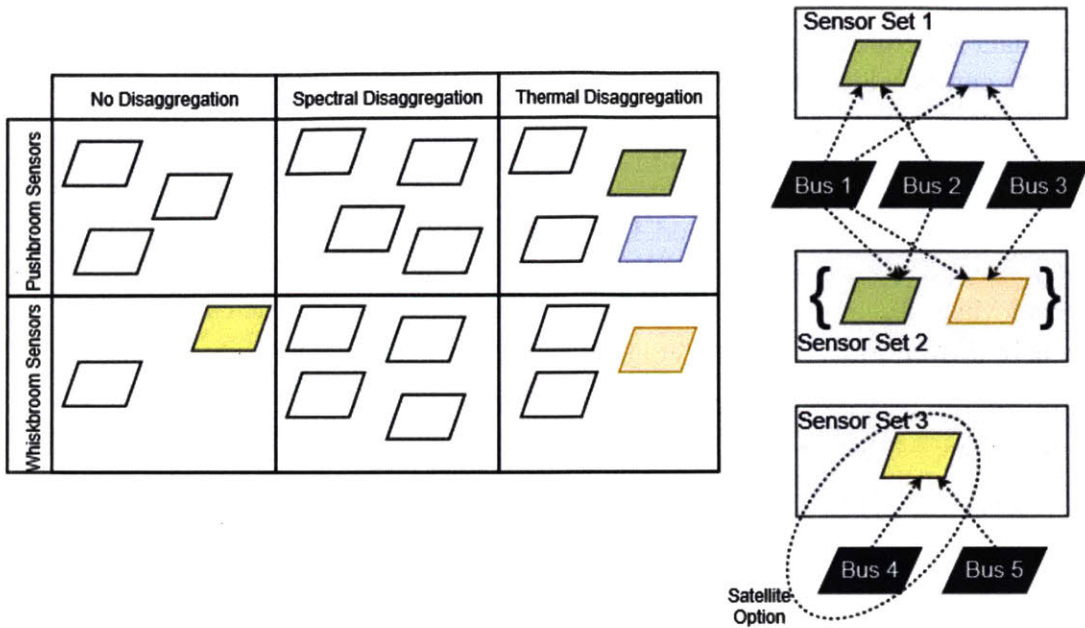


Figure 33: Sensor options based on disaggregation and sensor scan method.

Each satellite data structure is composed of the information Table 32. The mass and costs of the satellites can be calculated by adding together the respective parameters of both the sensor(s) using Equation (60) and Equation (61).

$$M_{Sat} = \sum_{i \in N} m_{S_i} + M_B \quad (60)$$

$$C_{Sat_i} = \#_{Sat_i} C_{BLV_i} \quad (61)$$

Parameter	Description	Domain, Equation, or Assignment
Sensors	The sensor data structures assigned to the bus.	Sensor data structures.
Bus	The satellite bus the sensors are assigned to	Bus data structure.
Mass	The composite mass of the satellite.	Equation (60)
Total Number of Satellites	The total number of this type of satellite needed to meet a subset of mission requirements.	All real positive integers.
Compatible Launch Vehicles	All launch vehicles which can lift this satellite to the specified orbit.	-
Best Launch Vehicle	The most affordable launch vehicle of all compatible launch vehicles.	Compatible Launch Vehicles
Satellite Launch Costs	The total cost to launch the total number of this type of satellite to orbit using the "Best Launch Vehicle"	Equation (61)

Table 32: Satellite data structures.

11.2 Assessing Launch Vehicle Options

The final step in creating an architecture solution is to assess launch vehicle options. This is accomplished via a method similar to the bus assignment algorithm, except all compatible launch vehicles are assigned to the satellite data structure as an array with most affordable launch vehicle being designated “best option.” For an assignment to be valid, the launch vehicle must be capable of delivering the satellite to the required orbit and inclination angle. A look up table is used for each launch vehicle to calculate the maximum mass a launch vehicle can lift to the specific orbit in question. If Equation (62) is true and the satellites mass is less than the maximum mass the launch vehicle can carry to orbit, then the launch vehicle is added to the compatibility array. The cost of launching a constellation is simply the summation of the number of satellites in the constellation multiplied by the cost of each satellite’s best launch option as show in Equation (63). Recommended future work includes determining the benefit and reduced costs if multiple satellites can be launched on the same vehicle. For this model, Spaceflight Industries data provides the only metric for projecting the cost of rideshare.

$$M_{Sat_i} \leq M_{LV_k} \quad (62)$$

$$C_{Launch} = \sum_{i \in N} \#_{Sat_i} C_{BLV_i} \quad (63)$$

Where:

C_{Launch} is the total launch cost of a mission

N is the number of different types of satellites in a mission.

$\#_{Sat_i}$ is the number of satellite type i in a mission

C_{BLV_i} is the cost of the best launch option for satellite type i

12 Constructing Mission Architecture Solutions

After launch vehicle options have been assessed, all solutions are output in mission architecture data structures, shown in Table 33. Key parameters are also exported to an excel file for analysis.

Parameter	Description	Domain, Equation, or Assignment
ID	Architecture ID	Any real positive integer.
Orbit	The orbit the architecture applies to.	Orbit data structure assigned to generate the solution.
Satellites	Satellites comprising the Architecture	Satellite data structures.

Table 33: Architecture data structures

Appendix F: VIIRS Requirements

		Specification										
Band No.	Driving EDR(s)	Spectral Range (um)	Horiz Sample Interval (km) (track x Scan)		Band Gain	Ltyp or Ttyp (Spec)	Lmax or Tmax	SNR or NEdT (K)	Measured SNR or NEdT (K)	SNR Margin (%)		
			Nadir	End of Scan								
Reflective Bands	VisNIR	M1	Ocean Color Aerosol	0.402 - 0.422	0.742 x 0.259	1.60 x 1.58	High Low	44.9 155	135 615	352 316	723 1327	105% 320%
		M2	Ocean Color Aerosol	0.436 - 0.454	0.742 x 0.259	1.60 x 1.58	High Low	40 146	127 687	380 409	576 1076	51.5% 163%
		M3	Ocean Color Aerosol	0.478 - 0.498	0.742 x 0.259	1.60 x 1.58	High Low	32 123	107 702	416 414	658 1055	58.2% 155%
		M4	Ocean Color Aerosol	0.545 - 0.565	0.742 x 0.259	1.60 x 1.58	High Low	21 90	78 667	362 315	558 882	54.1% 180%
		I1	Imagery EDR	0.600 - 0.680	0.371 x 0.387	0.80 x 0.789	Single	22	718	119	265	122.7%
		M5	Ocean Color Aerosol	0.662 - 0.682	0.742 x 0.259	1.60 x 1.58	High Low	10 68	59 651	242 360	360 847	49% 135%
		M6	Atmosph. Correct.	0.739 - 0.754	0.742 x 0.776	1.60 x 1.58	Single	9.6	41	199	394	98.0%
		I2	NDVI	0.846 - 0.885	0.371 x 0.387	0.80 x 0.789	Single	25	349	150	299	99.3%
	M7	Ocean Color Aerosol	0.846 - 0.885	0.742 x 0.259	1.60 x 1.58	High Low	6.4 33.4	29 349	215 340	545 899	154% 164%	
	SWMIR	M8	Cloud Particle Size	1.230 - 1.250	0.742 x 0.776	1.60 x 1.58	Single	5.4	165	74	349	371.6%
		M9	Cirrus/Cloud Cover	1.371 - 1.386	0.742 x 0.776	1.60 x 1.58	Single	6	77.1	83	247	197.6%
		I3	Binary Snow Map	1.580 - 1.640	0.371 x 0.387	0.80 x 0.789	Single	7.3	72.5	6	165	2650.0%
		M10	Snow Fraction	1.580 - 1.640	0.742 x 0.776	1.60 x 1.58	Single	7.3	71.2	342	695	103.2%
		M11	Clouds	2.225 - 2.275	0.742 x 0.776	1.60 x 1.58	Single	0.12	31.8	10	18	80.0%
		I4	Imagery Clouds	3.550 - 3.930	0.371 x 0.387	0.80 x 0.789	Single	270	353	2.5	0.4	84.0%
		M12	SST	3.660 - 3.840	0.742 x 0.776	1.60 x 1.58	Single	270	353	0.396	0.12	69.7%
Emissive Bands	LWIR	M13	SST Fires	3.973 - 4.128	0.742 x 0.259	1.60 x 1.58	High Low	300 380	343 634	0.107 0.423	0.044 --	59% --
		M14	Cloud Top Properties	8.400 - 8.700	0.742 x 0.776	1.60 x 1.58	Single	270	336	0.091	0.054	40.7%
	M15	SST	10.263 - 11.263	0.742 x 0.776	1.60 x 1.58	Single	300	343	0.07	0.028	60.0%	
	I5	Cloud Imagery	10.500 - 12.400	0.371 x 0.387	0.80 x 0.789	Single	210	340	1.5	0.41	72.7%	
M16	SST	11.538 - 12.488	0.742 x 0.776	1.60 x 1.58	Single	300	340	0.072	0.036	50.0%		

Table 34: Sensitivity specifications for the VIIRS instrument. Reprinted from Visible Infrared Imaging Radiometer Suite (VIIRS) Sensor Data Record (SDR) User's Guide Ver. 1.2, NOAA, 2013, p. 13.

Appendix G: Architecture Concept Creation Model Use Case

1 User Inputs

Requirements		
Requirement	Quantity	Units
Required Wavelengths	Enter in Sheet 2	N/A
Spatial Resolution	375	meters
Coverage Area		
Minimum Latitude	-90	degrees
Maximum Latitude	90	degrees
Minimum Longitude	-180	degrees
Maximum Longitude	180	degrees
Revisit Rate	86400	seconds
Diffraction Limited		
Wavelength	1.20E-05	meters

Focal Plane Technology
Enter in Sheet 3

Engineering Constraints		
Parameter	Quantity	Unit
Minimum Elevation Angle	20	degrees
Cross Track Overlap	1.05	Factor
Along Track Overlap	1.05	Factor
Heat Load	2	Watts

Sheet instructions:

- 1) Input mission requirements, engineering constraints, and scenario analysis in this sheet.
- 2) Under scenario analysis, toggle options for automated circular LEO analysis and disaggregation schema.
- 3) 'Sensors Input' is designed to handle any kind of orbit that is wished for consideration. Type 'STK' in the 'Estimated Planes' field of 'Sensor Inputs' table analysis if you would like STK to estimate the number of planes required for a non LEO orbit. See 'readInputSensors.m' and 'coverageAnalysis.m' for details. Otherwise, type the number of planes required, as previously calculated.
- 4) Ensure the correct directories are marked for loading SEER component files and for saving sensor SEER cost estimates generated by the model.
- 5) Ensure the spreadsheet is saved for any changes before running

Figure 34: Model inputs sheet part 1 (excluding Wavelength requirements).

Leo Constellation Analysis		
Parameter	Quantity	Unit
Run LEO Analysis	Yes	
Minimum LEO Altitude	8.00E+05	Meters
Maximum LEO Altitude	8.00E+05	Meters
Altitude Increment Step	5.00E+04	Meters
Special Orbit Type	Sun-Synchronous	N/A
Include Whiskbroom Solutions	Yes	N/A
Include Pushbroom Solutions	Yes	N/A
Combined Solutions	Yes	N/A
Disaggregation by Spectral Category (emissive vs. Reflective)	Yes	N/A
Disaggregation by Thermal Regulation of Focal Planes (required vs. unrequired)	No	N/A

Scenario Analysis

STK Scenario Save Directory:
SEER-H Save Dir: H 7.3\Projects\EO Components\thesis_case\test_case_1\
SEER-H Elements: SEER-H 7.3\Projects\EO Components\SEER Gold Files\

Include Subsystems in SEER? Yes

Sensor Inputs										
Include in Model	Sensor Type	Semi-Major Axis	Perigee Altitude	Apogee Altitude	Eccentricity	Inclination Ang	RAAN	Argument of Perigee	Initial True Anomaly	Estimated Planes Required
No	Whiskbroom	7.18E+06	8.00E+05	8.00E+05	0	98	0	0	180	4

Sensor Inputs

Figure 35: Model inputs sheet part 2 (excluding Wavelength requirements).

2 Test Case Whiskbroom Sensor Information

The screenshot displays the SEER software interface, divided into several key sections:

- Work Elements:** A hierarchical tree structure showing the breakdown of the test case. The selected path is: 1.1.1.1: Acceptance testing > 1.1.1.1.1: Payload Integration > 1.1.1.1.1.1: Integrated Optics Subsystem > 1.1.1.1.1.1.1: Telescope Assembly > 1.1.1.1.1.1.1.1: TMA - RT > 1.1.1.1.1.1.1.1.1: Motor Assembly > 1.1.1.1.1.1.1.1.2: Optics Assembly > 1.1.1.1.1.1.2: Focal Plane Subsystem > 1.1.1.1.1.2.2: HgCdTe - S/MWIR.
- Inputs:** A detailed view of the selected work element. It includes:
 - PRODUCT DESCRIPTION:** EOS Detector: HgCdTe - S/MWIR.
 - KEY TECHNICAL/PERFORMANCE PARAMETERS:**
 - Array Size (pixels): 26,988
 - Rows (pixels): 519
 - Columns (pixels): 52
 - Radiation Tolerance (rad): 4,000
 - Cutoff Wavelength (microns): 5.20
 - Dead Pixels: 0.01%
 - Pitch (microns): 18
 - MISSION DESCRIPTION:** Environment, Vehicle.
 - PROGRAM DESCRIPTION:**
 - New Design: 0.01%
 - Design Replication: 0.00%
 - Design Complexity: Nom
 - Subsystem Integration Level: Nom
- Reports:** A table showing cost estimates for various items.

Item	Estimate
Development Cost	69,873.41
Development Labor Hours	231.68
Production Cost	2,899,027.86
Total Production Units	3
APUC	966,342.62
Total Equipment Support Cost	0.00
Element Weight	0
- Charts:** A pie chart titled "HgCdTe - S/MWIR Production Cost by Activity". The chart is divided into six segments representing different cost activities: Material, Fabrication, Int and Asm, Prod Sup, Sus Eng, and Prog Mgmt (Prod). A legend on the right identifies each segment with a color-coded square.

Figure 37: Snapshot of the test case whiskbroom work breakdown structure in SEER.

Focal Plane	Wavelength (m)	Quantum Efficiency	Transmission Efficiency	Frame Rate (Hz)	Max Integration Time (s)	Single Frame Electrons	Shot Noise	Dark Noise	ADC Noise	Single Frame SNR	Samples Required	SNR Check
Si PIN	4.12E-07	8.1E-01	6.4E-01	4653	2.15E-04	23455	153	0.1	1	130	8	369
Si PIN	4.45E-07	8.5E-01	7.1E-01	4653	2.15E-04	23676	154	0.1	1	131	9	394
Si PIN	4.88E-07	9.0E-01	7.4E-01	4653	2.15E-04	25275	159	0.1	1	137	10	432
Si PIN	5.55E-07	9.4E-01	7.4E-01	4653	2.15E-04	19866	141	0.1	1	117	10	370
Si PIN	6.72E-07	9.7E-01	7.4E-01	4653	2.15E-04	11784	109	0.1	1	82	9	246
Si PIN	7.46E-07	9.8E-01	7.2E-01	4653	2.15E-04	9213	96	0.1	1	68	9	205
Si PIN	8.56E-07	9.4E-01	7.0E-01	4653	2.15E-04	17310	132	0.1	1	107	5	239
HgCdTe SMWIR	1.24E-06	9.5E-01	7.5E-01	5170	1.93E-04	8512	92	0.0	5	64	2	91
HgCdTe SMWIR	1.38E-06	9.5E-01	7.5E-01	5170	1.93E-04	7894	89	0.0	2	61	2	86
HgCdTe SMWIR	1.61E-06	9.5E-01	7.5E-01	5170	1.93E-04	44822	212	0.0	8	193	4	387
HgCdTe SMWIR	2.25E-06	9.5E-01	7.5E-01	5170	1.93E-04	7151	85	0.0	4	56	3	98
HgCdTe SMWIR	3.70E-06	9.8E-01	7.5E-01	5170	1.93E-04	4435	67	0.0	2	38	2	54
HgCdTe SMWIR	4.05E-06	9.8E-01	7.5E-01	5170	1.93E-04	33350	183	0.0	3	162	3	281
HgCdTe LWIR	8.55E-06	9.4E-01	7.5E-01	10339	9.67E-05	107125	327	1.0	6	314	3	545
HgCdTe LWIR	1.08E-05	9.7E-01	7.5E-01	10339	9.67E-05	851054	923	1.0	26	917	2	1297
HgCdTe LWIR	1.20E-05	9.6E-01	7.5E-01	10339	9.67E-05	828431	910	1.0	24	905	2	1280

Table 36: Sensitivity information for the test case whiskbroom sensor.

Work Element Name	Total Development Cost	Total Production Cost	Total Acquisition Cost	Total APUC	Risk Independent Total Cost	Risk Independent Total Cost Mean	Risk Independent Total Cost Std Dev
EO Sensor	\$5,277,898	\$21,511,321	\$26,789,218	\$7,170,440	\$26,318,971	\$27,813,064	\$1,616,655
Program Management	\$5,277,898	\$21,511,321	\$26,789,218	\$7,170,440	\$26,318,971	\$27,813,064	\$1,616,655
System Engineering	\$5,070,958	\$20,836,723	\$25,907,681	\$6,945,574	\$25,531,249	\$26,900,320	\$1,562,646
Acceptance testing	\$4,681,752	\$19,669,911	\$24,351,663	\$6,556,637	\$24,005,435	\$25,317,154	\$1,520,834

Payload Integration	\$4,495,682	\$19,669,911	\$24,165,593	\$6,556,637	\$23,794,498	\$25,110,577	\$1,509,615
Integrated Optics Subsystem	\$263,821	\$2,604,211	\$2,868,031	\$868,070	\$2,678,906	\$2,897,075	\$249,391
Telescope Assembly	\$249,505	\$2,479,470	\$2,728,976	\$826,490	\$2,553,336	\$2,765,406	\$234,510
TMA - RT	\$216,678	\$887,247	\$1,103,925	\$295,749	\$1,038,397	\$1,130,635	\$112,404
Motor Assembly	\$32,827	\$1,592,223	\$1,625,050	\$530,741	\$1,458,159	\$1,634,772	\$199,197
Optics Assembly	\$0	\$0	\$0	\$0	\$0	\$0	\$0
Focal Plane Subsystem	\$202,102	\$6,634,769	\$6,836,871	\$2,211,590	\$6,578,734	\$6,984,356	\$494,404
SiPIN	\$43,431	\$378,393	\$421,823	\$126,131	\$375,063	\$476,055	\$117,310
HgCdTe - S/MWIR	\$69,873	\$2,899,028	\$2,968,901	\$966,343	\$2,713,087	\$3,018,468	\$369,845
HgCdTe - LWIR	\$79,880	\$3,314,213	\$3,394,093	\$1,104,738	\$3,184,318	\$3,429,882	\$340,024
Thermal Regulation Subsystem	\$285,239	\$5,651,916	\$5,937,155	\$1,883,972	\$5,452,035	\$6,328,654	\$1,032,593
Cryocooler	\$113,463	\$4,841,156	\$4,954,618	\$1,613,719	\$4,296,167	\$5,171,440	\$943,414
Cryocooler Electronics	\$154,884	\$621,390	\$776,274	\$207,130	\$562,469	\$945,099	\$436,755
Instrument Electronics Subsystem	\$3,106,707	\$1,033,715	\$4,140,422	\$344,572	\$3,911,573	\$4,281,805	\$427,468
Electronics Power Supply	\$14,462	\$103,908	\$118,369	\$34,636	\$99,123	\$130,188	\$39,840
Electronics Data Processor	\$387,845	\$246,169	\$634,014	\$82,056	\$563,787	\$659,444	\$122,337
Digital Focal Plane Interface Electronics	\$1,497,682	\$246,169	\$1,743,850	\$82,056	\$1,505,383	\$1,774,036	\$317,604
Electronics Heater Component	\$75,701	\$345,552	\$421,253	\$115,184	\$273,832	\$468,719	\$249,461
Electronics Mechanisms Component	\$997,381	\$35,097	\$1,032,478	\$11,699	\$901,109	\$1,058,835	\$184,638

Calibration Subsystem	\$328,525	\$1,818,779	\$2,147,304	\$606,260	\$2,048,610	\$2,191,957	\$169,079
Solar Diffuser	\$150,012	\$614,982	\$764,993	\$204,994	\$681,006	\$771,159	\$96,212
SDSM	\$25,080	\$614,982	\$640,061	\$204,994	\$594,045	\$648,960	\$76,086
Black Body Assembly	\$103,939	\$293,600	\$397,539	\$97,867	\$362,589	\$418,263	\$75,985

Table 37: SEER-H WBS Estimate for Whiskbroom Sensor.

3 Test Case Pushbroom Sensors

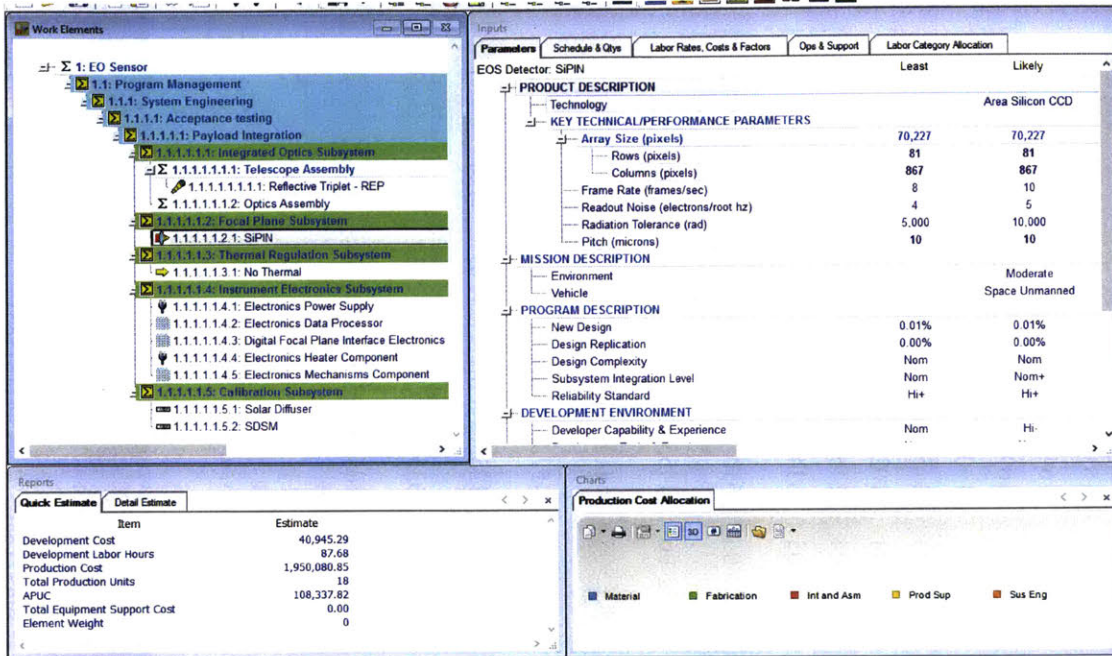


Figure 38: SEER-H estimate for the reflective bands sensor.

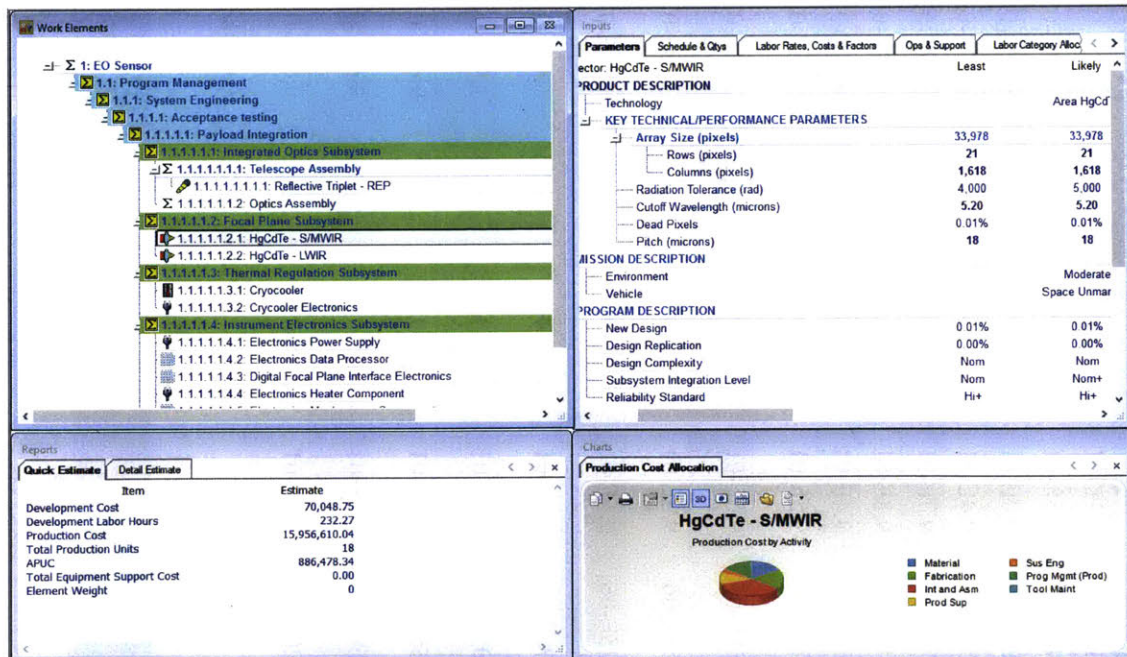


Figure 39: SEER-H estimate for the emissive bands sensor.

Focal Plane	Wavelength (m)	Quantum Efficiency	Transmission Efficiency	Frame Rate (Hz)	Max Integration Time (s)	Single Frame Electrons	Shot Noise	Dark Noise	ADC Noise	Single Frame SNR	Samples Required	SNR Check
Si PIN	4.12E-07	8.1E-01	6.4E-01	20.76	4.82E-02	2.67.E+04	163.6	0.2	1	142	8	7
Si PIN	4.45E-07	8.5E-01	7.1E-01	20.76	4.82E-02	2.70.E+04	164.3	0.2	2	142	9	8
Si PIN	4.88E-07	9.0E-01	7.4E-01	20.76	4.82E-02	2.88.E+04	169.8	0.2	2	148	10	8
Si PIN	5.55E-07	9.4E-01	7.4E-01	20.76	4.82E-02	2.27.E+04	150.5	0.2	1	128	10	9
Si PIN	6.72E-07	9.7E-01	7.4E-01	20.76	4.82E-02	1.34.E+04	115.9	0.2	1	90	9	8
Si PIN	7.46E-07	9.8E-01	7.2E-01	20.76	4.82E-02	1.05.E+04	102.5	0.2	1	75	9	7
Si PIN	8.56E-07	9.4E-01	7.0E-01	20.76	4.82E-02	1.97.E+04	140.5	0.2	2	117	5	4
HgCdTe SMWIR	1.24E-06	9.5E-01	7.5E-01	38.75	2.58E-02	4.02.E+05	634.3	0.1	216	594	2	1
HgCdTe SMWIR	1.38E-06	9.5E-01	7.5E-01	38.75	2.58E-02	3.73.E+05	610.8	0.1	84	598	2	1
HgCdTe SMWIR	1.61E-06	9.5E-01	7.5E-01	38.75	2.58E-02	2.12.E+06	1455.5	0.1	364	1409	4	1
HgCdTe SMWIR	2.25E-06	9.5E-01	7.5E-01	38.75	1.93E-04	3.38.E+05	581.4	0.1	189	546	3	1
HgCdTe SMWIR	3.70E-06	9.8E-01	7.5E-01	38.75	1.93E-04	2.10.E+05	457.8	0.1	109	437	2	1
HgCdTe SMWIR	4.05E-06	9.8E-01	7.5E-01	38.75	1.93E-04	1.58.E+06	1255.5	0.1	123	1246	3	1
HgCdTe LWIR	8.55E-06	9.4E-01	7.5E-01	38.75	9.67E-05	4.05.E+07	6364.5	16.2	2439	5942	3	1
HgCdTe LWIR	1.08E-05	9.7E-01	7.5E-01	38.75	9.67E-05	1.14.E+08	10654.6	9.6	3524	10115	2	1
HgCdTe LWIR	1.20E-05	9.6E-01	7.5E-01	38.75	9.67E-05	828431	11113.4	10.1	3524	10593	2	1

Table 38: Sensitivity results for the test case pushbroom sensors.

Work Element Name	Total Development Cost	Total Production Cost	Total Acquisition Cost	Total APUC	Risk Independent Total Cost	Risk Independent Total Cost Mean	Risk Independent Total Cost Std Dev
EO Sensor	\$4,196,200	\$8,728,623	\$12,924,824	\$1,454,771	\$12,654,991	\$13,675,317	\$1,118,912
Program Management	\$4,196,200	\$8,728,623	\$12,924,824	\$1,454,771	\$12,654,991	\$13,675,317	\$1,118,912

System Engineering	\$4,038,701	\$8,433,466	\$12,472,167	\$1,405,578	\$12,236,526	\$13,199,573	\$1,083,249
Acceptance testing	\$3,742,483	\$7,922,948	\$11,665,431	\$1,320,491	\$11,469,809	\$12,365,895	\$1,020,264
Payload Integration	\$3,600,868	\$7,922,948	\$11,523,816	\$1,320,491	\$11,319,968	\$12,208,062	\$1,017,219
Integrated Optics Subsystem	\$9,430	\$212,709	\$222,139	\$35,451	\$209,809	\$229,428	\$23,117
Telescope Assembly	\$8,930	\$204,629	\$213,559	\$34,105	\$201,998	\$221,167	\$22,324
Reflective Triplet - REP	\$8,930	\$204,629	\$213,559	\$11,368	\$201,998	\$221,167	\$22,324
Optics Assembly	\$0	\$0	\$0	\$0	\$0	\$0	\$0
Focal Plane Subsystem	\$42,876	\$1,961,383	\$2,004,259	\$326,897	\$1,831,221	\$2,214,131	\$550,008
SiPIN	\$40,945	\$1,950,081	\$1,991,026	\$108,338	\$1,817,948	\$2,197,699	\$546,774
Thermal Regulation Subsystem	\$0	\$0	\$0	\$0	\$0	\$0	\$0
No Thermal	\$0	\$0	\$0	\$0	\$0	\$0	\$0
Instrument Electronics Subsystem	\$3,106,707	\$2,014,263	\$5,120,970	\$335,710	\$4,926,900	\$5,436,005	\$692,474
Electronics Power Supply	\$14,462	\$196,853	\$211,315	\$32,809	\$167,744	\$217,026	\$63,608
Electronics Data Processor	\$387,845	\$492,337	\$880,182	\$82,056	\$832,735	\$947,252	\$146,572
Digital Focal Plane Interface Electronics	\$1,497,682	\$492,337	\$1,990,019	\$82,056	\$1,784,604	\$2,034,479	\$312,179
Electronics Heater Component	\$75,701	\$658,504	\$734,205	\$109,751	\$473,422	\$891,000	\$518,300
Electronics Mechanisms Component	\$997,381	\$70,194	\$1,067,575	\$11,699	\$940,363	\$1,104,519	\$201,598
Calibration Subsystem	\$206,460	\$2,891,679	\$3,098,139	\$481,946	\$2,885,485	\$3,137,805	\$277,195
Solar Diffuser	\$150,012	\$1,209,408	\$1,359,420	\$201,568	\$1,198,572	\$1,371,546	\$171,520
SDSM	\$25,080	\$1,209,408	\$1,234,488	\$201,568	\$1,105,136	\$1,252,011	\$158,029

Table 39: SEER-H WBS Estimate for Pushbroom Reflective Sensor.

Outline Number	Type	Work Element Name	Total Development Cost	Total Production Cost	Total Acquisition Cost	Total APUC	Risk Independent Total Cost	Risk Independent Total Cost Mean	Risk Independent Total Cost Std Dev
1	PROJECT	EO Sensor	\$4,777,433	\$60,248,688	\$65,026,122	\$10,041,448	\$63,879,117	\$66,962,862	\$3,638,689
1.1	ROLLUP	Program Management	\$4,777,433	\$60,248,688	\$65,026,122	\$10,041,448	\$63,879,117	\$66,962,862	\$3,638,689
1.1.1	ROLLUP	System Engineering	\$4,592,731	\$58,587,662	\$63,180,393	\$9,764,610	\$62,105,699	\$65,071,469	\$3,588,688

1.1.1.1	ROLLUP	Acceptance testing	\$4,245,349	\$55,714,681	\$59,960,030	\$9,285,780	\$58,979,434	\$61,817,838	\$3,479,303
1.1.1.1.1	ROLLUP	Payload Integration	\$4,079,274	\$55,714,681	\$59,793,955	\$9,285,780	\$58,861,873	\$61,631,956	\$3,479,886
1.1.1.1.1.1	ROLLUP	Integrated Optics Subsystem	\$132,199	\$2,981,913	\$3,114,112	\$496,985	\$2,914,695	\$3,171,590	\$293,049
1.1.1.1.1.1.1	ROLLUP	Telescope Assembly	\$125,186	\$2,868,651	\$2,993,836	\$478,108	\$2,808,319	\$3,057,369	\$285,928
1.1.1.1.1.1.1.1	EOS Optical Device	Reflective Triplet - REP	\$125,186	\$2,868,651	\$2,993,836	\$159,369	\$2,808,319	\$3,057,369	\$285,928
1.1.1.1.1.1.1.1.1	ROLLUP	Optics Assembly	\$0	\$0	\$0	\$0	\$0	\$0	\$0
1.1.1.1.1.1.1.1.1.1	ROLLUP	Focal Plane Subsystem	\$157,012	\$34,410,712	\$34,567,724	\$5,735,119	\$33,469,001	\$35,672,389	\$2,744,094
1.1.1.1.1.1.1.1.1.1.1	EOS Detector	HgCdTe - S/MWIR	\$70,049	\$15,956,610	\$16,026,659	\$886,478	\$14,608,239	\$16,313,895	\$1,932,214
1.1.1.1.1.1.1.1.1.1.1.1	EOS Detector	HgCdTe - LWIR	\$80,076	\$18,240,832	\$18,320,909	\$1,013,380	\$17,282,048	\$19,097,342	\$2,060,614
1.1.1.1.1.1.1.1.1.1.1.1.1	ROLLUP	Thermal Regulation Subsystem	\$285,239	\$10,925,268	\$11,210,507	\$1,820,878	\$9,802,889	\$11,196,892	\$1,598,121
1.1.1.1.1.1.1.1.1.1.1.1.1.1	EOS Cooler	Cryocooler	\$113,463	\$9,396,398	\$9,509,861	\$1,566,066	\$8,306,888	\$9,446,261	\$1,439,702
1.1.1.1.1.1.1.1.1.1.1.1.1.1.1	Electronics	Cryocooler Electronics	\$154,884	\$1,185,837	\$1,340,721	\$197,640	\$903,145	\$1,404,991	\$639,455
1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1	ROLLUP	Instrument Electronics Subsystem	\$3,106,707	\$2,014,263	\$5,120,970	\$335,710	\$4,798,215	\$5,402,165	\$693,374
1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1	Electronics	Electronics Power Supply	\$14,462	\$196,853	\$211,315	\$32,809	\$171,347	\$215,503	\$56,299
1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1	Field Programmable Gate Arrays (FPGA)	Electronics Data Processor	\$387,845	\$492,337	\$880,182	\$82,056	\$821,974	\$930,487	\$136,775
1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1	Field Programmable Gate Arrays (FPGA)	Digital Focal Plane Interface Electronics	\$1,497,682	\$492,337	\$1,990,019	\$82,056	\$1,734,948	\$2,061,378	\$374,036
1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1	Electronics	Electronics Heater Component	\$75,701	\$658,504	\$734,205	\$109,751	\$507,795	\$834,100	\$424,820
1.1	Field Programmable Gate Arrays (FPGA)	Electronics Mechanisms Component	\$997,381	\$70,194	\$1,067,575	\$11,699	\$925,850	\$1,119,539	\$205,467
1.1	ROLLUP	Calibration Subsystem	\$122,065	\$638,948	\$761,013	\$106,491	\$686,630	\$804,063	\$147,787
1.1	EOS Calibrator	Black Body Assembly	\$103,939	\$569,462	\$673,400	\$94,910	\$600,132	\$711,009	\$132,900

Table 40: SEER-H WBS Estimate for Pushbroom Emissive Sensor.

4 SEER Command Template Example

```

WBSCreate SiPIN EOS Detector 6
HApplicationKBase Area Si CCD 1
HPlatformKBase Space-Unmanned 1
HOSDescriptionKBase
HAcquisitionCategoryKBase Build To Print 1
HStandardKBase Space - Science, Command & Cntrl 1
HClassKBase
PRODUCT DESCRIPTION
Technology Area Silicon CCD
KEY TECHNICAL/PERFORMANCE PARAMETERS
Array Size (pixels)
Rows (pixels) 768 1024 1280
Columns (pixels) 768 1024 1280
Frame Rate (frames/sec) 8 10 12
Readout Noise (electrons/root hz) 4 5 6
Radiation Tolerance (rad) 5000 10000 15000
Pitch 10 10 10
MISSION DESCRIPTION
Environment Moderate
Vehicle Space Unmanned
PROGRAM DESCRIPTION
New Design 0.01% 0.01% 0.01%
Design Replication 0.00% 0.00% 0.00%
Design Complexity Nom Nom Nom
Subsystem Integration Level Nom Nom+ Hi
Reliability Standard Hi+ Hi+ VHi-
DEVELOPMENT ENVIRONMENT
Developer Capability & Experience Nom Hi- VHi-
Development Tools & Practices Nom Nom Nom
Requirements Volatility Low Low+ Nom
PRODUCTION ENVIRONMENT
Production Experience Nom Hi- VHi-
Production Tools & Practices VLo VLo VLo+
PROBABILITY 50.00%
ENGINEERING INPUTS (Optional)
Use In System Level Calculation YES
Include in Subsystem Total YES
Weight0.00 0.00 0.00
PROGRAM SCHEDULE
Required Development Sched (Mos) 0.00 0
Start Date for Development 7/26/2017 YES
Preliminary Design Review Readiness Date 8/02/2017 YES
Detail Design Start Date 8/03/2017 YES

```

Critical Design Review Readiness Date	8/11/2017	YES
System Validation Start Date	8/12/2017	YES
System Validation Review Readiness Date	8/31/2017	YES
Start Date for Production	9/01/2017	
Quantity Per Next Higher Element	1.00	
Prototype Quantity	0.00	YES
Development Labor Profile	Rayleigh	
Left Truncation	0.00%	
Right Truncation	100.00%	
Production Learning Curve	95.00%	
Prior Production Units	0	0 0
Stop Learning Quantity	0	
PRODUCTION QUANTITY PER YEAR		
Production Quantity Year 1	12	YES
Production Quantity Year 2	12	YES
Production Quantity Year 3	0	YES
ENGINEERING HOURLY RATE	179.96	YES
Development Management Hourly Rate	179.96	
Systems Engineering Hourly Rate	179.96	
Design Engineering Hourly Rate	179.96	
Prototype Engineering Hourly Rate	179.96	
Test Engineering Hourly Rate	179.96	
Tooling Engineering Hourly Rate	179.96	
Development Support Hourly Rate	179.96	
MANUFACTURING HOURLY RATE	156.53	YES
Production Management Hourly Rate	156.53	
Fabrication Hourly Rate	156.53	
Assembly Hourly Rate	156.53	
Test and QA Hourly Rate	156.53	
Sustaining Engineering Hourly Rate	156.53	
Tool Maintenance Hourly Rate	156.53	
Production Support Hourly Rate	156.53	
ECONOMIC FACTORS (Optional)		
Development Labor Cost Wraps/Fee	0.00%	
Development Material Cost Wraps/Fee	0.00%	
Production Labor Cost Wraps/Fee	0.00%	
Production Material Cost Wraps/Fee	0.00%	
PURCHASED ITEMS		
Percentage of Item Purchased	0.01%	
Prototype Unit Purchase Cost	0.00	
Production Unit Purchase Cost	0.00	
OPERATIONS & SUPPORT		
CALIBRATION INPUTS		
Actual Development Labor Hours	0.00	
Actual Development Material Cost	0.00	
Actual Development Schedule (Mos)	0.00	

Actual Production Labor Hours	0.00
Actual Production Material Cost	0.00
OPERATIONS & SUPPORT CALIBRATION FACTORS	
Level 1 Support Training Factor	0.00%
Level 2 Support Training Factor	0.00%
Level 3 Support Training Factor	0.00%
Level 1 Parts Factor	0.00%
Level 2 Parts Factor	0.00%
Level 3 Parts Factor	0.00%
Condemnation Labor Factor	0.00%
Spares Inventory Management Factor	2.00
Data Management Factor	0.75
Initial MTBF	0
MTBF Growth Factor	0.00
CLASS ADJUSTMENTS	
DEVELOPMENT LABOR(C)	0.00%
Design(C)	0.00%
Prototype Hardware Labor(C)	0.00%
Engineering Test(C)	0.00%
Integration and Test(C)	0.00%
Systems Engineering(C)	0.00%
Program Management (Dev)(C)	0.00%
Engineering Data(C)	0.00%
Management Data(C)	0.00%
Support Data(C)	0.00%
Peculiar Support Equipment Labor(C)	0.00%
Tooling Labor(C)	0.00%
DEVELOPMENT MATERIAL(C)	0.00%
Prototype Hardware Material(C)	0.00%
Peculiar Support Equipment Material(C)	0.00%
Tooling Material(C)	0.00%
PRODUCTION LABOR(C)	0.00%
Fabrication(C)	0.00%
Integration and Assembly(C)	0.00%
Production Support(C)	0.00%
Sustaining Engineering(C)	0.00%
Program Management (Prod)(C)	0.00%
Tooling Maintenance(C)	0.00%
Material(C)	0.00%
Development Schedule(C)	0.00%
Mature MTBF(C)	0.00%
Operational Hours To Maturity(C)	0.00%
Mean Time To Repair(C)	0.00%
APPLICATION ADJUSTMENTS	
DEVELOPMENT LABOR(AP)	0.00%
Design(AP)	0.00%

Prototype Hardware Labor(AP)	0.00%
Engineering Test(AP)	0.00%
Integration and Test(AP)	0.00%
Systems Engineering(AP)	0.00%
Program Management (Dev)(AP)	0.00%
Engineering Data(AP)	0.00%
Management Data(AP)	0.00%
Support Data(AP)	0.00%
Peculiar Support Equipment Labor(AP)	0.00%
Tooling Labor(AP)	0.00%
DEVELOPMENT MATERIAL(AP)	0.00%
Prototype Hardware Material(AP)	0.00%
Peculiar Support Equipment Material(AP)	0.00%
Tooling Material(AP)	0.00%
PRODUCTION LABOR(AP)	0.00%
Fabrication(AP)	0.00%
Integration and Assembly(AP)	0.00%
Production Support(AP)	0.00%
Sustaining Engineering(AP)	0.00%
Program Management (Prod)(AP)	0.00%
Tooling Maintenance(AP)	0.00%
Material(AP)	0.00%
Development Schedule(AP)	0.00%
Mature MTBF(AP)	0.00%
Operational Hours To Maturity(AP)	0.00%
Mean Time To Repair(AP)	0.00%
ACQUISITION CATEGORY ADJUSTMENTS	
DEVELOPMENT LABOR(A)	0.00%
Design(A)	-75.00%
Prototype Hardware Labor(A)	-20.00%
Engineering Test(A)	-90.00%
Integration and Test(A)	-85.00%
Systems Engineering(A)	-75.00%
Program Management (Dev)(A)	-75.00%
Engineering Data(A)	-80.00%
Management Data(A)	-80.00%
Support Data(A)	-80.00%
Peculiar Support Equipment Labor(A)	0.01%
Tooling Labor(A)	-100.00%
DEVELOPMENT MATERIAL(A)	0.00%
Prototype Hardware Material(A)	0.01%
Peculiar Support Equipment Material(A)	0.01%
Tooling Material(A)	-100.00%
PRODUCTION LABOR(A)	0.00%
Fabrication(A)	0.01%
Integration and Assembly(A)	0.01%

Production Support(A)	0.01%		
Sustaining Engineering(A)	0.01%		
Program Management (Prod)(A)	0.01%		
Tooling Maintenance(A)	0.01%		
Material(A)	0.01%		
Development Schedule(A)	0.01%		
Mature MTBF(A)	0.00%		
Operational Hours To Maturity(A)	0.00%		
Mean Time To Repair(A)	0.00%		
STANDARD ADJUSTMENTS			
DEVELOPMENT LABOR(S)	0.00%		
Design(S)	0.01%		
Prototype Hardware Labor(S)	0.01%		
Engineering Test(S)	-10.00%		
Integration and Test(S)	-5.00%		
Systems Engineering(S)	0.01%		
Program Management (Dev)(S)	0.01%		
Engineering Data(S)	-20.00%		
Management Data(S)	-40.00%		
Support Data(S)	-10.00%		
Peculiar Support Equipment Labor(S)	0.01%		
Tooling Labor(S)	0.01%		
DEVELOPMENT MATERIAL(S)	0.00%		
Prototype Hardware Material(S)	0.01%		
Peculiar Support Equipment Material(S)	0.01%		
Tooling Material(S)	0.01%		
PRODUCTION LABOR(S)	0.00%		
Fabrication(S)	0.01%		
Integration and Assembly(S)	-2.50%		
Production Support(S)	-5.00%		
Sustaining Engineering(S)	-5.00%		
Program Management (Prod)(S)	-5.00%		
Tooling Maintenance(S)	0.01%		
Material(S)	0.01%		
Development Schedule(S)	0.01%		
Mature MTBF(S)	0.00%		
Operational Hours To Maturity(S)	0.00%		
Mean Time To Repair(S)	0.00%		
SUBSYSTEM DEV PHASE LABOR ALLOCATION			
Design(L)	20.00%	42.00%	38.00%
Prototype Hardware Labor(L)	10.00%	81.00%	9.00%
Engineering Test(L)	12.00%	37.00%	51.00%
Integration and Test(L)	10.00%	81.00%	9.00%
Systems Engineering(L)	42.00%	36.00%	22.00%
Program Management (Dev)(L)	33.00%	38.00%	29.00%
Engineering Data(L)	12.00%	46.00%	42.00%

Management Data(L)	36.00%	42.00%	22.00%				
Support Data(L)	33.00%	33.00%	34.00%				
Peculiar Support Equipment Labor(L)		10.00%	81.00%	9.00%			
Tooling Labor(L)	0.01%	40.99%	59.00%				
SUBSYSTEM DEV PHASE MATERIAL ALLOCATION							
Prototype Hardware Material(M)	10.00%	81.00%	9.00%				
Peculiar Support Equipment Material(M)	10.00%	81.00%	9.00%				
Tooling Material(M)	0.01%	40.99%	59.00%				
LABOR ALLOCATION BY ACTIVITY (DEV)							
Design	5.00%	3.00%	68.00%	3.00%	8.00%	5.00%	8.00%
Prototype Hardware	2.00%	0.00%	20.00%	60.00%	10.00%	3.00%	5.00%
Engineering Test	5.00%	10.00%	5.00%	20.00%	5.00%	55.00%	0.00%
Integration and Test	10.00%	40.00%	10.00%	5.00%	5.00%	30.00%	0.00%
Systems Engineering	5.00%	60.00%	5.00%	5.00%	5.00%	15.00%	5.00%
Program Management (Dev)	90.00%	0.00%	0.00%	0.00%	0.00%	0.00%	10.00%
Engineering Data	18.00%	8.00%	20.00%	4.00%	5.00%	25.00%	20.00%
Management Data	50.00%	0.00%	15.00%	0.00%	0.00%	0.00%	35.00%
Support Data	20.00%	15.00%	30.00%	0.00%	0.00%	10.00%	25.00%
Peculiar Support Equipment	5.00%	15.00%	10.00%	15.00%	45.00%	5.00%	5.00%
Tooling	2.00%	8.00%	5.00%	15.00%	40.00%	25.00%	5.00%
LABOR ALLOCATION BY ACTIVITY (PROD)							
Fabrication	3.00%	70.00%	0.00%	10.00%	6.00%	8.00%	3.00%
Integration and Assembly	5.00%	0.00%	60.00%	10.00%	15.00%	5.00%	5.00%
Production Support	30.00%	0.00%	0.00%	5.00%	8.00%	15.00%	42.00%
Sustaining Engineering	10.00%	5.00%	5.00%	10.00%	62.00%	3.00%	5.00%
Program Management (Prod)	80.00%	0.00%	0.00%	5.00%	5.00%	0.00%	10.00%
Tooling Maintenance	15.00%	15.00%	5.00%	5.00%	12.00%	45.00%	3.00%

Table 41: SEER command file template for the Si PIN focal plane component.

Appendix H: HyspIRI Example for Satellite Network Mixed Integer Linear Program

1 HyspIRI Example Decoder

Object	Object Type	Location Node	Child Index	Network Node	Data Capacity (Mbit)	Data Generated (Mbit/s)	Power(W)	Mass(M)	Cost(K) + [LCost(K)]	Fixed	Dependencies	Mutual Exclusion
s_0	sensor	l_0	0	0	0	300	-44	-55	50000	0	-	-
s_0	sensor	l_1	0	1	0	300	-44	-55	50000	0	-	-
s_1	sensor	l_0	1	2	0	24	-60	-103	50000	0	-	-
s_1	sensor	l_1	1	3	0	24	-60	-103	50000	0	-	-
b_0	bus	l_0	0	4	16000	0	85	65	16500+[5940]	0	-	-
b_1	bus	l_1	1	5	16000	0	140	150	23500+[7950]	0	-	-
g_1	gs	l_2	0	6	Inf	0	Inf	Inf	0	1	-	-
c_0	comm	l_0	0	7	0	0	0	0	0	0	b_0	-
c_1	comm	l_1	1	8	0	0	0	0	0	0	b_1	-
c_2	comm	l_0	2	9	0	0	5	5	200	0	-	c_3
c_3	comm	l_1	3	10	0	0	5	5	200	0	-	c_2
c_4	comm	l_1	4	11	0	0	20	10	700	0	-	c_1
c_5	comm	l_2	5	12	0	0	20	10	700	1	-	-

Table 42: HyspIRI example decoder

2 Network Model JSON Example

```
1  [
2    {
3      "orbit_params": {
4        "propagation_method": "matlab_delkep",
5        "a_km": 7004,
6        "e": 0,
7        "i_deg": 98,
8        "RAAN_deg": 0,
9        "arg_per_deg": 0,
10       "M": 0
11     },
12
13     "sensors": [
14       {
15         "name": "HYSPIRI VSWIR",
16         "power_required": 41,
17         "mass": 55,
18         "data_rate": 0.120,
19         "cost": 50000
20       },
21
22       {
23         "name": "HYSPIRI TIR",
24         "power_required": 60,
25         "mass": 103,
26         "data_rate": 0.024,
27         "cost": 50000
28       }
29     ],
30
31     "constraints": {
32       "allow_disaggregation": "yes",
33       "fixed": "no"
34     }
35   }
36 ]
37
38
39 ]
40
```

Table 43: Mission sensors JSON input file for network model.

3 HypsIRI Example Accesses

Predecessor Node, v_i	Successor Node, v_j	Start of Access	End of Access	Time Sample(s)	Data Rate
0	4	0	200	[0,100]	[300,300]
0	7	0	200	[0,100]	[2,2]
0	9	0	200	[0,100]	[300,300]
1	5	0	200	[0,100]	[300,300]
1	8	0	200	[0,100]	[105,105]
1	10	0	200	[0,100]	[300,300]
1	11	0	200	[0,100]	[400,400]
2	4	0	200	[0,100]	[24,24]
2	7	0	200	[0,100]	[2,2]
2	9	0	200	[0,100]	[24,24]
3	5	0	200	[0,100]	[24,24]
3	8	0	200	[0,100]	[24,24]
3	10	0	200	[0,100]	[24,24]
3	11	0	200	[0,100]	[24,24]
4	7	0	200	[0,100]	[2,2]
4	9	0	200	[0,100]	[300,300]
5	8	0	200	[0,100]	[105,105]
5	10	0	200	[0,100]	[300,300]
5	11	0	200	[0,100]	[400,400]
7	4	0	200	[0,100]	[2,2]
7	9	0	200	[0,100]	[2,2]
8	5	0	200	[0,100]	[105,105]
8	10	0	200	[0,100]	[105,105]
9	4	0	200	[0,100]	[Inf, Inf]
9	7	0	200	[0,100]	[2,2]
10	5	0	200	[0,100]	[Inf, Inf]
10	8	0	200	[0,100]	[105,105]
10	11	0	200	[0,100]	[300,300]
11	5	0	200	[0,100]	[Inf, Inf]
11	10	0	200	[0,100]	[300,300]
12	6	0	200	[0,100]	[800,800]
8	12	100	200	[100]	[105]
9	10	0	200	[0,100]	[300,300]
10	9	0	200	[0,100]	[300,300]
11	12	100	200	[100]	[400]
12	8	100	200	[100]	[105]
12	11	100	200	[100]	[400]

Table 44: HypsIRI example access data.

Appendix I: Satellite Network Mixed Integer Linear Program General Form

The network problem's general form is best explained by first reviewing the sets used in the model, then the decision variables, followed by a review of the constraints, and then objective functions.

1 Sets

The number of unique mission sensors forms an index set: I_S

The location nodes form an index set: L

For every index in L , there exists a set of child objects. For instance, V_l represents all of the child object network nodes that can be assigned to $l \in L$

$$\exists V_l \forall l \in L \quad (64)$$

The sub-problems form an index set: P

The network node set is composed of bus, mission sensor, communications, and ground station subsets.

$$V = S \cup B \cup C \cup G \quad (65)$$

The complete edge set is the union of all edge subsets previously discussed.

$$E = E_p \cup E_B \cup E_{sink} \cup E_{source} \quad (66)$$

For all network nodes, $v \in V$, there exists the following sets:

$E_{pv_{pred}}$ contains all predecessor edges to node v in problem p where a predecessor edge is defined as:

$$(p_{pred} \in P, v_{pred} \in V, p, v) \in E_{pv_{pred}} \subset E \quad (67)$$

$E_{pv_{succ}}$ contains all successor edges to v in problem p where a successor edge is defined as:

$$(p, v, p_{succ} \in P, v_{succ} \in V) \in E_{pv_{succ}} \subset E \quad (68)$$

E_{Av} contains all edges, that if used, imply that network node v must be present in the network.

$$\begin{aligned} & (p_{pred} \in P, v_{pred} \in V, p_{succ} \in P, v) \\ & \cup (p_{pred} \in P, v, p_{succ} \in P, v_{succ} \in V) \in E_{Av} \subset E \end{aligned} \quad (69)$$

D_v contains all dependencies for node v as indicated by the decoder.

$$D_v \subset V \quad (70)$$

2 Decision Variables

There is a decision variable for whether the network node is present in the network:

$$\exists X_{Nv} \forall v \in V: X_{Nv} = x \in \{0,1\} \quad (71)$$

There is a decision variable for how much data is transfer across each edge in the graph:

$$\exists Y_e \forall e \in E: Y_e = x \in \{\mathbb{R} | x \geq 0\} \quad (72)$$

There is a decision variable for the maximum amount of data that a bus can store:

$$\exists Z_{Dv} \forall v \in B: Z_{Dv} = x \in \{\mathbb{R} | x \geq 0\} \quad (73)$$

There is a decision variable for how much data is stored on the satellite bus at the end of the simulation:

$$\exists Z_{Ev} \forall v \in B: Z_{Ev} = x \in \{\mathbb{R} | x > 0\} \quad (74)$$

There is a decision variable for the maximum amount of data stored on the satellite bus throughout the simulation:

$$\exists Z_{Mv} \forall v \in B: Z_{Mv} = x \in \{\mathbb{R} | x \geq 0\} \text{ s. t.} \quad (75)$$

$$Z_{Dv} - Z_{Mv} \leq 0 \quad (76)$$

$$Z_{Ev} - Z_{Mv} \leq 0 \quad (77)$$

3 Data Flow Constraints:

For all network node edges, the data volume over the edge must be less than the maximum data volume capacity:

$$\forall e \in E_n: Y_{e_n} - d_n \leq 0 \quad (78)$$

For all bus data storage problem transfer edges, the data volume transferred must be less than the bus data storage capacity:

$$\forall e \in E_B: Y_{e_b} - Z_{Db} \leq 0 \quad (79)$$

For all source to sensor edges, the edge must transmit the data being generated

$$\forall e \in E_S: Y_{e_{source}} - d_{source} = 0 \quad (80)$$

The sum of all data sent to the sink plus the data stored on busses at the end of the simulation must equal the sum of all data into the network.

$$\sum_{e \in E_{sink}} Y_e + \sum_{v \in B} Z_{Ev} - \sum_{e \in E_{source}} Y_e = 0 \quad (81)$$

4 Flow Balance Constraints:

For each network node $v \in V$, in each sub-problem $p \in P$, the model creates a predecessor set, $E_{pv_{pred}}$, and successor set, $E_{pv_{succ}}$ of edges. The amount of data coming into every node in every problem must equal the data exiting every node in every problem except for the last sub-problem P_{last} . During the last problem, it is necessary to subtract ending data storage on the bus Z_{Ev} so that balances hold. Also, recall that the source and sink nodes are not included in V .

$$\forall p \in (P - P_{last}), \forall v \in V: \sum_{e \in E_{pv_{pred}}} Y_e - \sum_{e \in E_{pv_{succ}}} Y_e = 0 \quad (82)$$

$$\forall v \in S \cup C \cup G: \sum_{e \in E_{P_{last}v_{pred}}} Y_e - \sum_{e \in E_{P_{last}v_{succ}}} Y_e = 0 \quad (83)$$

$$\forall v \in B: \sum_{e \in E_{P_{last}v_{pred}}} Y_e - \sum_{e \in E_{P_{last}v_{succ}}} Y_e - Z_{Ev} = 0 \quad (84)$$

5 Assignment Constraints:

Anytime data is sent to or from a network node, the network node must be present. To enforce this constraint, the model creates a set of all edges where the network node is present either as a successor or predecessor, E_{Av} .

$$\exists E_{Av} \forall v \in V: \sum_{e \in E_{Av}} Y_e - M * X_{Nv} \leq 0 \mid M \gg 0 \quad (85)$$

6 Dependencies Constraints:

For all network nodes, there exists a set of dependencies as determined by the decoder data frame. The network node can only be present if all of its dependencies are present.

$$\exists D_v \forall v \in V: \sum_{u \in D_v} X_{Nu} - M * X_{Nv} \leq 0 \mid M \gg 0 \quad (86)$$

7 Fixed Architecture Constraints:

For all network nodes there exists a binary value, F_v , which is determined by the decoder data frame for whether the network node must be present:

$$\exists F_v \forall v \in V: F_v - X_{Nv} \leq 0 \quad (87)$$

8 Power Constraints:

For all location nodes, the sum of the power for each child object must be greater than or equal to zero. Recall, that power supplies are positive and power draws are negative.

$$\forall l \in L: \sum_{v \in V_l} W_v X_{Nv} \geq 0 \quad (88)$$

9 Mass Constraints:

For all location nodes, the sum of the power for each child object must be greater than or equal to zero. Recall, that mass “supply” by a bus is positive and mass of comm and sensor objects are negative.

$$\exists V_l \forall l \in L: \sum_{v \in V_l \subset V} M_v X_{Nv} \geq 0 \quad (89)$$

10 Sensor Assignment Constraints:

Each mission sensor must be assigned once. For ever value in the unique sensor index, there exists a set of sensor assignments.

$$\exists V_{Si} \forall i \in I_S: \sum_{v \in V_{Si}} X_{Nv} = 1 \quad (90)$$

11 Max Bus Constraints:

At each orbital location, there can only be at most one bus.

$$\exists V_{Bl} \forall l \in L_O: \sum_{v \in V_{Bl}} X_{Nv} \leq 1 \quad (91)$$

12 Objective Functions:

The two objective functions discussed in this thesis are (1) maximizing the amount of data transmitted to ground and (2) minimizing the total architectures cost. Minimizing total cost will also require the use of an additional constraint.

13 Maximizing Data to Ground -Minimizing the Maximum Data Stored on Busses:

Maximizing the amount of data sent to the ground station as efficiently as possible is equivalent to minimizing the maximum amount of data stored on satellite busses during the simulation. This logically follows from the continuity data flow constraint that all data into the network via the source must equal all data out via the sink less data stored on busses at the end of the simulation. Thus, this objective function is simply:

$$\min \sum_{v \in B} Z_{Mv} \quad (92)$$

14 Minimizing Total Architecture Cost:

The objective for minimizing the total architecture cost is presented in Equation (93). The first term is the sum of the cost for all child objects present in the final solution. The additional term in the objective function penalizes storing data in excess of a bus's stock data storage capacity. It assumes that data storage can be "purchased" if necessary for a cost of U_v per Mbit. Note that Z_{Mv} is the maximum data over the bus's capacity that is stored at the bus during the simulation or at the end of the simulation. Essentially the second term balances sending as much data to ground as possible with the cost of doing so.

$$\min \sum_{v \in V} c(v)X_{Nv} + \sum_{v \in B} U_v(Z_{Mv} - cap_v) \quad (93)$$

There are two notable optional constraints for this objective which may be useful but also may result in the problem having no solution. These constraints are:

- Imposing a maximum amount of data that can be stored on all buses at the end of the simulation, and thus imposing a minimum amount of data that must be sent to ground.
- Imposing a constraint on the total amount of data storage a bus can have or limiting its upgrade potential.

Appendix J: Satellite Network MILP Use Case Example

1 Base Scenario – Max Data to Ground Objective – Link Schedule

```
36.449 data sent from X-Band Ground 2 at location 2 to Standard GS at location 2 during t=[7408:7535]
3.048 data sent from SOURCE to HYSPIRI TIR at location 1 during t=[7408:7535]
50.8 data sent from SSTL-600 at location 0 to X-Band Upgrade-2 at location 0 during t=[7408:7535]
36.449 data sent from X-Band Ground 1 at location 2 to Standard GS at location 2 during t=[7408:7535]
19.05 data sent from HYSPIRI VSWIR at location 1 to X-Band Upgrade-1 at location 1 during t=[7408:7535]
50.8 data sent from SSTL-600 at location 0 to X-Band Upgrade-1 at location 0 during t=[7408:7535]
36.449 data sent from X-Band Upgrade-1 at location 1 to X-Band Ground 1 at location 3 during t=[7408:7535]
11.049 data sent from X-Band Upgrade-2 at location 1 to X-Band Ground 2 at location 2 during t=[7408:7535]
3.048 data sent from HYSPIRI TIR at location 1 to X-Band Upgrade-1 at location 1 during t=[7408:7535]
36.449 data sent from X-Band Upgrade-1 at location 1 to X-Band Ground 2 at location 3 during t=[7408:7535]
125.4 data sent from X-Band Upgrade-2 at location 1 to X-Band Ground 1 at location 3 during t=[7535:7991]
53.972 data sent from X-Band Upgrade-2 at location 0 to X-Band Ground 1 at location 3 during t=[7535:7991]
53.972 data sent from X-Band Upgrade-2 at location 0 to X-Band Ground 2 at location 3 during t=[7535:7991]
3.028 data sent from X-Band Upgrade-1 at location 1 to X-Band Ground 1 at location 3 during t=[7535:7991]
178.54 data sent from SSTL-600 to ITSELF at location 1 At t=7991
364.8 data sent from Standard GS at location 3 to SINK during t=[7535:7991]
3.028 data sent from X-Band Upgrade-1 at location 1 to X-Band Ground 2 at location 3 during t=[7535:7991]
68.4 data sent from HYSPIRI VSWIR at location 1 to X-Band Upgrade-2 at location 1 during t=[7535:7991]
182.4 data sent from SSTL-600 at location 1 to X-Band Upgrade-2 at location 1 during t=[7535:7991]
182.4 data sent from X-Band Ground 2 at location 3 to Standard GS at location 3 during t=[7535:7991]
136.8 data sent from FSO Crosslink Config 2 at location 1 to SSTL-600 at location 1 during t=[7535:7991]
6.056 data sent from SSTL-600 at location 1 to X-Band Upgrade-1 at location 1 during t=[7535:7991]
136.8 data sent from SSTL-600 at location 1 to FSO Crosslink Config 2 at location 1 during t=[7535:7991]
10.944 data sent from SOURCE to HYSPIRI TIR at location 1 during t=[7535:7991]
182.4 data sent from X-Band Ground 1 at location 3 to Standard GS at location 3 during t=[7535:7991]
68.4 data sent from SOURCE to HYSPIRI VSWIR at location 1 during t=[7535:7991]
10.944 data sent from HYSPIRI TIR at location 1 to FSO Crosslink Config 2 at location 1 during t=[7535:7991]
10.944 data sent from FSO Crosslink Config 2 at location 1 to FSO Crosslink Config 1 at location 0 during t=[7535:7991]
10.944 data sent from FSO Crosslink Config 1 at location 0 to SSTL-600 at location 0 during t=[7535:7991]
107.944 data sent from SSTL-600 at location 0 to X-Band Upgrade-2 at location 0 during t=[7535:7991]
125.4 data sent from X-Band Upgrade-2 at location 1 to X-Band Ground 2 at location 3 during t=[7535:7991]
686.3 data sent from HYSPIRI VSWIR at location 1 to SSTL-600 at location 1 during t=[7991:12577]
1.6 data sent from X-Band Ground 1 at location 2 to Standard GS at location 2 during t=[7991:12577]
1.6 data sent from HYSPIRI VSWIR at location 1 to X-Band Upgrade-1 at location 1 during t=[7991:12577]
```

Figure 40: Base scenario network MILP link schedule.

2 Base Scenario – Minimizing Costs Solver Results

```

Presolve removed 19047 rows and 4043 columns
Presolve time: 0.22s
Presolved: 10905 rows, 5246 columns, 32991 nonzeros
Variable types: 5224 continuous, 22 integer (22 binary)

Root relaxation: objective 1.457673e+05, 6090 iterations, 1.59 seconds

```

Nodes		Current Node			Objective Bounds			Work		
Expl	Unexpl	Obj	Depth	IntInf	Incumbent	BestBd	Gap	It/Node	Time	
0	0	145767.344	0	7	-	145767.344	-	-	1s	
0	0	146314.113	0	7	-	146314.113	-	-	2s	
0	0	146325.495	0	7	-	146325.495	-	-	2s	
0	0	148722.100	0	9	-	148722.100	-	-	3s	
0	0	152352.211	0	8	-	152352.211	-	-	3s	
0	0	154327.123	0	10	-	154327.123	-	-	4s	
0	0	154327.123	0	10	-	154327.123	-	-	4s	
0	0	154333.871	0	10	-	154333.871	-	-	5s	
0	0	154334.295	0	10	-	154334.295	-	-	5s	
0	0	154337.232	0	10	-	154337.232	-	-	5s	
0	0	154337.457	0	10	-	154337.457	-	-	5s	
0	0	154340.848	0	10	-	154340.848	-	-	5s	
0	0	154342.105	0	10	-	154342.105	-	-	5s	
0	0	154344.511	0	10	-	154344.511	-	-	6s	
H	0	0			166300.00000	154344.511	7.19%	-	7s	
H	0	0			157300.00000	154344.511	1.88%	-	8s	
	0	2	154344.511	0	10	157300.000	154344.511	1.88%	-	8s
H	1	4			156600.00100	154346.039	1.44%	0.0	8s	

```

Cutting planes:
Cover: 2
Implied bound: 5
MIR: 87
Flow cover: 11
GUB cover: 2

Explored 18 nodes (22260 simplex iterations) in 9.83 seconds
Thread count was 4 (of 4 available processors)

Solution count 3: 156600 157300 166300

Optimal solution found (tolerance 1.00e-04)
Best objective 1.566000010000e+05, best bound 1.566000010000e+05, gap 0.0000%

```

Figure 41: Base scenario, minimizing cost objecting solver result

3 Network MILP Scenario – Added Input Options

Location Node	Location Name	Lat	Long	Alt	Fixed – Must Have
l_4	McMurdo	-77.846323	166.668235	0	g_0 , Standard GS c_6 , X-Band Ground
l_5	Troll	-72.0167	2.5333	0	g_0 , Standard GS c_6 , X-Band Ground
l_6	Optional GS 1 (Dongara)	-29.25	114.93	0	-
l_7	Optional GS 2 (South Point)	37.85	-75.466667	0	-
l_8	Optional GS 3 (Wallops)	8.9111	-155.681111	0	-
l_9	Optional GS 4 (Santiago)	-33.45,	-70.666667	0	-

Table 45 Additional ground sites for network MILP scenario.

Child Object	Name	Data Storage	Data Processing	Constraints
g_1	Optional GS	0	Yes	Limited to Optional Sites

Table 46: Additional ground station for network MILP scenario.

Child Object	Name	Comm . Tech.	Data Rate (Gbit/s)	Trans/Receive	Field of View	Orien .	Orbit Avg. Power (W)	Mas s (kg)	Cos t (K)	Limited To	Mutua l Excl.
c_9	X-Band Ground Optional	X-Band	.2	TR	90	[0,90]	0	0	200	None-bus Optiona l GS	-

Table 47: Additional communications technology for network MILP scenario.

4 Network MILP Scenario – Added Input Options, Max Data to Ground Objective

```

Optimize a model with 76396 rows, 24059 columns and 183537 nonzeros
Variable types: 24019 continuous, 40 integer (40 binary)
Coefficient statistics:
  Matrix range      [1e-08, 1e+09]
  Objective range   [1e+00, 1e+00]
  Bounds range      [1e+00, 1e+00]
  RHS range         [3e-02, 9e+02]
Warning: Model contains large matrix coefficient range
         Consider reformulating model or setting NumericFocus parameter
         to avoid numerical issues.
Presolve removed 46088 rows and 9938 columns
Presolve time: 0.83s
Presolved: 30308 rows, 14121 columns, 90158 nonzeros
Variable types: 14099 continuous, 22 integer (22 binary)
Found heuristic solution: objective 581.2090000

Root simplex log...

Iteration   Objective      Primal Inf.    Dual Inf.      Time
   11185    5.5306964e+02  2.549812e+02  0.000000e+00   5s
   11625    5.5306400e+02  0.000000e+00  0.000000e+00   5s

Root relaxation: objective 5.530640e+02, 11625 iterations, 4.06 seconds

  Nodes      |      Current Node      |      Objective Bounds      |      Work
  Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd  Gap | It/Node Time
*    0       0              |      0      553.0640000  553.06400  0.00%  -   5s

Explored 0 nodes (12847 simplex iterations) in 5.52 seconds
Thread count was 4 (of 4 available processors)

Solution count 2: 553.064 581.209

Optimal solution found (tolerance 1.00e-04)
Best objective 5.530640000000e+02, best bound 5.530640000000e+02, gap 0.0000%

```

Figure 42: Added options scenario - maximize data to ground solver results.

```

The total data generated by sensors in this scenario is: 15033.426
The total data sent to ground is:14480.361999999999
The maximum amount of data stored on SSTL-600 at location 0 is: 211.804
The end amount of data stored on SSTL-600 at location 0 is: 211.804
The maximum amount of data stored on SSTL-600 at location 1 is: 341.26
The end amount of data stored on SSTL-600 at location 1 is: 341.26

```

Figure 43: Added options scenario - maximize data to ground - data throughput.

```
3564.233999999995 was received at location: 2
2333.696 was received at location: 3
3411.405999999995 was received at location: 4
2910.6140000000005 was received at location: 5
516.172 was received at location: 6
683.9519999999999 was received at location: 7
496.6 was received at location: 8
563.688 was received at location: 9
```

Figure 44: Added options scenario - maximize data to ground - data received at each ground station.

```
HYSPIRI TIR is assigned to location node 0
HYSPIRI VSWIR is assigned to location node 1
SSTL-600 is assigned to location node 0
SSTL-600 is assigned to location node 1
Standard GS is assigned to location node 2
Standard GS is assigned to location node 3
Standard GS is assigned to location node 4
Standard GS is assigned to location node 5
Optional GS is assigned to location node 6
Optional GS is assigned to location node 7
Optional GS is assigned to location node 8
Optional GS is assigned to location node 9
FSO Crosslink Config 1 is assigned to location node 0
X-Band Upgrade-1 is assigned to location node 0
X-Band Upgrade-2 is assigned to location node 0
FSO Crosslink Config 2 is assigned to location node 1
X-Band Upgrade-1 is assigned to location node 1
X-Band Upgrade-2 is assigned to location node 1
X-Band Ground 2 is assigned to location node 2
X-Band Ground 2 is assigned to location node 3
X-Band Ground 2 is assigned to location node 4
X-Band Ground 2 is assigned to location node 5
X-Band Ground Optional is assigned to location node 6
X-Band Ground Optional is assigned to location node 7
X-Band Ground Optional is assigned to location node 8
X-Band Ground Optional is assigned to location node 9
```

Figure 45: Added options scenario - maximize data to ground - architecture decisions.

```
The total cost of this architecture is: 220800.0
```

Figure 46: Added options scenario - maximize data to ground - total cost.

5 Network MILP Scenario – Added Input Options, Minimize Total Cost

```
Explored 35 nodes (104170 simplex iterations) in 64.36 seconds
Thread count was 4 (of 4 available processors)

Solution count 4: 156600 157300 166300 167300

Optimal solution found (tolerance 1.00e-04)
Best objective 1.566000010000e+05, best bound 1.566000010000e+05, gap 0.0000%
```

Figure 47: Added options scenario – minimize cost - solver results.

```
The total data generated by sensors in this scenario is: 15033.426
The total data sent to ground is:14057.701999999581
The maximum amount of data stored on SSTL-300 ESPA at location 0 is: 512.0
The end amount of data stored on SSTL-300 ESPA at location 0 is: 463.724
The maximum amount of data stored on SSTL-150 ESPA at location 1 is: 512.0
The end amount of data stored on SSTL-150 ESPA at location 1 is: 512.0
```

Figure 48: Added options scenario – minimize cost - data throughput.

```
HYSPIRI TIR is assigned to location node 0
HYSPIRI VSWIR is assigned to location node 1
SSTL-300 ESPA is assigned to location node 0
SSTL-150 ESPA is assigned to location node 1
Standard GS is assigned to location node 2
Standard GS is assigned to location node 3
Standard GS is assigned to location node 4
Standard GS is assigned to location node 5
Optional GS is assigned to location node 7
Optional GS is assigned to location node 9
SSTL-300 ESPA Comm is assigned to location node 0
X-Band Upgrade-1 is assigned to location node 1
X-Band Ground 2 is assigned to location node 2
X-Band Ground 2 is assigned to location node 3
X-Band Ground 2 is assigned to location node 4
X-Band Ground 2 is assigned to location node 5
X-Band Ground Optional is assigned to location node 7
X-Band Ground Optional is assigned to location node 9
```

Figure 49: Added options scenario – minimize cost - architecture decisions.

6 Published HypsIRI Architecture Results

```
HYSPIRI VSWIR is assigned to location node 0
HYSPIRI TIR is assigned to location node 0
SSTL-600 is assigned to location node 0
Standard GS is assigned to location node 1
Standard GS is assigned to location node 2
X-Band Upgrade-1 is assigned to location node 0
X-Band Upgrade-2 is assigned to location node 0
X-Band Ground 1 is assigned to location node 1
X-Band Ground 2 is assigned to location node 1
X-Band Ground 1 is assigned to location node 2
X-Band Ground 2 is assigned to location node 2
```

Figure 50: Published HypsIRI Architecture

```
The total data generated by sensors in this scenario is: 15033.6
The total data sent to ground is:11266.1659999999
The maximum amount of data stored on SSTL-600 at location 0 is: 3898.868
The end amount of data stored on SSTL-600 at location 0 is: 3767.434
```

Figure 51: Data throughput of published HypsIRI architecture.

```
The total cost of this architecture is: 157400.0
```

Figure 52: Cost of published HypsIRI architecture.

Bibliography

- [1] NASA, NASA Systems Engineering Handbook, Rev 2, Washington D.C.: NASA, 2016.
- [2] The Institute of Electrical and Electronics Engineers, Inc, "IEEE Recommended Practice for Architectural Description of Software-Intensive Systems," The Institute of Electrical and Electronics Engineers, Inc., New York, 2000.
- [3] NOAA, "Mission and Instruments, Joint Polar Satellite System," [Online]. Available: http://www.jpss.noaa.gov/mission_and_instruments.html. [Accessed 12 December 2017].
- [4] National Oceanic and Atmospheric Administration, "NOAA Technical Report NESDIS 142: Visible Infrared Imaging Radiometer Suite (VIIRS) Sensor Data Record (SDR) User's Guide, Ver 1.2," National Oceanic and Atmospheric Administration, Washington D.C., 2013.
- [5] B. Doncaster, C. Williams and J. Shulman, "Nano/Microsatellite Market Forecast," SpaceWorks Enterprises, Inc, Atlanta, 2017.
- [6] T. Dubois, "OneWeb Satellites' Factory Gears Up For Mass Production," 28 June 2017. [Online]. [Accessed 3 January 2018].
- [7] J. Bennett, "Popular Mechanics," 13 July 2017. [Online]. Available: <http://www.popularmechanics.com/space/rockets/a27290/one-chart-spacex-dominate-rocket-launches/>. [Accessed 20 December 2017].
- [8] D. Z. Morris, "SpaceX Launch Costs Substantially Undercut Competition, Analysis Says," 17 June 2017. [Online]. [Accessed 8 January 2018].
- [9] B. d. Selding, "Space News," 6 April 2016. [Online]. [Accessed 3 January 2018].
- [10] Spaceflight Industries, "Spaceflight Industries," 2017. [Online]. Available: <http://spaceflight.com/about/>. [Accessed 22 December 2017].
- [11] Rocket Lab USA, "Electron," 2018. [Online]. Available: <https://www.rocketlabusa.com/electron/>. [Accessed 20 January 2018].
- [12] A. Rogalski, "Recent progress in infrared detector technologies, Vol 54," *Infrared Physics and Technology*, pp. 136-154, 2011.
- [13] NASA, "Landsat 8 Instruments," 2016. [Online]. [Accessed 7 January 2018].
- [14] P. M. J. Puschell, "Uncooled emissive infrared imagers for Cubesats," 2014.
- [15] J. Lewis, "Wafer Scale Infrared Detectors (WIRED), DARPA," [Online].
- [16] New Infrared Technologies, "QBITO Cubesat," 2017. [Online].
- [17] A. C. X. P. J.-P. D. B. d. M. M. L. E. M. Gil Denis, "Towards disruptions in Earth observation? New Earth Observation systems and markets evolution: Possible scenarios and impacts," *Act Astronautica 137*, pp. 415-433, 2017.
- [18] Satellogic, "Sole supplier of hyperspectral imagery for advanced geo-intelligence," 2017. [Online]. [Accessed 8 1 2018].

- [19] National Geospatial-Intelligence Agency, "NGA purchases \$14 million subscription to utilize small satellite capabilities," 19 July 2017. [Online]. [Accessed 8 January 2018].
- [20] D. Werner, "NOAA sees great promise and challenges in using data from small satellite constellation - SpaceNews.com," 8 January 2018. [Online]. [Accessed 1 January 2018].
- [21] J. Foust, "President signs commercial satellite weather bill," 21 April 2017. [Online]. [Accessed 1 January 2018].
- [22] DIUX, "Portfolio," 2017. [Online]. Available: <https://www.diux.mil/portfolio>.
- [23] J. R. Schott, *Remote Sensing: The Image Chain Approach*, 2nd ed., New York: Oxford University Press, 2007.
- [24] J. R. Wertz, D. F. Everett and J. J. Puschell, *Space Mission Engineering: The New SMAD*, Hawthorne: Microcosm Press, 2011.
- [25] M. T. Eismann, *Hyperspectral Remote Sensing*, Bellingham: SPIE, 2012.
- [26] P. Fortescue, S. Graham and J. Stark, *Spacecraft Systems Engineering*, 4 ed., West Sussex: John Wiley & Sons Ltd, 2011.
- [27] Galorath, "SEER Cost Estimation for Hardware, Electronics & Systems," [Online]. [Accessed 23 December 2017].
- [28] Galorath, "SEER-H Space Guidance Rev 2.2," 2016.
- [29] J. Wertz, "Commentary | Reiventing Space: Dramatically Reducing SPace Mission Cost," 4 February 2013. [Online]. [Accessed 9 January 2018].
- [30] O. L. De Weck and M. B. Jones, "Isoperformance: Analysis and Design of Complex Systems with Desired Outcomes," *Systems Engineering Vol 9*, pp. 45-61, 2006.
- [31] S. Russell and P. Norvig, *Artificial Intellegince: A Modern Approach*, Third Edition, Upper Saddle River, New Jersey: Pearson Education, Inc., 2010.
- [32] Raytheon Vision Systems, "The Infrared Wall Chart," 2008. [Online]. Available: http://www.pblprojects.org/EOC/docs/Raytheon_IR_rvs_wallchart.pdf. [Accessed 3 January 2018].
- [33] Raytheon, "Visible/Infrared Imager Radiometer Suite (VIIRS)," 2011. [Online]. Available: https://jointmission.gsfc.nasa.gov/images/VIIRS_DS152%20Approved%208-10-11.pdf. [Accessed 19 February 2018].
- [34] NASA, "HyspIRI Mission Study," 2017. [Online]. Available: <https://hyspiri.jpl.nasa.gov/>. [Accessed 20 February 2018].
- [35] R. O. Green, "HyspIRI Mission Study," 16 October 2017. [Online]. Available: <https://hyspiri.jpl.nasa.gov/documents/2017-workshop/>. [Accessed 21 02 2018].
- [36] Surrey Satellite Technology US, "SSTL-150 ESPA Satellite Platform," June 2016. [Online]. Available: https://rsdo.gsfc.nasa.gov/images/201608/SSTL-150ESPA_NASABrochure-2016.pdf. [Accessed 20 January 2018].
- [37] Surrey Satellite Technology US, "SSTL-300 Satellite Platform," June 2016. [Online]. Available: <http://www.sst-us.com/getfile/cc443f25-7ea7-471c-b67e-4061401257cf>. [Accessed 20 January 2018].
- [38] M. Mercury, "HyspIRI 2012 Technology Readiness Assessment," Jet Propulsion Laboratory, Pasadena, 2012.

- [39] Surrey Satellite Technology US LLC, "SSTL-600 Satellite Platform," May 2016. [Online]. Available: <https://rsdo.gsfc.nasa.gov/images/catalog/SSTL600.pdf>. [Accessed 20 January 2018].
- [40] Spaceflight Industries, "Pricing Information," Spaceflight Industries, 2018. [Online]. Available: <http://spaceflight.com/schedule-pricing/#pricing>. [Accessed 1 March 2018].
- [41] Surrey Satellite Technology, "XTx-400 X-Band Transmitter," August 2012. [Online]. Available: <https://www.sst-us.com/SSTL/files/d1/d1a3c6a1-3142-4104-93d7-35e625ffff03.pdf>. [Accessed 2 March 2018].
- [42] U.S. General Services Administration, "GSA Advantage," 24 June 2012. [Online]. Available: https://www.gsaadvantage.gov/ref_text/GS35F0261Y/0KS9BE.2HT2VQ_GS-35F-0261Y_SSTUSLLCOFFICIALGSASCHEDULE70PRICELISTJUNE2012.PDF. [Accessed 2 March 2018].
- [43] Blue Canyon Technologies, "Microsat Spacecraft," 2017. [Online].
- [44] Millennium Space Systems, Inc., "Platforms," 2017. [Online]. Available: <http://www.millennium-space.com/platforms.html#altair>. [Accessed 20 January 2018].
- [45] York Space Systems, "Platform," 2017. [Online]. Available: https://www.yorkspacesystems.com/?page_id=30. [Accessed 20 January 2018].
- [46] Surrey Satellite Technology US LLC, "SSTL-100 Satellite Platform," June 2016. [Online]. Available: <http://www.sst-us.com/getfile/6a4b7182-1672-4334-8e66-33e0d7ade08a>. [Accessed 20 January 2018].
- [47] Surrey Satellite Technology US LLC, "SSTL-150 Satellite Platform," June 2016. [Online]. Available: <http://www.sst-us.com/getfile/c0fca153-4731-4103-92f5-880c28869b21>. [Accessed 20 January 2018].
- [48] Lockheed Martin, "LM100," [Online]. Available: https://rsdo.gsfc.nasa.gov/images/201608/LM-100_Brochure.pdf. [Accessed 20 January 2018].
- [49] Sierra Nevada Corporation, "SNC Spacecraft Systems," [Online]. Available: https://www.sncorp.com/media/2398/ssg_spacecraft-systems-bifold_for-web_4-4-18.pdf. [Accessed 20 January 2018].
- [50] Ball Aerospace and Technologies Corp, "BCP 300," [Online]. Available: https://rsdo.gsfc.nasa.gov/images/catalog/BCP300_Spacecraft%20Data%20Package.pdf. [Accessed 20 January 2018].
- [51] QinetiQ, "The P200 Platform," 2016. [Online]. Available: https://rsdo.gsfc.nasa.gov/images/201608/Rapid_III_Spacecraft_Datapackage_for_QS_P200-10Aug16.pdf. [Accessed 20 January 2018].
- [52] Orbital ATK, "LEOSTar-2 Bus," [Online]. Available: https://www.orbitalatk.com/space-systems/spacecraft-buses/docs/LEOSTar-2_Factsheet.pdf. [Accessed 20 January 2018].
- [53] Firefly Aerospace Inc., "Alpha Payload User's Guide," March 2018. [Online]. Available: <http://www.fireflyspace.com/assets/files/Firefly%20Alpha%20Payload%20User's%20Guide.pdf>. [Accessed 28 March 2018].
- [54] Rocket Lab USA, "Electron," 2018. [Online]. Available: <https://www.rocketlabusa.com/electron/>. [Accessed 28 March 2018].

- [55] Virgin Orbit, "Virgin Orbit," 2018. [Online]. Available: <https://virginorbit.com/>. [Accessed 28 March 2018].
- [56] Gilmoure Space Technologies, "Launch Vehicles," 2018. [Online]. Available: <https://www.gspacetechnologies.com/launch-vehicles>. [Accessed 28 March 2018].
- [57] R. Müller, "Calibration and Verification of Remote Sensing Instruments and Observations," *Remote Sensing*, pp. 5692-5695, 2014.
- [58] "Visible Infrared Imaging Radiometer Suite (VIIRS), p 11," [Online]. Available: <https://ncc.nesdis.noaa.gov/documents/documentation/viirs-users-guide-tech-report-142a-v1.3.pdf>.