The Pennsylvania State University
The Graduate School

SYSTEM DESIGN AND INSTRUMENT DEVELOPMENT FOR THE OSIRIS-3U CUBESAT MISSION

A Thesis in
Electrical Engineering
by
Allen Theodore Kummer

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2012

The thesis of Allen Theodore Kummer was reviewed and approved* by the following:

Sven G. Bilén
Associate Professor of Engineering Design, Electrical Engineering, and Aerospace Engineering
Thesis Advisor

Julio V. Urbina
Associate Professor of Electrical Engineering

John D. Mathews
Professor of Electrical Engineering

Kultegin Aydin
Professor of Electrical Engineering
Interim Department Head

*Signatures are on file in the Graduate School.

Abstract

Cube satellites (CubeSats) have increased in popularity in recent years as a tool for space research and education. Students and faculty at Penn State's Student Space Programs Laboratory (SSPL) have been developing the Orbital Satellite to Investigate the Response of the Ionosphere to Stimulation and Space weather (OSIRIS-3U) CubeSat. By measuring the spatial and temporal response of the ionosphere using *in situ* and remote sensing instruments, OSIRIS-3U will help answer questions on plasma transport in the stimulated ionosphere. This work focuses on the mission development and incremental development of a science instrument for OSIRIS-3U.

Mission development is a balance between system capabilities and desired measurements to meet the mission objectives. Building a system model is key to understanding system sensitivity to tradeoffs of operational time to power usage, and numbers of orbits through a science region to generated data volume. A Monte Carlo simulation of the OSIRIS-3U mission in AGI's Satellite Tool Kit (STK) provides the data points to observe and analyze system sensitivities using the Applied Research Laboratory's Trade Space Visualizer (ATSV) tool. This understanding is required to develop a clear set of requirements to guide future system development and implementation.

The OSIRIS-3U system carries three primary instruments to address the science question including a Langmuir probe (LP) developed at Penn State, a UHF beacon developed by the Naval Research Laboratory, and an occultation instrument developed by The Aerospace Corporation. Previous work has been done at Penn State to develop another instrument that combines the LP with a plasma impedance probe (PIP), providing increased mission flexibility in small volumes such as on CubeSats. The present work incrementally builds on previous work at Penn State by building, calibrating, and testing the PIP and comparing the results with a network analyzer and laying a path for future development.

Table of Contents

Chapter 3

Chapter 4

Chapter 5

List of Figures

List of Tables

Acknowledgments

I would like to begin by thanking the students of the Student Space Programs Laboratory for volunteering to spend long nights working on the various projects over the course of the years. Without your efforts, we would not have been able to progress the OSIRIS-3U mission to where it is today. I would especially like to thank the OSIRIS subsystem leads—while physically small, a CubeSat project is much too complex to be lead by any one person. Your knowledge in your respective subsystems has been invaluable and it has been a great to learn about them with you.

I would like to thank my friend and mentor Brian Schratz for getting me involved freshman year and introducing me to systems engineering. Your support and guidance over the years has helped keep me going through the ups and downs of project development.

Mr. Bob Capuro, I would like to thank you for bringing your many years of experience into the lab. Your anecdotes have been a great view into the broader applications of what I am learning.

To my mentor and advisor, Dr. Sven Bilén, thank you for the constant reminder to never let my schooling get in the way of my education. This has helped me balance learning between the classroom and hands on SSPL work over the years.

Above all, I would like to thank my loving family—parents, Pam and Karl, and sister, Donna—for their support during my long stints away from home and their understanding as school and SSPL work kept me at school for long stints over the years.

Allen Theodore Kummer

Chapter 1

Introduction

Space exploration has captured the interest of many as the final frontier; however, with launch costs estimated at around $10K per kg [1] of payload, entrance into this challenging domain is tempered by the high cost of launching exploration satellites into space. Therefore, it follows that it is difficult for academic programs with the goal of training students for careers in the space industry to affordably enter this domain. Nevertheless, many academic institutions and national organizations have set out to pursue this goal by developing smaller spacecraft and satellites.

Spacecraft "size" is classified by mass as shown in Table 1.1. Educational programs have focused on smaller satellites in the micro- and nanosat classes due to the potential of lower launch costs.

In 1994, Stanford University (SU) developed a program to expose students to satellite development within the scope of their course curriculum on the premise that a low–cost or no–cost target–of–opportunity launch could be found [3]. One of the first satellites developed through this program was the Orbital Pod And Launcher (OPAL) satellite. Beginning in April 1995 [3], this small satellite was built to demonstrate a mother–daughter system in which the OPAL spacecraft (mother) carried four nonosatellites (daughters), sized at 4 inches by 3 inches by one inch, into orbit for subsequent release. OPAL demonstrated the carrier and deployment system concept that later became known as P-POD.

While continuing to focus on their nanosatellite program, SU partnered with the California Polytechnic State University (CalPoly) to continue developing microsatellite deployment technology [3]. Students and faculty at CalPoly collabo-

Table 1.1: Satellite classifications [2].

Class	Mass (kg)
Large Satellite	> 1000
Small Satellite	500 – 1000
Minisatellite	100 – 500
Microsatellite	10 – 100
Nanosatellite	1 – 10
Picosatellite	< 1

rated with the community to design, build, and evaluate a prototype microsatellite sized to enable storage of up to three units in a single carrier/launcher. Thus, the P-Pod and the CubeSat standard was born.

The P-Pod is a self-contained and standard carrier/launcher that weighs 10 kg when fully loaded. The P-Pod provides a standard launch vehicle interface for easy microsatellite integration. Once P-POD and launch vehicle integration has been validated, subsequent system integrations have much short timelines and, therefore, lower costs for each CubeSat launch. To minimize risk to the host spacecraft, the P-Pod is designed to contain the potential energy stored inside in case of an issue with the contained CubeSat(s). By defining and maintaining a standard for P-POD–based secondary payloads, the complexity of securing a slot and installing a daughter payload in a launch vehicle mission can be substantially reduced.

A CubeSat is defined in terms of 10-cm cubic units, with one such unit being a 1U CubeSat increasing in length up to a 3U CubeSat. Each unit is allocated 1.33 kg of mass, so a single CubeSat can be up to 4 kg in a 3U form factor. Figure 1.1 shows an example of both the P-Pod and a model CubeSat. Once integrated with the P-Pod, CubeSats are required to be inactive and batteries charged until ejection by the P-POD. At ejection from the P-POD, the fully charged CubeSat batteries are disconnected from the spacecraft. Once on orbit the CubeSat power is derived from solar cells.

As of May 2012, there were 38 CubeSats on orbit as summarized in Table 1.2. Many of the currently orbiting CubeSats are 1U CubeSats, although the recent trend in CubeSat launches has been moving increasingly towards 3U CubeSats to facilitate more complex missions.

Table 1.2: Current CubeSat missions as of May 2012 [5][6].

	Altitude at Perigee [km]	Altitude at Apogee [km]	Inclination [degrees]	Eccentricity
DTUSAT	814.91	827.04	98.695	0.001
CUTE-1 (CO-55)	815.52	828.22	98.702	0.001
QUAKESAT	817.06	828.53	98.714	0.001
AAU CUBESAT	814.91	827.05	98.692	0.001
CANX-1	814.99	827.17	98.691	0.001
CUBESAT XI-IV (CO-57)	678.76	703.48	97.916	0.002
UWE-1	679.43	702.98	97.917	0.002
CUBESAT XI-V (CO-58)	816.53	828.90	98.711	0.001
NCUBE-2	678.94	701.80	97.918	0.002
CSTB1	644.83	765.17	97.866	0.008
MAST	645.29	778.90	97.875	0.009
LIBERTAD-1	643.99	788.58	97.882	0.010
POLYSAT CP3	643.43	787.86	97.880	0.010
CAPE1	643.71	788.44	97.883	0.010
POLYSAT CP4	645.50	766.75	97.867	0.009
NTS (CANX-6)	610.21	633.77	97.801	0.002
CUTE-1.7+APD II (CO-65)	608.91	631.01	97.801	0.002
COMPASS-1	605.34	628.29	97.801	0.002
AAUSAT-II	604.67	626.97	97.804	0.002
DELFI-C3 (DO-64)	600.08	622.34	97.814	0.002
CANX-2	608.90	631.69	97.799	0.002
SEEDS II (CO-66)	606.86	630.40	97.802	0.002
SWISSCUBE	709.31	722.20	98.340	0.001
BEESAT	710.55	717.83	98.341	0.001
UWE-2	709.63	719.36	98.335	0.001
ITUPSAT 1	709.34	723.20	98.345	0.001
TISAT 1	613.24	633.28	98.063	0.001
DICE-F	456.05	804.57	101.720	0.025
DICE-Y	455.93	803.11	101.722	0.025
RAX-2	456.21	802.79	101.716	0.025
AUBIESAT-1	454.78	801.88	101.716	0.025
M-CUBED & EXP-1 PRIME	454.52	802.98	101.719	0.025
E-ST@R	305.98	1384.61	69.494	0.075
Goliat	305.48	1386.00	69.486	0.075
MaSat-1	305.34	1387.09	69.485	0.075
Xatcobeo	305.70	1376.87	69.496	0.074
PW-Sat1	306.13	1380.15	69.494	0.074
UNICubeSAT	304.56	1391.41	69.495	0.075
	Altitude at Perigee [km]	Altitude at Apogee [km]	Inclination [degrees]	Eccentricity
Average	594.88	841.91	94.074	0.017
Minimum	304.56	622.34	69.485	0.001
Maximum	817.06	1391.41	101.722	0.075

1.1 CubeSat Limitations

While the small size of the CubeSat standard makes it less costly to launch, it limits the surface area available for solar cell exposure to the Sun while on orbit. Some CubeSats are equipped with deployable solar panels (e.g., available from Boeing, Pumpkin Space), but the vast majority of CubeSats rely on face-mounted solar cells to provide system power. Therefore, the small surface area limits the power available to operate the CubeSat subsystems. Consequently, the CubeSat subsystems and instrumentation must operate at low power levels or low operating duty cycles to conserve energy.

CubeSats typically fly in low earth orbit (LEO), i.e., in orbits that expose them to atmospheric drag, which will cause loss of altitude followed by deorbit. This phenomenon will cause a shorter in-service lifetime unless a orbital-boost propulsion system is incorporated (still rare on CubeSats). While mission lifetime increases exponentially with altitude, spacecraft/satellites with orbits below 350 km typically experience lifetimes of less than 6 months. CubeSats without

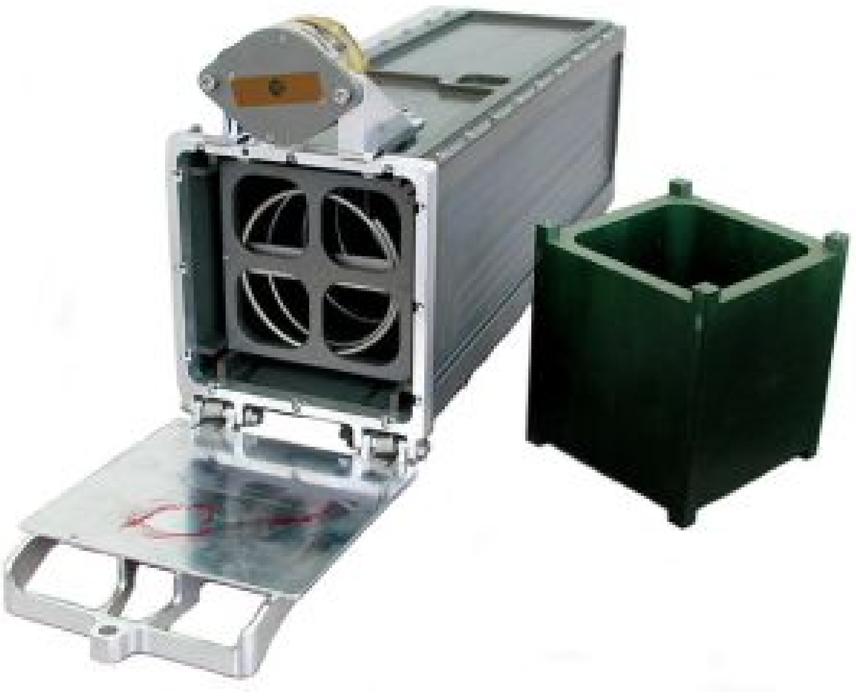


Figure 1.1: The Cal Poly P-Pod CubeSat launcher next to a CubeSat model [4].

propulsion systems require an altitude of at most 600 km to meet the requirement of deorbiting within 25 years of launch. Some methods currently in development to help deorbit CubeSats sooner include electrodynamic tethers and solar sails, although these have yet to be demonstrated operationally.

The relatively low success rate of CubeSat missions, the small size of CubeSats and the interest by a number of researchers to launch clouds of CubeSats, has lead the space industry to view them as “Space Junk”. While CubeSat size is marginally large enough to be tracked by systems like NORAD (North American Aerospace Defense Command), they are large enough to cause major problems with existing satellites. This especially is of concern when human lives are at risk as in operations on the International Space Station (ISS) and the future presence of commercial space vehicles ferrying astronauts. On numerous occasions the crew members on the ISS seek shelter in escape vehicles when debris passes near the station for fear of collision and subsequent loss of life. The danger caused by “Space Junk” has become an increasingly important topic since the Chinese destroyed their own satellite in 2007 and the 2009 collision of a defunct Russian satellite and an Iridium communications satellite. This causes pressure on CubeSat designers to limit some of the concepts implemented such as those involving clusters of CubeSats flying in formation.

1.2 CubeSats Are Not *All* Bad!

While CubeSats have some undesirable characteristics, their (relatively) low cost to develop and launch makes them a great platform for high risk missions. Typically, CubeSats are used as educational tools at the university level to teach students interdisciplinary design, project management, systems engineering, and space systems. In this capacity, CubeSat developers can partner with industry, scientists, and engineers to fly new technology that might otherwise wait years to earn flight heritage or target specific science questions that might not otherwise require a full complement of instruments on the flight system.

1.3 Thesis Motivation

The Pennsylvania State University Student Space Programs Laboratory (SSPL) is a faculty-led, student-run laboratory focused on teaching systems engineering through hands-on experiences at primarily the undergraduate level. SSPL has been working on developing various satellite missions since 2006. The SSPL has developed into a place for students from the disciplines of mechanical engineering, electrical engineering, aerospace engineering, computer science, computer engineering, physics, and many more, to come together as an interdisciplinary team to develop systems from concept through flight, operations, and data analysis. This environment provides a unique atmosphere for students to learn about the field of systems engineering while applying their discipline-specific skills to solve a common problem. Students often benefit from the additional hands on experience to complement the experiences afforded them through their respective course work.

1.3.1 OSIRIS-3U Mission Summary

The faculty and students of the SSPL have been working to develop a CubeSat to study the interaction between the Sun and the Earth. The Orbital Satellite to Investigate the Response of the Ionosphere to Stimulation and Space Weather (OSIRIS-3U) CubeSat mission concept is built around a plasma diagnostic instrument designed to measure *in situ* plasma parameters of density and temperature. OSIRIS-3U uses the high frequency atmospheric heater located at the National Science Foundation's Arecibo Observatory in Puerto Rico to excite ("stimulate") the ionosphere prior to OSIRIS-3U passing over Puerto Rico in its orbit. The OSIRIS-3U CubeSat will use onboard instruments to observe the heated region at various spatial scales using remote sensing instruments; the Aerospace Corporation's GPS Radio Occultation (GPSRO) receiver and the Naval Research Laboratory's Coherent Electromagnetic Radio Tomography (CERTO) radio beacon; and a Penn State-built Langmuir Probe for high spatial resolution *in situ* measurements.

In addition to the science goals, OSIRIS-3U serves as an educational platform for students at SSPL to learn about engineering through hands on project work. This unique environment helps to prepare them for work as part of a multidisciplinary team and to understand the interrelation between individual subsystems.

This allows students to learn about systems engineering and helps to prepare students to work within the aerospace industry as the "*Apollo* generation" reaches retirement.

1.3.2 OSIRIS-3U Mission Objectives

Primary Objectives

1. Provide *in situ* and remote sensing measurements of the stimulated (heated) ionosphere produced by ground-based heaters
2. Characterize the spatial extent of the heated region "bite out"
3. Correlate *in situ* and remote sensing heated ionosphere measurements with ground-based measurements including incoherent scatter radars and ionosondes
4. Develop the aerospace workforce by training students in space systems engineering through hands-on projects

Secondary Objectives

1. Investigate anomalous electron number density enhancements in the nighttime ionosphere
2. Correlate *in situ* and remote sensing measurements at Arecibo's conjugate point with heating events

1.4 Thesis Contributions

As a result of this effort, a system model of the OSIRIS-3U system was developed to understand and define mission-level requirements and flow those to the instruments and subsystems. A subset of the mission requirements were applied to two high altitude balloon missions demonstrating key mission technologies. These missions followed systems engineering principles discussed in this thesis.

Further, the plasma impedance probe was incrementally developed including characterization testing. This thesis presents that work and the path forward to instrument testing in a plasma chamber.

1.5 Organization

This thesis overviews the overall OSIRIS-3U CubeSat system design. The instrument requirements are a major driver in developing this system; as such, the mission science including the theory of Langmuir and plasma frequency probes is discussed. The science mission is modeled to develop high level mission requirements. The following chapters are organized as follows:

- Chapter 2 overviews the OSIRIS-3U science mission;
- Chapter 3 overviews some fundamentals of systems engineering;
- Chapter 4 provides high level modeling and analysis for the system design;
- Chapter 5 discusses the plasma impedance probe design and test;
- Chapter 6 discusses the OSIRIS-3U system development approach; and
- Chapter 7 is the conclusion and future work.

Chapter 2

OSIRIS-3U Science Fundamentals

The atmosphere is divided into layers based on temperature gradient. Figure 2.1 shows the profile of temperature as a function of altitude. The bottom layer, which extends upward from the Earth's surface, is defined as the troposphere. This bottom layer of the atmosphere is where weather, as most people know it, occurs.

The temperature in this layer decreases up to the tropopause, defined as the inflection point where the temperature again begins to rise. This region where the temperature again rises is the stratosphere and the temperature increase is attributed to the increased energy absorption of solar radiation by ozone. Above these layers are the mesosphere and thermosphere. In the thermosphere, the local temperature again begins to increase. The atmosphere in this layer is dominated by plasma formed by the atmosphere absorbing energy from the solar wind.

Plasma is the fourth state of matter in which electrons are stripped from atoms creating a positively charged ion and a disassociated electron on the microscopic scale, while maintaining neutrality at the macroscopic scale. This plasma is formed from the absorption of electromagnetic radiation and large particles given off by the Sun. The region of the atmosphere where plasma exists is called the ionosphere.

The charged particle number density, which will be shortened to density, is dependent on altitude and the diurnal cycle. During the day, the plasma density increases due to the direct influence of the Sun and decreases during the night. Figure 2.2 shows the daytime and nighttime profiles of the reference ionosphere. Additionally, these densities differ with the solar cycle. As the Sun becomes more active, the minima and maxima vary. This is shown by the difference between the dotted and solid lines in Figure 2.2.

2.1 Ionospheric Effects on Everyday Life

Ionospheric effects or "events" can affect many elements in everyday life. One of the most famous events occured in March 1989, when a major solar storm caused a massive blackout in Quebéc, Canada [8]. When solar flares hurtle into the Earth's ionosphere, they generate massive ring currents around the Earth. This current couples with large infrastructure elements on the Earth, in this case the electrical distribution grid, and induces currents within them. This is a modern-day concern when considering the design and construction of large cross-country oil pipelines, electric distribution lines, railroads, etc., that traverse long distances.

Some communications systems rely on the plasma in the ionosphere in order to "skip" signals beyond line-of-sight. To understand this, we begin by realizing that plasma is characterized by its plasma frequency, which depends on the density of the plasma. As the density of the plasma increases, the plasma frequency also

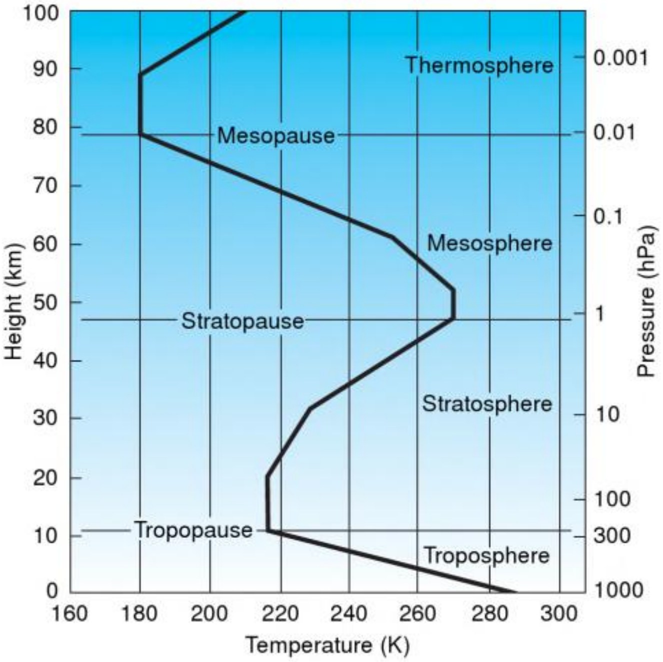


Figure 2.1: Earth's atmosphere profile of altitude versus temperature [7].

increases. Radio transmissions below this plasma frequency that are transmitted at an incidence angle below a critical angle will reflect. As the plasma density changes throughout the day and through the solar cycle, the distances by which transmissions are reflected will vary, e.g., increased plasma density during the day leads to a lower reflection height and skip shorter distance. This is the phenomenon that allows AM radio stations to be heard from across the country or around the world at night when plasma densities are lower and skip occurs at higher altitudes.

Small variations within the makeup of the ionosphere also result in distortion or scintillation of higher frequency signals. This is evident in GPS signals, which use weak transmitters to achieve global coverage, but also applies to satellite uplink and downlink, particularly over the equatorial regions. Scintillation causes phase and amplitude variations as the signal "bounces" around and the path to the ground varies. For GPS signals, this can result in inaccuracies in position determination and, in some cases, complete blackout. This is one of the reasons that commercial airlines avoid flights directly over the north pole even though it may result in a shorter trip: communications and positioning are unreliable, particularly in periods

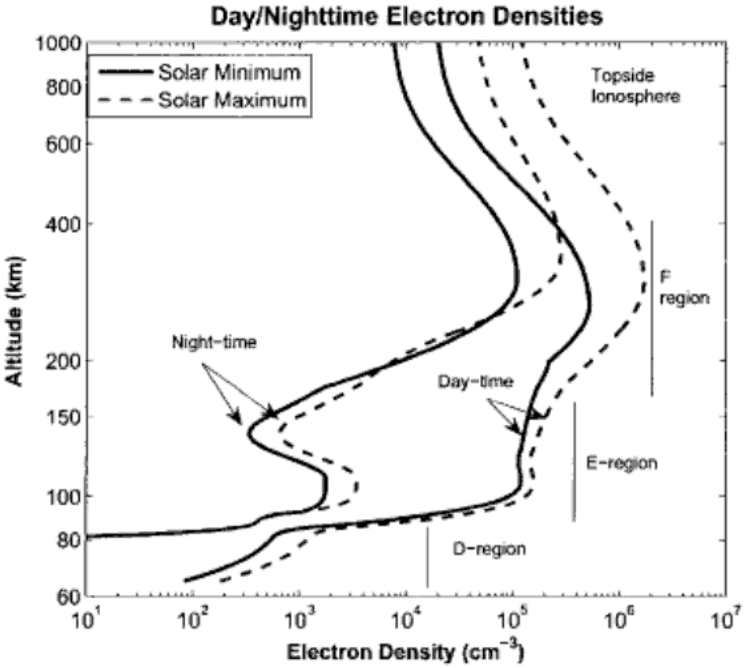


Figure 2.2: Electron number density profile of the midlatitude ionosphere showing diurnal density variations and the layers of the ionosphere [7].

of heightened solar activity.

The aurora is a beautiful display of charged particles entering the ionosphere. The aurora is mostly seen at high latitudes and is the result of charged particles being captured within the Earth's magnetic field and following it down into the lower parts of the atmosphere. These charged particles collide with neutral particles in the atmosphere, emitting photons, or light. Emission could also be caused by charged particles regaining an electron.

2.2 Ionospheric Modification

Ionospheric modification facilities are used to simulate the effects of solar flares on the Earth's ionosphere. These facilities include the High Frequency (HF) heater at the Arecibo Observatory, expected to be completed within the next year, and operational heaters such as the High Frequency Active Auroral Research Program (HAARP) in Alaska, the European Incoherent Scatter Scientific Association (EISCAT) near Tromsø, Norway, and the Sura Ionospheric Heating Facility in Russia. A summary of the capabilities of these facilities is listed in Table 2.1.

Table 2.1: Ionospheric modification facility specifications.

	Arecibo	HAARP	EISCAT	Sura
Location	018°20′37″ N 066°45′13″ W	062°23′28″ N 145°07′57″ W	069°35′11″ N 019°13′38″ E	056°13′04″ N 046°10′17″ E
Lower Frequency	4.65 MHz	2.80 MHz	3.85 MHz	4.30 MHz
Upper Frequency	8.35 MHz	9.50 MHz	8.00 MHz	9.50 MHz
Tilt (Beam Steer)	None	±30	±30 (N/S)	±40 (Mag. N/S)
ERP	90–240 MW	400–3000 MW	300 MW	80–280 MW
Beam width	6.5–10.1	4.5–15	14.5 (avg)	8 (avg)
Antenna Gain	22.1–22.65 dBi	20–31 dBi	24 dBi (avg)	26 dBi (avg)
Pulse Duration	TBD	15 μs	20 μs	40 μs

Treating the Arecibo Observatory as the primary heater for the OSIRIS-3U mission drives the launch date and overall mission duration requirements. The National Science Foundation (NSF), which funds the Arecibo Observatory, has provided yearly funding to the HF facility to support two weeks of heater operation a year. The operating time is constrained by the expense of generating the power required to run the high power transmitters in the HF facility. This limited operating period drives a requirement for the OSIRIS-3U mission to plan for an intensive two-week science period.

Additionally, the HF antenna contains elements that are susceptible to high winds. Arecibo is located in an area prone to hurricanes, so, during hurricane season from June through the end of November, the HF antenna is expected to be stowed for safe keeping. Assuming two months to bring the station from stowed condition to operating condition and allowing another four-month period during which the two-week operating window would fall, drives a requirement for a minimum mission lifetime of six months to ensure that the CubeSat will be operational during the periods that the HF heater is operational.

2.3 Background of OSIRIS-3U Instrumentation

The OSIRIS-3U mission will observe the results of ionospheric modification events with a complement of onboard instruments. These instruments include remote sensing and *in situ* instruments provided by the Naval Research Laboratory, The Aerospace Corporation, and Penn State. These instruments include a CERTO (Coherent Electromagnetic Radio Tomography) beacon, a radio occultation experiment built on a NovAtel GPS receiver, and a Langmuir probe (or the Hybrid Plasma Probe, if flight ready), respectively.

2.3.1 Radio Tomography

Radio tomography techniques measure integrated electron density along vertical or oblique paths from orbital transmitters to ground-based receivers [9][10][11]. Electron density measurement using this technique is known as total electron content (TEC). The orbital segment transmits two phase-related frequencies (f_1 and f_2). Assuming these transmitted frequencies are much higher than the plasma resonance frequency, a fundamental property of a plasma, it can be assumed that the refraction and magnetic field effects are negligible making the effects on the signals dependent on plasma density.

2.3.1.1 Total Electron Content

The received phase of a signal is given by the equation,

$$\Phi_1(f_1) = \frac{2\pi f_1 D}{c} - \frac{\pi\kappa}{f_1 c}\int n_e(s)ds + \eta_1, \tag{2.1}$$

where

$$\kappa = \left(\frac{e}{2\pi}\right)^2 \frac{1}{m_e\epsilon_0}, \tag{2.2}$$

D is the path distance between the transmitter and receiver, and η is the built-in hardware bias (or offset). Since the transmitted frequencies are phase related, f_2 can be scaled by the ratio of the two frequencies and the difference found resulting in

$$\Phi_{12} = \Phi_1 - \Phi_2\frac{f_1}{f_2} = \frac{\pi\kappa\left(f_1^2 - f_2^2\right)}{f_1 f_2^2 c}\int n_e(s)ds + \eta_{12}, \text{where } \eta_{12} = \eta_1 + \eta_2\frac{f_1}{f_2}. \tag{2.3}$$

TEC can then be found from

$$\int_R^T n_e(s)ds = \frac{f_1 f_2^2 c}{\pi\kappa\left(f_1^2 - f_2^2\right)}\left(\Phi_{12} - \eta_{12}\right) = \Psi_{12}\Phi_{12} - \Gamma_{12}\mathrm{m}^{-2}, \tag{2.4}$$

where

$$\Psi_{12} = \frac{f_1 f_2^2 c}{\pi\kappa\left(f_1^2 - f_2^2\right)} \tag{2.5}$$

is TEC ambiguity and

$$\Gamma_{12} = \frac{f_1 f_2^2 c}{\pi\kappa\left(f_1^2 - f_2^2\right)}\eta_{12} \tag{2.6}$$

is TEC system bias error associated with hardware characteristics.

The result here is a measure of the total electron content along the path from the satellite to one ground station. Radio tomography uses a plurality of receivers aligned along the path of a satellite. Each receiver makes measurements of TEC as the satellite passes overhead as depicted in Figure 2.3.

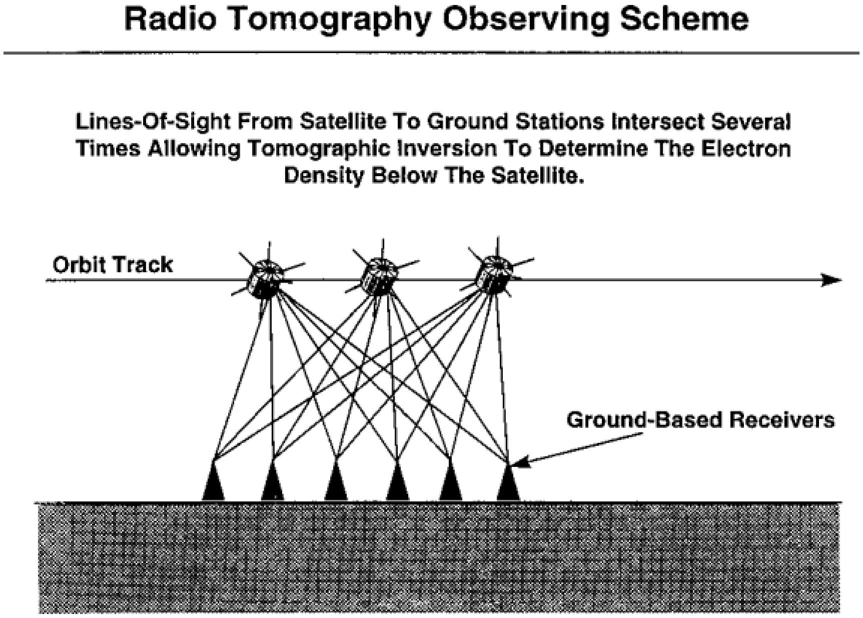

Radio Tomography Observing Scheme
Lines-Of-Sight From Satellite To Ground Stations Intersect Several Times Allowing Tomographic Inversion To Determine The Electron Density Below The Satellite.
Orbit Track
Ground-Based Receivers

Figure 2.3: Using multiple receivers on the ground, line-of-sight rays from the satellite intersect, allowing computer inversion to be performed to determine the electron density [9].

2.3.1.2 Computerized Ionospheric Tomography Processing

Raymund *et al.* [12] demonstrated the process to transform each ray path into an electron density map through a process called Computerized Ionospheric Tomography (CIT). Each TEC path is assumed to be the summation of many smaller integrals along the path such that

$$y_i = \int_{P_i} N(s)ds = \sum_{j=1}^{N} x_j \int_{Pi} A_{ij}(s)ds, \tag{2.7}$$

where P_i refers to a single path and $A_{ij}(s)$ is the length of the ith path that falls into jth pixel in the reconstruction. This leads to

$$y_i = \sum_{j=1}^{N} A_{ij}x_j + e_i \; i = 1, ..., M, \tag{2.8}$$

where y is the known array of TEC data and x_j the electron density in each pixel. The image is thus the solution of this matrix. An example of the resulting image is shown in Figure 2.4, which shows a contour plot of electron densities in an

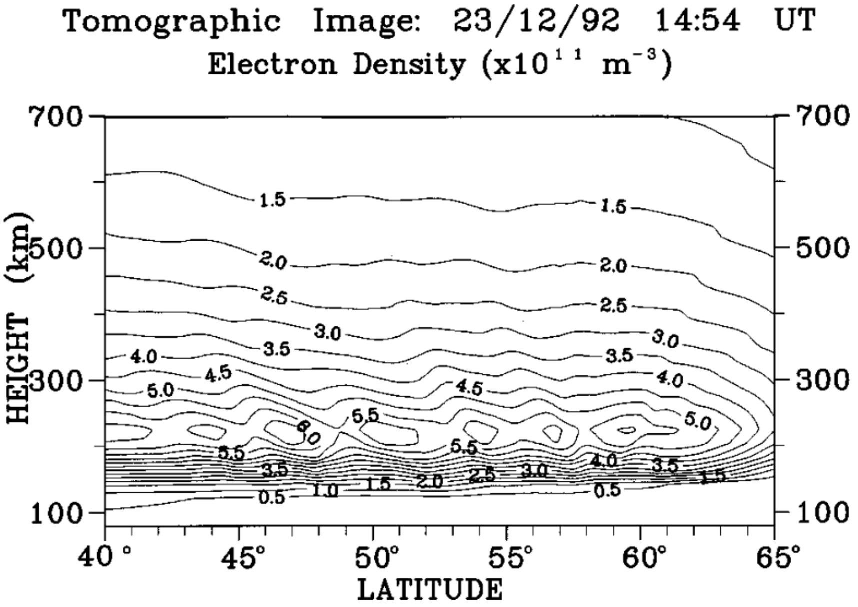


Figure 2.4: Tomographic image showing a traveling ionospheric disturbance (TID) in the winter daytime ionosphere over the United Kingdom [9].

December 1992 experiment observing traveling ionospheric disturbances.

2.3.2 GPS Occultation

GPS occultation experiments have been previously conducted on a number of satellites and constellations. They have recently been proposed as part of the GEOScan system to make use of extra space on the Iridium 2 constellation. GPS occultation monitors the L1 and L2 signals, specifically monitoring the phase difference of these signals.

Occultation measurements require a lock on a GPS satellite as it occults, or passes behind the limb of the Earth as seen by the monitoring satellite. This requires that the GPS satellites appear behind the monitoring satellite. As such, the GPS antenna on OSIRIS-3U is located on the aft face of the CubeSat.

TEC measurement by an occultation experiment is similar to the beacon experiment. The difference is that the lines of integration are horizontal measures of TEC rather than vertical from beacon measurements. Further, a vertical profile can be derived from the TEC measurement paths using Abel inversion [13][14].

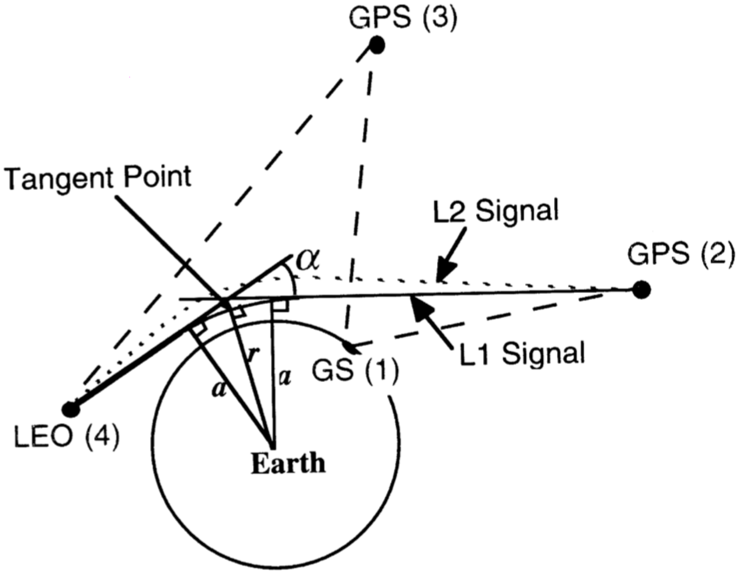


Figure 2.5: GPS occultation schematic showing the GPS receiver occulting with respect to a satellite in LEO. The resulting vertical profile describes the bending point [13].

2.3.3 Langmuir Probe Theory

A Langmuir probe is a plasma diagnostics tool that applies a DC voltage bias to a probe immersed in plasma and measures the current of the attracted ions and/or electrons at that given voltage. Two major theories, Child–Langmuir law and orbital-motion-limited (OML) theory, describe the current collection properties of the Langmuir probe. The determination of which theory to utilize is based on the probe size in relation to the plasma's Debye length.

2.3.3.1 Debye Length

Debye length is given by

$$\lambda_D = \sqrt{\frac{\epsilon_0 k T_e}{n e^2}}, \tag{2.9}$$

where kT_e is the thermal energy with the electron temperature being used because the electrons, being more mobile than ions, typically move to create a surplus or deficiency of negative charges.

OML theory is the focus of this discussion since it relies on probes for which

$r \ll \lambda_D$. Likewise, Child–Langmuir law probes require $r \gg \lambda_D$. Langmuir probes utilizing OML theory are the most commonly used on CubeSats due to size restrictions of the satellite itself making a smaller probe much more desirable.

2.3.3.2 Current Collection Characteristics

As the voltage on the Langmuir probe is swept from negative to positive, the collected current goes through three distinct domains. These regions, shown in Figure 2.6, are the ion saturation, electron retardation, and electron saturation regions. Each region is governed by different equations to determine the collected current.

Mott–Smith and Langmuir [16] derived the generalized equations to describe current collection from spherical, flat plate, and cylindrical collectors. In the electron saturation region, they derive the expression for a cylindrical probe as

$$I_e = An_e q \frac{2}{\sqrt{\pi}} \sqrt{\frac{qV}{2\pi m_e}}, \tag{2.10}$$

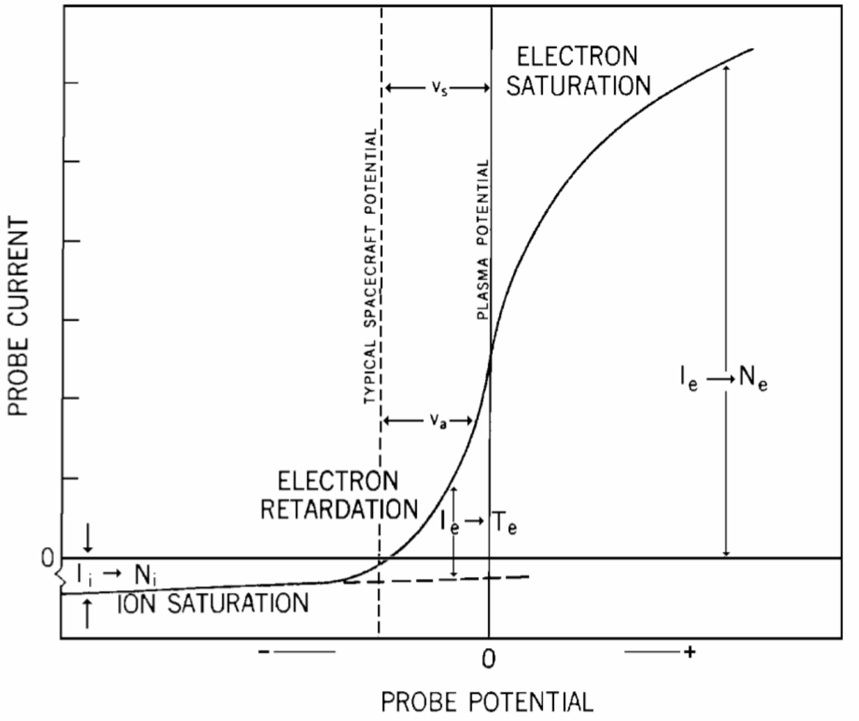


Figure 2.6: Langmuir probe current profile [15].

where

I_e is the collected current from a cylindrical probe,

A is the surface area of the probe,

n_e is the electron number density, and

V is the voltage applied to the probe.

This equation is valid while the applied voltage $V > V_{\text{plasma}}$. Solving Equation 2.10 for electron density yields

$$n_e = \frac{\pi I_e}{Aq}\sqrt{\frac{m_e}{2qV}}. \tag{2.11}$$

With all other parameters known, measuring I_e with the probe biased in the electron saturation region returns the electron density.

The electron retardation region is defined as the portion of the curve between the floating potential and plasma potential where the floating potential is defined as the zero current crossing point. This region is dominated by the electron temperature T_e. The characteristic equation for I_e in the electron retardation region is

$$I_e = Aqn_e\sqrt{\frac{kT_e}{2\pi m_e}}e^{\frac{qV}{kT_e}}. \tag{2.12}$$

The electron temperature can be solved for and is given as

$$T_e = \frac{q}{k}\frac{V_2 - V_1}{\ln(I_2) - \ln(I_1)}, \tag{2.13}$$

where (I_1, V_2) and (I_1, V_2) are points on the curve within the electron retardation region [17].

Finally, the ion saturation region (positively charged ions) is governed by more than electrostatic interactions. Since ions are relatively slow moving, the ion saturation region adds a term for the spacecraft velocity to account for collisions between ions and the collecting surface. The ion current I_i is thus given by

$$I_i = Aqn_iv_i\frac{1}{\sqrt{\pi}}\sqrt{1 + \frac{kT_i}{m_iv_i^2} + \frac{2qV}{m_iv_i^2}}, \tag{2.14}$$

where v_i is assumed to be the ion velocity, which is in this case dominated by the spacecraft velocity, v_{sc}. Solving Equation 2.14 for n_i, the ion density is

$$n_i = \frac{\pi I_i}{qA}\sqrt{\frac{m_i}{m_i v_{\text{sc}}^2 + kT_i + 2qV}}. \tag{2.15}$$

While the ion temperature is assumed to be the same as electron temperature for low altitudes in the ionosphere, this assumption is not valid for orbital instruments. Thus, either another instrument is required to provide the ion temperature or ionosphere models can be used to provide an estimate of the temperatures.

2.3.4 Plasma Impedance Probe Theory

When excited, a plasma supports a number of wave modes and thus has several resonance frequencies. To describe these, the assumption is first made that, within a plasma, the ions, being a much heavier than electrons, are stationary and the electrons are mobile. Additionally, in the development of this discussion, thermal motion can be neglected.

The plasma fundamental frequency is illustrated in Figure 2.7 and comes from the interaction of charges in the absence of a magnetic field. Assuming a background of stationary ions, excited electrons move through the ion field while electrostatic interaction between the ions and electrons attract the electrons to the ions. As the electron moves past the ion, an increasing electric field develops that pulls the electron back towards the ion. As the electron builds momentum moving back towards the ion, it will overshoot the ion and the process repeats thus causing the plasma frequency [18].

Thus, the plasma frequency can be expressed in terms of electron number density n_e, the fundamental charge of an electron, e, permittivity of free space, ϵ_0, and, since the electron is the particle in motion, the mass of an electron, m_e, i.e.,

$$f_p = \frac{\omega_p}{2\pi} = \frac{1}{2\pi}\sqrt{\frac{n_e e^2}{\epsilon_0 m_e}}, \tag{2.16}$$

and approximated as

$$f_p \approx 9\sqrt{n} \text{ for } n \text{ in m}^{-3}. \tag{2.17}$$

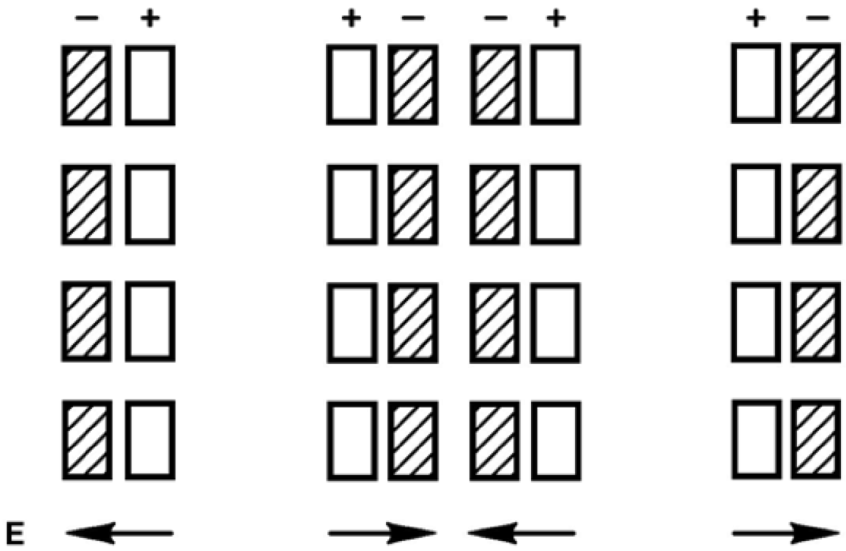


Figure 2.7: Plasma wave oscillations [18].

In the presence of a magnetic field, given an initial condition, the electron will again oscillate, this time around the magnetic field line itself. The fundamental frequency in this case is dependent on the strength of this magnetic field, thus this cyclotron frequency is expressed as

$$f_c = \frac{\omega_c}{2\pi} = \frac{eB_o}{2\pi m_e}, \tag{2.18}$$

where B_0 is the local magnetic field strength.

Finally, the upper hybrid frequency is the primary component where the cyclotron and plasma frequencies occur together, i.e.,

$$f_{\mathrm{uh}}^2 = f_p^2 + f_c^2. \tag{2.19}$$

Since the Earth's magnetic field is well known, once the upper hybrid frequency is determined, the electron density can be found by combining Equations 2.16, 2.18, and 2.19. Solving for n_e results in

$$n_e = \epsilon_0 \left(\frac{f_{\mathrm{uh}}^2 4\pi^2 m_e}{e^2} - \frac{B_0^2}{m_e} \right). \tag{2.20}$$

Plasma impedance probes are immersed in a plasma with a low amplitude

RF signal applied to a conductive probe. While applying the RF signal to the probe, measurement of the reflected signal enables generation of a plot of the real and imaginary portions of the plasma impedance. A typical plot is depicted in Figure 2.8. The major resonances evidenced can be seen as local maximums in the impedance measurements and drastic phase shifts. The upper hybrid frequency is most easily measured as both the maximums in the impedance plot and the zero crossing in the phase plot.

2.4 OSIRIS-3U Science Tie-in

Figure 2.9 shows how the three instruments on OSIRIS-3U will work together to map the extent and effects of a heating event. For the purposes of this discussion, it is assumed that operations are occurring above the Arecibo HF heater.

The HF heater is turned on before OSIRIS-3U enters the region of interest. The

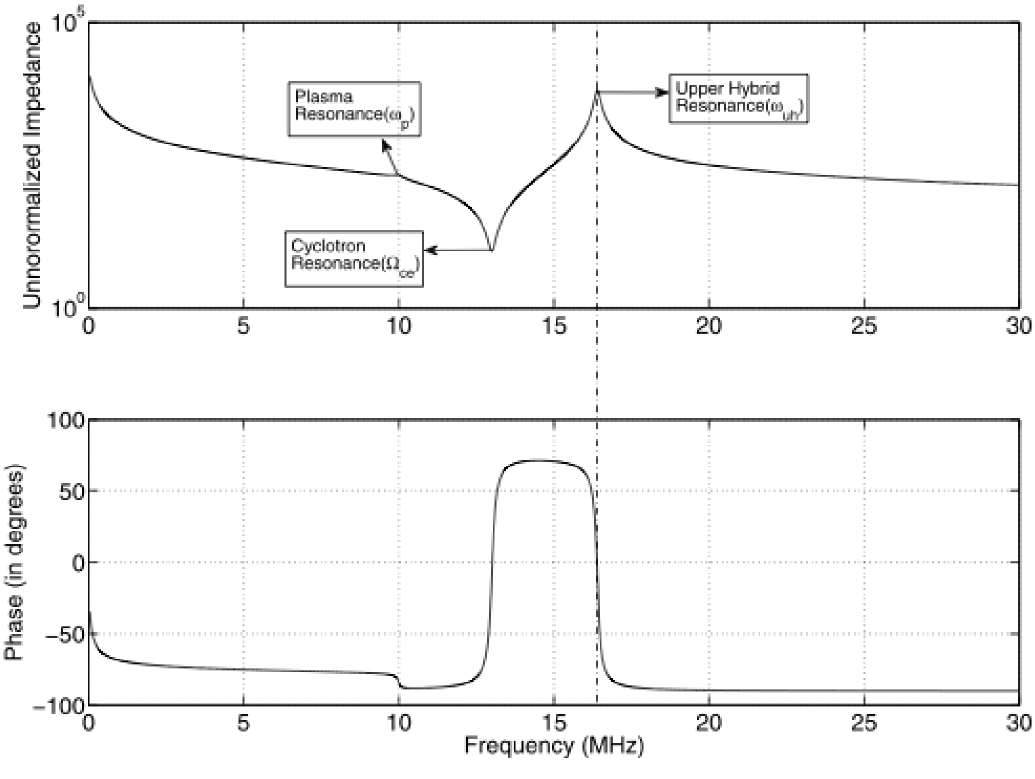


Figure 2.8: Plots of the real and imaginary portions of the plasma impedance as a probe has an RF signal sweeping in frequency applied to the probe [19].

region of space is assumed to be allowed to reach steady state before OSIRIS-3U flies through it. It is likely that the heating will occur below the altitude at which OSIRIS-3U will orbit (expected to be between 400 and 600 km). The energy from the heating event will follow the local magnetic field lines, which, above Arecibo, make a 45-degree angle with respect to the vertical.

As OSIRIS-3U travels through the this region, it will take *in situ* measurements of the local plasma density and temperature using the swept-bias Langmuir probe. While overflying the area, the onboard beacon transmits at 150 and 400 MHz, which is received and analyzed on the ground. Ground receivers records the phase difference between the 150- and 400-MHz frequencies, thereby enabling derivation of scintillation and TEC along the path. These derived parameters will be used to create a two-dimensional image of the heated region using computerized ionospheric tomography (CIT) as described in Section 2.3.1.2. Finally, the GPS occultation experiment will produce one or more vertical profiles over the area as

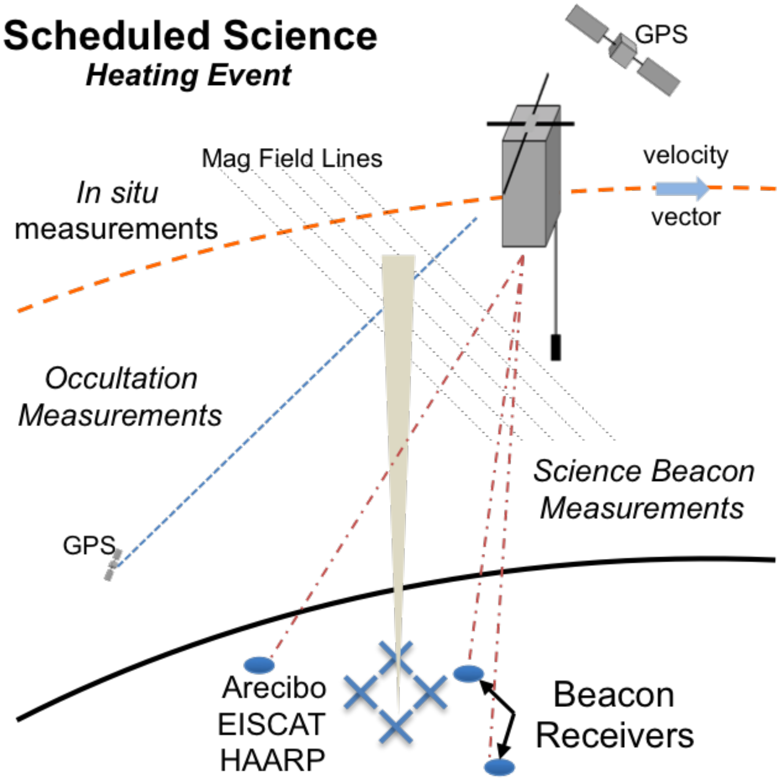


Figure 2.9: Schematic of the OSIRIS-3U science mission showing the heating of and relationship between the various onboard measurements.

it tracks GPS satellites occulting behind OSIRIS-3U and on the other side of the heated region. This data will also be inputted to the CIT processing to increase the fidelity of the generated images [9].

Chapter 3

Fundamentals of Systems Engineering

With the boom in technology, the last century has seen a shift in scope and complexity of projects. Projects and systems have grown in size, complexity, interdependence, and interactions that do not always manifest themselves until elements are integrated. As systems grow in complexity, so do these interactions. This has driven the need for a change in how project development is approached. Today's complex system projects require interdisciplinary teams that are ever-growing in size and in geographic dispersion. It is no longer possible for one person or even a small group of people to understand all of the intricacies of a system. Team member interaction and communication in addition to efficient capture and sharing of information and data are essential for these projects to succeed. These dynamics of complex system development have driven the growth of a new discipline widely known as systems engineering (SE) with its practitioners called systems engineers or SysEs.

The function of systems engineering is to guide the engineering of complex systems and to form bridges across traditional engineering disciplines that are designing the individual systems elements that must interact with each other. Systems engineering tasks include:

- Leading the development of the systems architecture;
- Defining, verifying, and validating system requirements, and their flow down

through the product hierarchy;

- Evaluating design tradeoffs from trade studies;
- Assuming responsibility for guiding the integration and test phases of the project;
- Balancing technical risk between systems and performing failure mode analysis;
- Defining and assessing interfaces; and
- Providing oversight of verification and validation activities.

Systems engineering itself is defined many ways:

> "Systems engineering is a robust approach to the design, creation, and operation of systems. In simple terms, the approach consists of identification and quantification of system goals, creation of alternative system design concepts, performance of design trades, selection and implementation of the best design, verification that the design is properly built and integrated, and post-implementation assessment of how well the system meets (or met) the goals. The approach is usually applied repeatedly and recursively, with several increases in the resolution of the system baselines (which contain requirements, design details, verification procedures and standards, cost and performance estimates, and so on)." —National Aeronautics and Space Administration SP-6105 [20]

> "Systems engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem."
> —International Council on Systems Engineering [21]

> "Systems Engineering is defined as an interdisciplinary approach and means to enable the realization of successful systems. It focuses on

> holistically and concurrently understanding stakeholder needs; exploring opportunities; documenting requirements; and synthesizing, verifying, validating, and evolving solutions while considering the complete problem, from system concept exploration through system disposal."
> —System Engineering Body of Knowledge [22]

While differing, these definitions capture the essence of systems engineering as a rigorous, disciplined methodology for designing products using a multidisciplinary approach to define and understand system requirements early on in the project. Figure 3.1 illustrates the importance of clearly defining requirements and how different disciplines may approach a problem. While asked to put a swing in a tree, the layperson understands the general concept from their own experience; someone that has not seen a swing will approach the problem differently and end up at a different solution.

To ensure that this goal is met, systems engineers follow an interdisciplinary process that ensures that the customer's needs are satisfied throughout a system's entire life cycle. This process is comprised of the following seven tasks.

1. *State the problem.* Stating the problem is the most important systems engi-

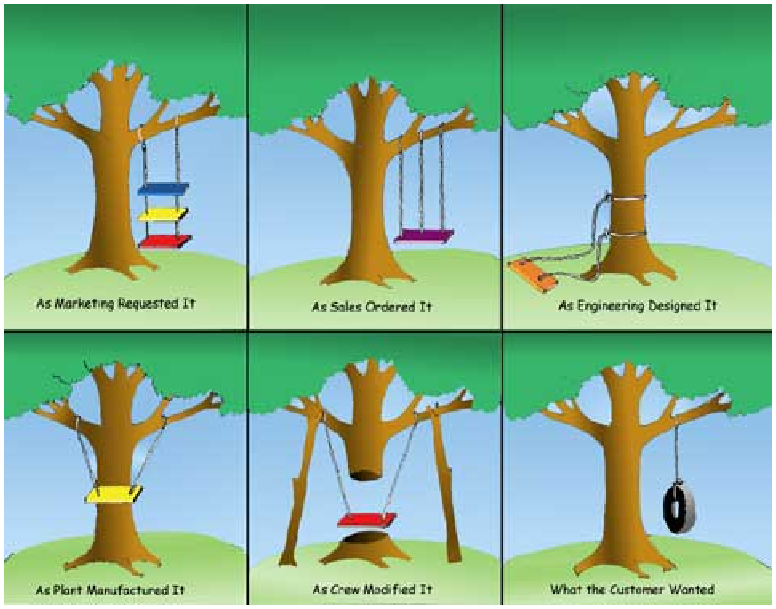


Figure 3.1: Without clearly defined requirements, a diverse team can see the problem very differently [thingdesigner.com].

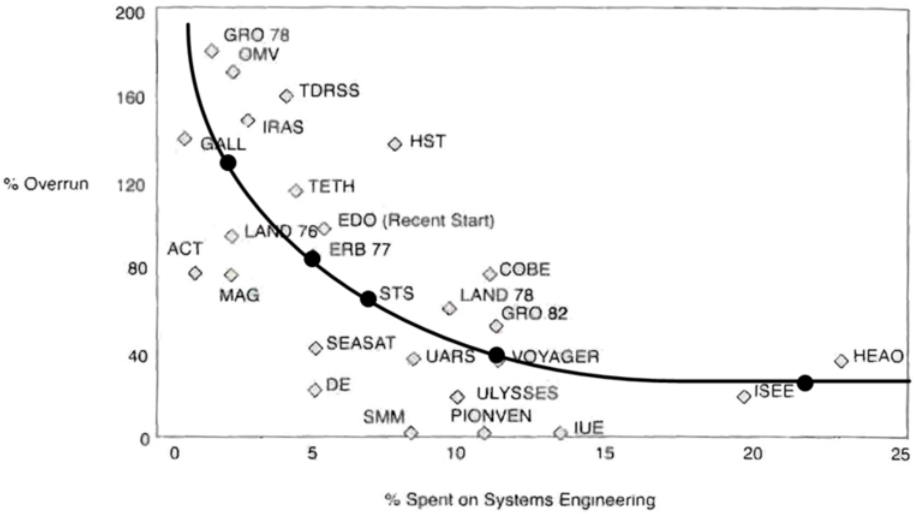
neering task. It entails identifying customers, understanding customer needs, establishing the need for change, discovering requirements, and defining system functions.

2. *Investigate alternatives.* Alternatives are investigated and evaluated based on performance, cost, and risk.
3. *Model the system.* Running models clarifies requirements, reveals bottlenecks and fragmented activities, reduces cost, and exposes duplication of efforts.
4. *Integrate.* Integration means designing interfaces and bringing system elements together so they work as a whole. This requires extensive communication and coordination.
5. *Launch the system.* Launching the system means running the system and producing outputs — making the system do what it was intended to do.
6. *Assess performance.* Performance is assessed using evaluation criteria, technical performance measures and measures — measurement is the key. If you cannot measure it, you cannot control it. If you cannot control it, you cannot improve it.
7. *Re-evaluation.* Re-evaluation should be a continual and iterative process with many parallel loops.

With the ever increasing number of complex systems, misunderstanding or misinterpretation of the problem and requirements ultimately leads to non-compliant system implementations and/or subsequent multiple iterations prior to achieving compliance, which translate into extended system development time and increased cost to complete. Figure 3.2 shows the impact that early investment in developing requirements on budget overruns. Projects that spent more in the early project phases show a reduction in overall budget overruns [23][24].

3.1 Project Life Cycle

There are a many project life cycle models, the selection of which model to follow is very situation-dependent and can be influenced by the stakeholder, team dy-



*Source: Werner M. Gruhl, Chief Cost & Economics Analysis Branch, NASA Headquarters

Figure 3.2: In the world of budget overruns, early investment in requirements development makes a significant impact on minimizing overruns [22].

namics, or project scope. NASA breaks down the mission life cycle into six phases, labelled A through F. Table 3.1 shows a summary of the NASA-defined mission phases found in SP-6105 Systems Engineering Handbook [20]. Project assessment gates are provided by independent reviews conducted throughout the project life cycle. These reviews must meet set criteria in order to proceed to the next phase. These reviews, through stakeholder and user participation, assure appropriate architecture and requirements are defined; good design practices are implemented; lessons learned from similar projects are taken into account; and cost and schedule are within predefined limits.

3.1.1 Spiral Model

The spiral model for project execution emphasizes the iterative nature of systems. Figure 3.3 shows the development of a project by considering layers of the spiral. The spiral is divided into four quadrants to show the type of work being done within each cycle. Quadrant 1 is determining the project objectives and generating concepts and constraints on the new iteration. Quadrant 2 identifies and resolves

Table 3.1: NASA definition of mission phases.

Mission Phase	Name	Description
A	Concept & Technology Development	To determine the feasibility and desirability of a suggested new major system and establish an initial baseline compatibility with NASA's strategic plans
B	Preliminary Design & Technology Completion	To define the project in enough detail to establish an initial baseline capable of meeting mission needs
C	Final Design & Fabrication	To complete the detailed design of the system (and its associated subsystems, including its operations systems), fabricate hardware, and code software
D	System Assembly, Integration & Test, Launch	To assemble and integrate the products and create the system, meanwhile developing confidence that it will be able to meet the system requirements; conduct launch and prepare for operations
E	Operations & Sustainment	To conduct the mission and meet the initially identified need and maintain support for that needs
F	Closeout	To implement the systems decommissioning/disposal plan developed in Phase C and analyze any returned data and samples

risks, and involves evaluating the concepts and identifying and managing the risks to the implementation of those concepts. Quadrant 3 develops and builds up the chosen alternative(s) into functional models. In the final Quadrant 4, a plan is established for the next iteration. Progression from one iteration to the next is allowed through successful progress reviews.

This model parallels the NASA mission phases as each phase can be represented by an iteration in the spiral. The inner most iteration is Phase A, where the concept is developed and requirements are gathered. The establishment of a baseline in Phase B parallels the development of Prototype 1, development of the final design in Phase C parallels the next iteration into Prototype 2, and Phase D is the final build and fabrication through release or launch.

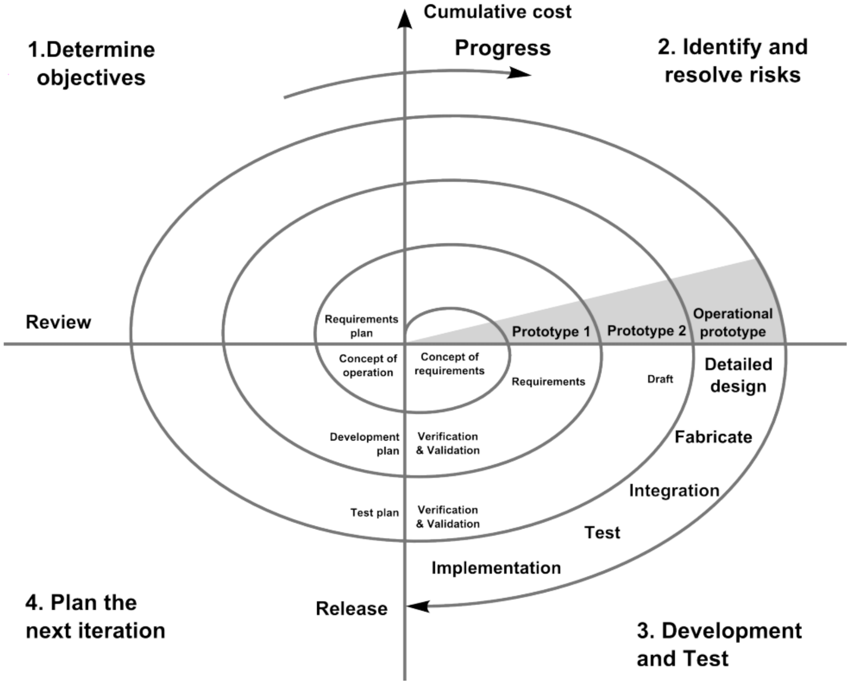


Figure 3.3: Spiral model for project execution [25].

3.1.2 Vee Model

Another model for system design considers design in the shape of a "V". Hence this model is clled the Vee model. In this model, the left leg of the "V" is for project decomposition, breaking down the project in a hierarchical structure from the overall system concept to the subsystems that make it up. These subsystems subsequently can be broken down into assemblies and components. Decomposition involves flowing requirements down from the system level to these lower levels and defining a system to meet these requirements. As the system is decomposed and designed, verification and validation are planned at each level. This includes an outline of the tests that need to be performed for each level to assure that the system meets the requirements as specified.

The right hand side of the "V" then involves building and testing the system. At each level, the components, assemblies, and subsystems are tested in accordance with the verification plan or test plan developed during system decomposition. This assures that the incremental products as built meet the design requirements. As each incremental product is tested, it is validated as it is integrated into the

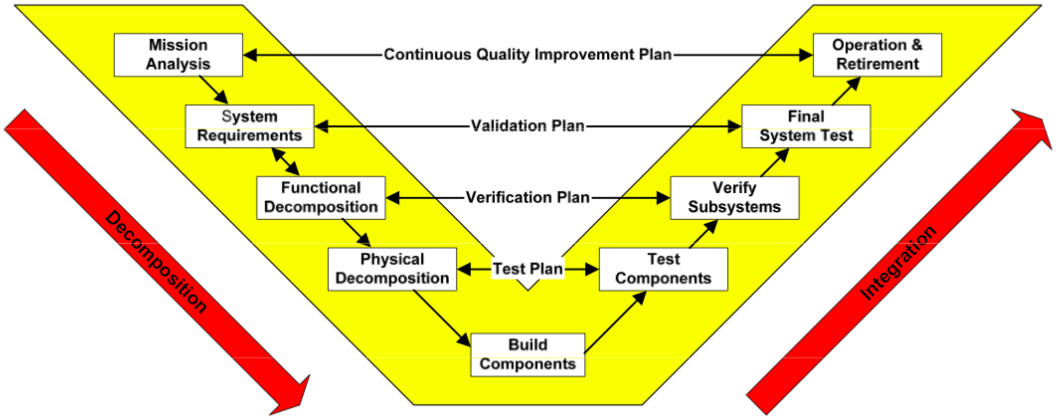


Figure 3.4: Vee model for project execution [26].

next level to assure that the component, as built to the specified requirements, meets the needs of the next higher level.

3.2 Technology Readiness Levels

Technology readiness levels (TRL) are a method for assessing the maturity of a technology and its preparation for use. It allows a common metric that is technology-independent to allow comparisons of competing technologies [27]. It also provides a basis for understanding the technological areas in which further investment is required before inclusion on a flight mission. Figure 3.5 provides a graphical summary of the TRL levels.

A technology starts off at the bottom of the TRL scale as a level 1 technology when it enters the realm of applied development from a purely scientific exercise. TRL 2 and 3 develop the technology to demonstrate concept feasibility. From there the technology is developed with basic breadboard testing on a bench top and relevant environment in TRL 4 and 5 respectively. TRL 6 and 7 demonstrate the system operating in relevant environments at the brassboard and prototype level. Finally, TRL 8 and 9 involve the unit preparing for flight and finally operating continuously on orbit bug free respectively.

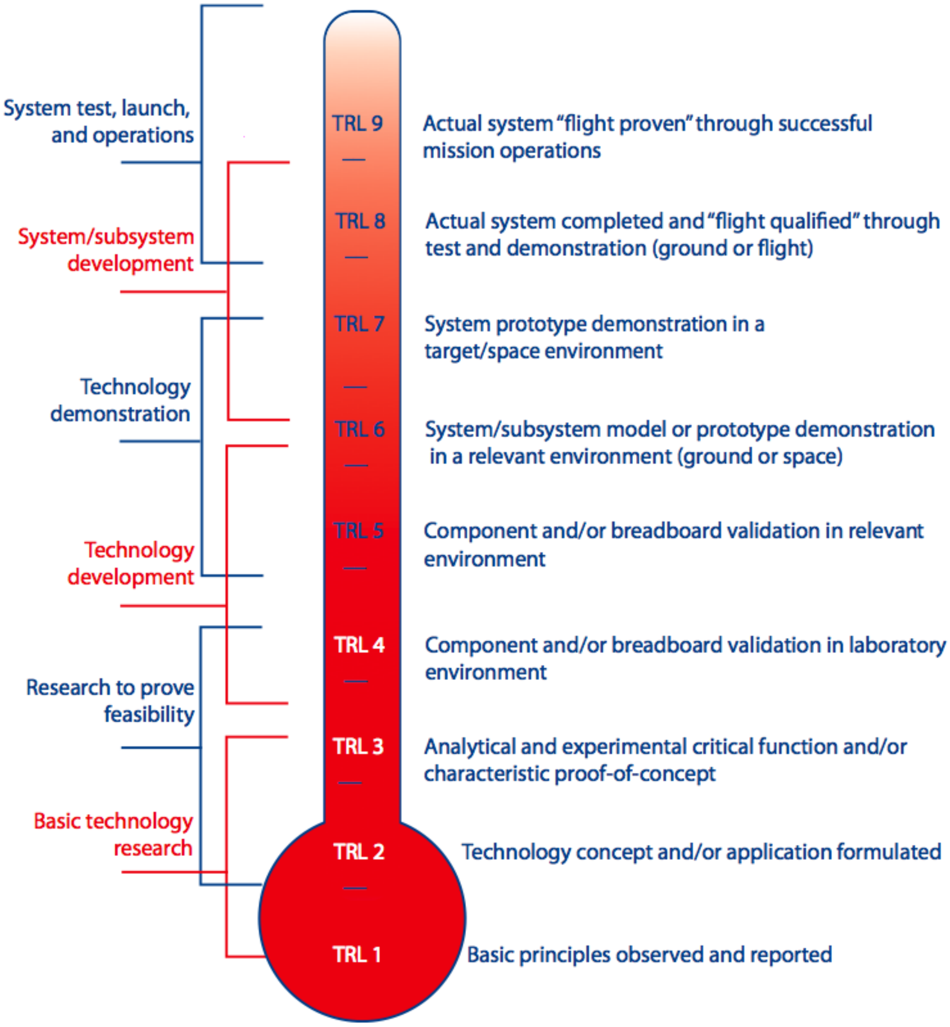

System test, launch, and operations
System/subsystem development
Technology demonstration
Technology development
Research to prove feasibility
Basic technology research
TRL 9
Actual system "flight proven" through successful mission operations
TRL 8
Actual system completed and "flight qualified" through test and demonstration (ground or flight)
TRL 7
System prototype demonstration in a target/space environment
TRL 6
System/subsystem model or prototype demonstration in a relevant environment (ground or space)
TRL 5
Component and/or breadboard validation in relevant environment
TRL 4
Component and/or breadboard validation in laboratory environment
TRL 3
Analytical and experimental critical function and/or characteristic proof-of-concept
TRL 2
Technology concept and/or application formulated
TRL 1
Basic principles observed and reported

Figure 3.5: Summary of technology readiness levels (TRL) [20].

3.3 Fault Modes and Effects Analysis

Fault modes and effects analysis is a tool for systems engineers to evaluate the system design for its susceptibility to failure or to reconcile the root cause of an anomaly. It is important to establish an understanding of the problems that can occur and establish a fault tree early on in the program. This allows the program to access the likelihood of an event occurring and the impact that it has on the final product. In the debugging process when an anomaly does occur, a fault tree assists in the establishment of the root cause of the anomaly, which is required before appropriate corrective actions may be implemented.

Figure 3.6 shows an example fault tree from the NASA Systems Engineering Handbook [20]. Considering a top-level event, other events can logically contribute to the occurrence of the top-level events. Using logical "and" and "or" operator symbols, two (or more) events that must occur simultaneously to cause another to occur are combined with a logical "and". Likewise, two (or more) events that could independently trigger a higher level event can be combined with a logical "or" operator. This leads to the development of a tree structure where the top-level event is a result of lower level event(s), which may be composed of other even lower-level events. This is the development of a *fault tree*.

Many specialized programs exist to build fault trees e.g. [28]. These tools facilitate a layer for predictive purposes where statistical models can be applied to each of these nodes. These models take inputs from test data or design analysis on probabilities of failure.

In the fault tree, an "or" operator shows that a failure in any one of the inputs can cause a failure event mode to occur. This implies an overall series implementation structure. The probability of failure $P_s[F]$ for a series structure can be found from the probability of failure for the individual parts $P_i[F]$ by

$$P_s[F] = 1 - ((1 - P_1[F])(1 - P_2[F]) \dots (1 - P_i[F])). \tag{3.1}$$

Likewise, an "and" operator shows that only the failure of all its inputs will trigger the event. This implies a parallel implementation structure where the system can tolerate the failure of a single leg of the structure and still operate. The probability of failure from this structure is found as

$$P_s[F] = P_1[F]P_2[F] \dots P_i[F]. \tag{3.2}$$

Another way of looking at a system is from the perspective of reliability. That is from the probability of success rather than failure. Since reliability, $P[R]$, is the complement of the probability of failure,

$$P[R] = 1 - P[F] \tag{3.3}$$

can be substituted into Equations 3.1 and 3.2 to yield equations for system reliability.

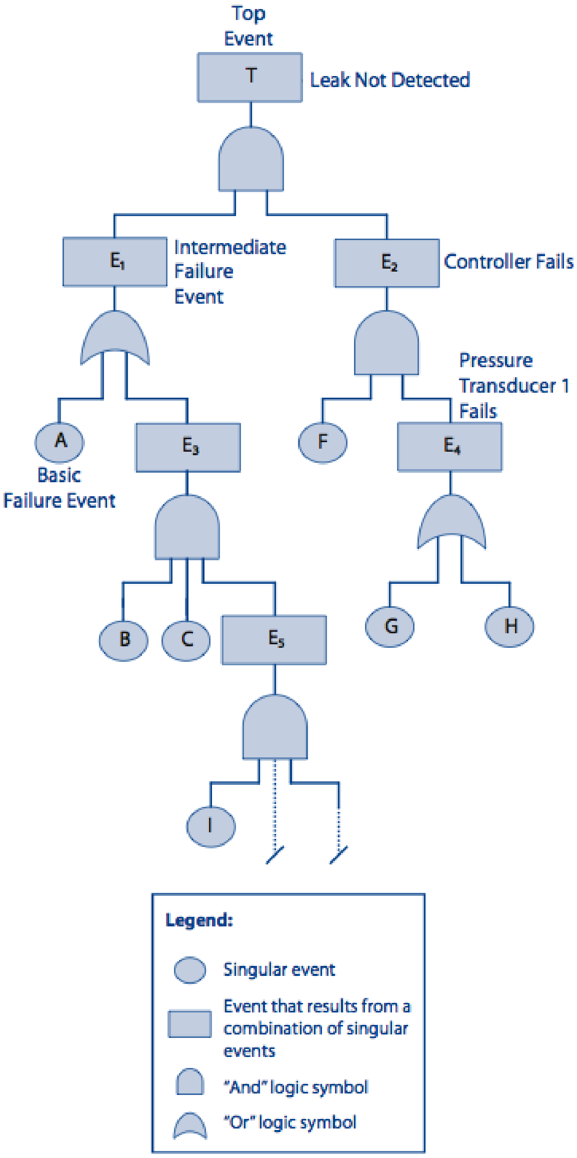


Figure 3.6: Example of a fault mode tree showing the decomposition of a negative event to the potential root causes of the event [20].

Chapter 4

OSIRIS-3U System Model

Spacecraft design is aided by developing a model of the mission environment. Subsequently, the model can be utilized to develop an understanding of the system and to facilitate the development of requirements. Ultimately, the model can be used as a tool during system verification to provide accurate conditions for a hardware-in-the-loop (HIL) simulation at the subsystem or system level.

An OSIRIS-3U model was developed to facilitate the development of CubeSat requirements based on the proposed science mission. This simulation links MathWorks MATLAB, Analytical Graphics, Inc. Satellite Tool Kit (STK), and Penn State's Applied Research Laboratory's Trade Space Visualizer (ATSV) as shown in Figure 4.1. The result is a *Monte Carlo* simulation with MATLAB providing input to an STK environment model. Key parameters from each set of conditions in STK are accessed by MATLAB and stored in a file. ATSV provides an interface to visualize the simulation data and explore it to understand complex relationships within the simulation data.

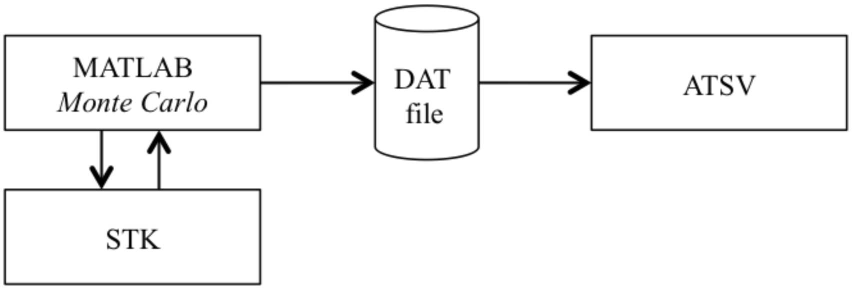


Figure 4.1: OSIRIS-3U system model information flow.

This chapter describes the OSIRIS-3U system modeling effort beginning with an overview of spacecraft operational modes and resource allocation through system budgets. After the budgets are developed, these are used with a *Monte Carlo* simulation to define mission inclination and altitude requirements. Finally, the simulation results that meet the established orbit requirements and mission success criteria define the probability of mission success.

4.1 Operational Modes

OSIRIS-3U operational modes are divided into four modes described below; safe mode with beacon, default mode, science mode, and transmit mode. These modes control the schedule for when each instrument and subsystem are on and active, thus defining the overall power usage and data collection during each mode. These are then taken into account with the respective budgets based on the system progressing through these operational modes.

4.1.1 Safe Mode with Beacon

During safe-with-beacon mode, OSIRIS-3U cycles through a six-minute predetermined basic mode. At the beginning of the cycle, OSIRIS-3U will attempt to turn on attitude determination sensors and get an attitude solution for a minute. Then OSIRIS-3U will enter a beacon-transmit mode for a minute where it will transmit a beacon packet with basic information about its status for debugging purposes. It then will open up its receiver to listen for commands from the ground station for a minute and finally sleep for three minutes. All the while, the flight computer will be polling for housekeeping data every ten seconds.

4.1.2 Default (Sleep) Mode

Default mode handles operations from a predetermined schedule that is uploaded from the ground. This commands OSIRIS-3U to enter modes at specified times as major events come up. This can be switching to science mode for an orbit over a HF heater, or to transmit mode near the Penn State ground station, or shutting down completely before traversing the South Atlantic Anomaly (SAA).

4.1.3 Science Mode

Science mode operates in addition to default mode. The instruments, GPS, Langmuir probe, and science beacon are turned on and operated during this mode. Table 4.1 provides a summary of the timing during a science pass. Times indexed from off a $t = 0$ entrance to the science region.

Table 4.1: Science modes timeline.

Time (min)	Description
–25	Turn on GPS and begin occultation measurements
–5	Turn on Langmuir probe, begin taking LP data and transmitting with the science beacon
0	Enter the science region
1	Exit the science region
5	Turn off LP and science beacon
45	Turn off GPS

4.1.4 Transmit Mode

Transmit mode can be entered from either safe mode or default mode when OSIRIS-3U is over State College, PA. In safe mode, transmit mode will be initiated during the one-minute listening cycle by a command uplinked from the ground station. In default mode, OSIRIS-3U will transition to transmit mode based on a predetermined schedule for when OSIRIS-3U is expected to be within communications range. A single pass over the ground station requires a maximum of 15 minutes of system on time between radio warmup and loss of signal.

4.2 System Budgets

Budgets for data generation and power usage are required for mission planning. The budgets track contributions of each instrument and subsystem and are useful to identify the largest contributors. Often, during the initial spacecraft and mission design process, budgets are used to optimize and allocate finite resource to individual systems. Later in the mission lifetime, these budgets are used to track progress and identify problem areas as systems become more defined and the initial estimates are updated with test data.

4.2.1 Data Budget

Developing the data budget focuses on the two-week science-intensive period during which the Arecibo heater is active. The major contributors to the total data budget during this period are housekeeping data, science data, and a copy of flight software for verification. Table 4.2 summarizes the inputs of the major data contributors.

Table 4.2: Summary of data contributions.

Source	Volume (MB)	Contingency (30%)	Total (MB)
GPS	10.2	3.1	13.3
LP	24.6	7.4	32.0
Housekeeping	12.1	3.6	15.7
Subtotal	46.9	14.1	61.0
Compressed			40.3

The worst case data mode for the Langmuir probe is operating in swept-bias mode. In this mode, a 256-sample voltage sweep is executed eight times a second. With 12 bits for each return current sampled and an additional 8 bits for a voltage counter, that is 20 bits/sample. This leads to a data rate of 41.0 kbps or 24.6 MB for 8 science passes.

From information provided in private communications with Aerospace Corporation personnel responsible for the custom GPS firmware, the GPS operates at a rate of 300 bytes/sec while tracking 8 satellites measuring total electron content (TEC). Additionally, position information is required once a minute and the GPS almanac is required twice during a science pass. This leads to data volumes of 1.26 MB, 8.7 kB, and 7.2 kB for TEC, navigation, and almanac data, respectively, leading to 1.28 MB per science pass. This equates to a total of 10.2 MB for 8 science passes.

The current communications system design supports a downlink rate of 12800 bits/sec (1.6 kB/sec), which results in 25200 seconds or 7 hours of downlink time to downlink the data. This will be used to constrain the allowed inclinations based on available downlink time.

4.2.2 Power Budget

The power budget considers the four major operational modes. These are Safe with Beacon Transmit, Default, Transmit, and Science. Table 4.3 provides a summary of the power requirements for each operational mode. Transmit mode is added on top of either default or safe mode as required, so the power number during transmit mode considers just the operational time during which OSIRIS-3U would be transmitting.

Table 4.3: Summary of the power requirements during each operational mode.

Operational Mode	Total Power (W)	Duty Cycle (%)	Total Energy (Wh)	Total Energy including 30% Margin (Wh)
Science	3.05	100	4.58	5.95
Sleep (Default)	1.01	100	1.51	1.96
Safe	1.20	100	1.80	2.34
Transmit*	7.97	11	1.33	1.73

(*) Power from transmit mode is added to any other operational mode as required.

In parallel to understanding the power budget, power generation is bounded by a worst case analysis. The worst case power generation scenario exists when OSIRIS-3U orbits in the same plane as the Sun as seen in Figure 4.2.

After initialization, the CubeSat is assumed to be in a stable attitude with the +Z face always towards the Earth and the –X face always facing aft or behind the spacecraft. The orbit is broken up into four quadrants. In each quadrant, the angle of incidence on a solar panel is calculated for each of 500 angles per quadrant. These incidence angles are used as a scaling factor for the maximum possible power generated by each face of the CubeSat. This yields the power generated during that step, which is converted to energy assuming each angle represents about 5 seconds of a 90-minute orbit. These are summed up to give the total energy harvested from the solar cells for a single orbit in Wh.

The solar cells used in this calculation are the SpectroLab Triangular Advanced Solar Cells (TASC) with an end-of-life (EOL) efficiency of 21%. The solar cells

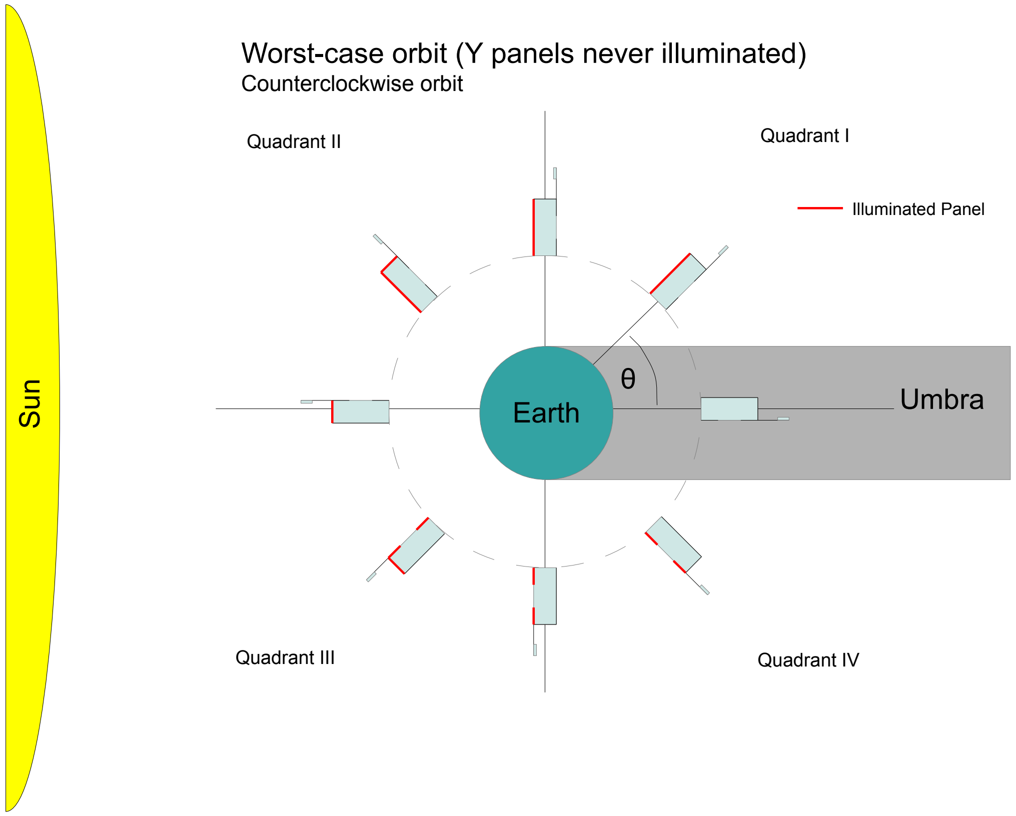


Figure 4.2: Schematic for the worst case power scenario of a CubeSat orbiting in the same plane as the Sun.

are assumed to have a coating of Indium Tin Oxide (ITO) on cover glass that will reduce the overall efficiency by 10% to 19% [29] [30]. The cells are assumed to be at 80 °C, which reduces the output voltage and increases the output current of the cells.

This analysis yields a worst case energy of 4 Wh/orbit, bounding the minimum power expected from the *Monte Carlo* simulation.

The power budget and worst case analysis show an overall energy margin of 1.66 Wh per orbit during sSafe mode and 2.04 Wh in Default mode, making both these modes power positive for a given orbit. While science and transmit modes make a given orbit power negative, the power subsystem is designed to facilitate the additional depth of discharge (DOD) during these orbits. As such, OSIRIS-3U requires under one orbit to fully charge in default mode following a science orbit, the batteries are sized to limit the DOD in support of a minimum of two consecutive science orbits. Additionally, only one science pass is expected in a given night, thus providing adequate time to recharge the system between science orbits.

4.3 *Monte Carlo* Model Development

4.3.1 Definition of Key Parameters

We define the following key parameters:

Total Downlink Time is the total amount of time available to downlink data to the ground station;

On Orbit Lifetime is the length of time before the satellite deorbits due to orbit decay;

Solar Power Available per Orbit is the solar power generated per orbit based on satellite attitude;

Number of Times in the Science Region is the total number of times that the satellite traverses the science region during a two week period; and

Volume of Data Generated is the total data generated that requires downlink back to the Penn State ground station.

4.3.2 *Monte Carlo* Model Inputs

Inputs driving the *Monte Carlo* simulation are altitude and inclination. Any acceptable orbital altitude is assumed to be achievable; therefore, orbital altitude is a uniform random variable with range of 300 to 700 km. Inclination is assumed to be highly dependent on the launch location and thus holds probabilities of realization based on the likelihood of a given launch site as shown in Table 4.4. The orbits are assumed to be nearly circular with uniformly chosen right hand ascensions.

Table 4.4: Potential US launch sites.

Launch Site	Inclination	Probability
Cape Canaveral	28.5°	25%
International Space Station	51.6°	25%
Vandenburg	50–110°	25%
Any	0–100°	25%

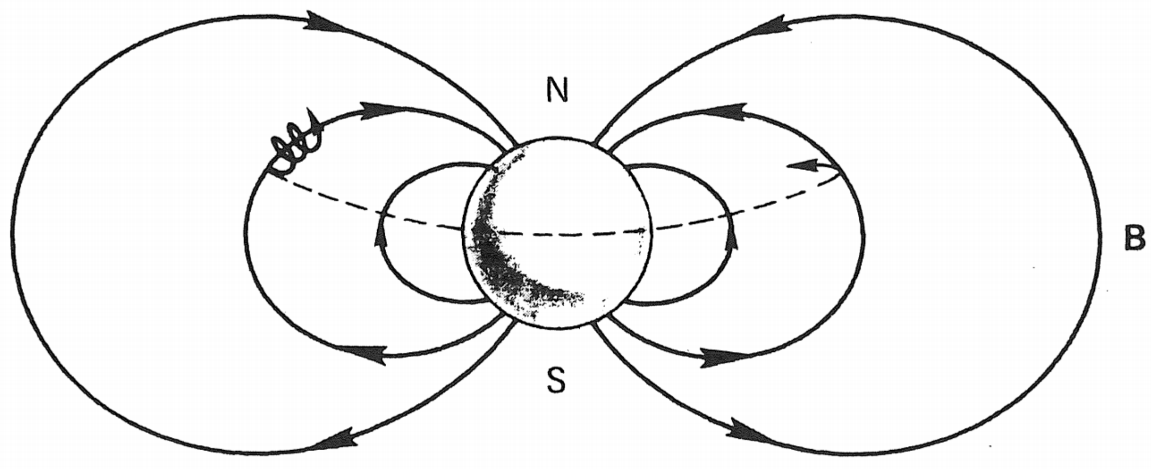


Figure 4.3: Plasma motion in the ionosphere [18].

4.3.3 Modeling the Science Area

The OSIRIS-3U primary science area is located within the area of the ionosphere influenced by the Arecibo heater. As the Arecibo heater heats the ionosphere, the excited plasma will move both perpendicular to the magnetic field lines and along the magnetic field lines as shown in Figure 4.3.

To define this region in STK, a virtual site is located along the tangent line of the magnetic field flux tubes where heating occurs. The magnetic field lines where heating will occur over Arecibo makes a 45° angle with the bore sight of the heater beam at an altitude of approximately 250 km. The virtual station is then set 250 km to the magnetic north of Arecibo to make the second leg of an equatorial triangle with the heater, heated point, and virtual station. At these latitudes, a degree of latitude is 110 km. Thus, the virtual point is placed at 20.6142°N latitude north of Arecibo.

Figure 4.4 shows the geometry for defining the vertical beam width of the virtual station. The virtual station's beam is defined by the difference between the angle made to the near edge of the heater beam and the maximum heated altitude and elevation angle of the intersection of the far side of the heater beam and the minimum heated altitude. From Figure 4.4, the angle θ_1 is defined as the angle to the minimum heating altitude and θ_2 as the angle to the maximum heating altitude. The vertical beam width, θ_{HPVBW}, of the virtual point station is thus the difference between the two angles, i.e.,

$$\theta_1 = \tan^{-1}\frac{h_{\min}}{d},$$

$$\theta_2 = \tan^{-1}\frac{h_{\max}}{d},$$

$$\mathrm{VP}_{\mathrm{HPVBW}} = \theta_2 - \theta_1,$$

The horizontal extent of the heated region is the region within Arecibo's main beam extending out 50 km on either side perpendicular to the magnetic field to account for plasma transport perpendicular to the magnetic field. The horizontal extent is calculated for the middle of the heated region and is expressed as

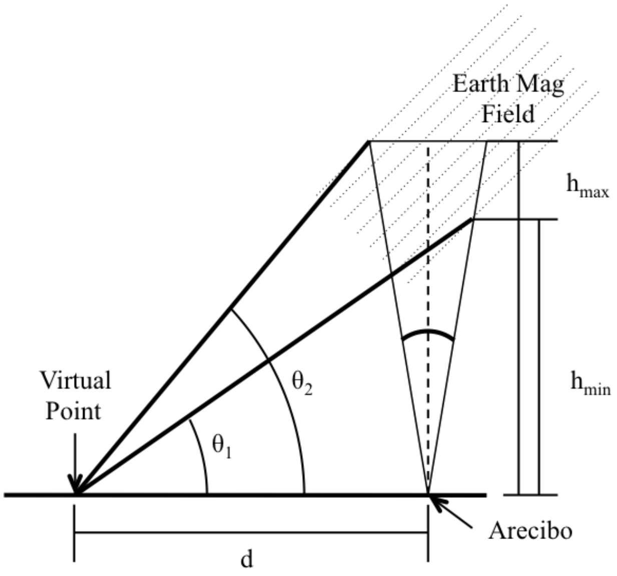


Figure 4.4: Definition of the location of the virtual point used to model the Arecibo science region.

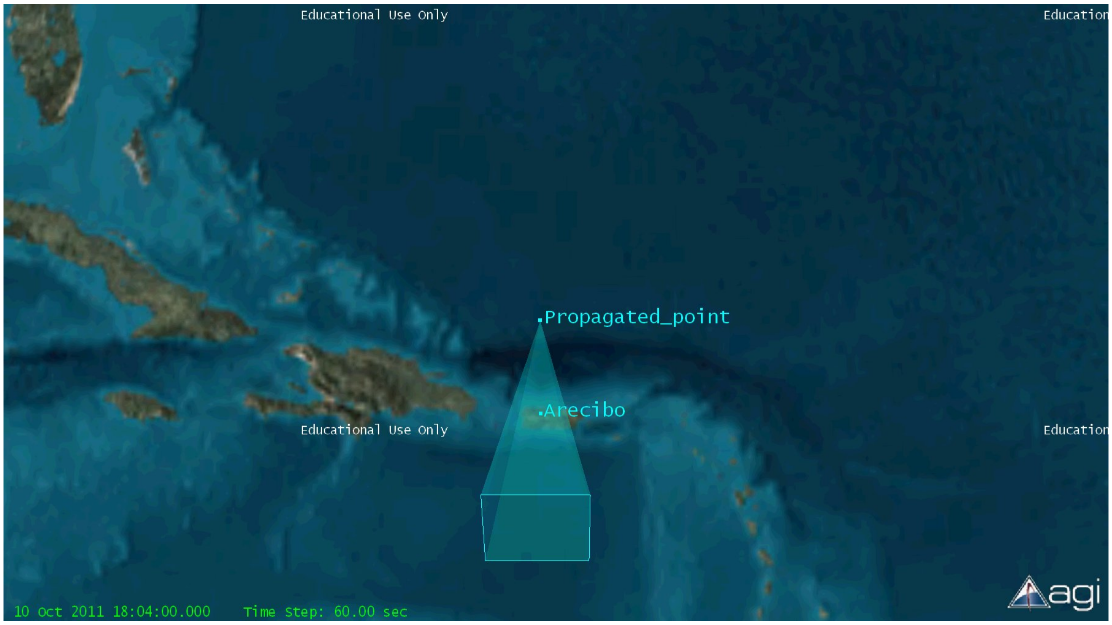


Figure 4.5: Image from STK showing the virtual point and the science region from the virtual point.

$$h_{\mathrm{avg}}\theta_{\mathrm{BW}} + 100 \text{ km} = \text{width [km]}.$$

Using the same relationship, the horizontal beam width, θ_{VPBW} is found at the virtual point to be

$$\theta_{\mathrm{HPHBW}} = \frac{\text{width}}{h_{\mathrm{avg}}\sqrt{2}}.$$

Figure 4.5 shows the resulting scenario in STK after implementation of the virtual point station.

Using STK's link access features, the times during which the satellite will be over this region can be determined and constrained by local time. The heater will operate predominately at night to isolate the effects of the heater from those of solar inputs, thus access would be constrained from the times of local midnight to 4:30 am to avoid solar input.

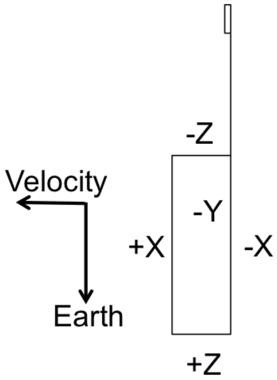


Figure 4.6: OSIRIS-3U attitude with the +Z face nadir and –X face aft.

4.3.4 Spacecraft Attitude

Within STK, the attitude can be set using the parameters within the satellite object. For OSIRIS-3U, the attitude is defined as a stable attitude with the +Z face always nadir (towards the Earth) and the –X face always facing aft or behind the spacecraft as seen in Figure 4.6. To maintain this attitude, the parameter for the spacecraft attitude is set to “Nadir alignment with ECF velocity constraint” in STK where ECF refers to the Earth Center Fixed. This keeps the spacecraft attitude oriented as required with a fixed face in the direction of travel and another fixed nadir. Further discussions of attitude control are left for another document.

4.3.5 Communication Link

While STK has the ability to model communications links, the AMSAT/IARU link budget tool was utilized to model the link budget for OSIRIS-3U to determine link losses and data rates [31]. This is an accepted and verified open source tool specifically for small satellite missions, such as OSIRIS-3U. STK was used to model the specific access time to the State College, PA ground station.

Within STK, a facility was inserted by selecting State College, Pennsylvania from the City Database. A sensor is added to this facility to simulate the ground station. The ground station has a minimum elevation angle of 15 degrees, which is applied as a constraint on the sensor just added. This allows MATLAB to poll STK for data on all access events between the satellite object and the State College ground station.

4.3.6 Satellite Lifetime

The STK lifetime tool has been shown to predict on-orbit lifetime for a satellite with reasonable success [6]. The parameters for the lifetime tool are configured from the MATLAB script command requesting the information. Specific parameters are drag area, which is set to 0.03 m^2 since the CubeSat is assumed to be traveling presenting a 3U face in the direction of travel. A 3U CubeSat mass is limited to 4-kg by the CubeSat specification rev 12; however, the 4 kg mass usage is typical for a flight system [32].

A major factor in satellite lifetime is atmospheric drag. Atmospheric drag varies during the solar cycle as the atmosphere density changes. A drag coefficient (C_d) of 2.2–2.3 is a good estimate for most conditions for a 3U CubeSat [6].

4.4 Simulation Results

4.4.1 Model Inputs

A 10,000 point *Monte Carlo* simulation as described above was executed to understand the CubeSat system specifically the impact of altitude and inclination on orbital lifetime, communications downlink time, and number of passes through the science region. Figure 4.7 shows the distribution of the model input variables for inclination and altitude. Inclination is weighted towards the major launch sites within the United States.

4.4.2 Lifetime

Lifetime simulation results, shown in Figure 4.8, illustrate an exponential increase in lifetime as the altitude increases. This is due to the exponential decrease in atmospheric density with altitude. This general trend is superimposed with the effects of the solar cycle on neutral atmosphere densities. Throughout the 11-year solar cycle, the neutral atmosphere expands and contracts changing the drag that a satellite experiences. During the solar maximum, more solar energy and particles are absorbed by the atmosphere causing the neutral atmosphere to appear to contract while solar minimums cause an expansion. The net result is that satellites

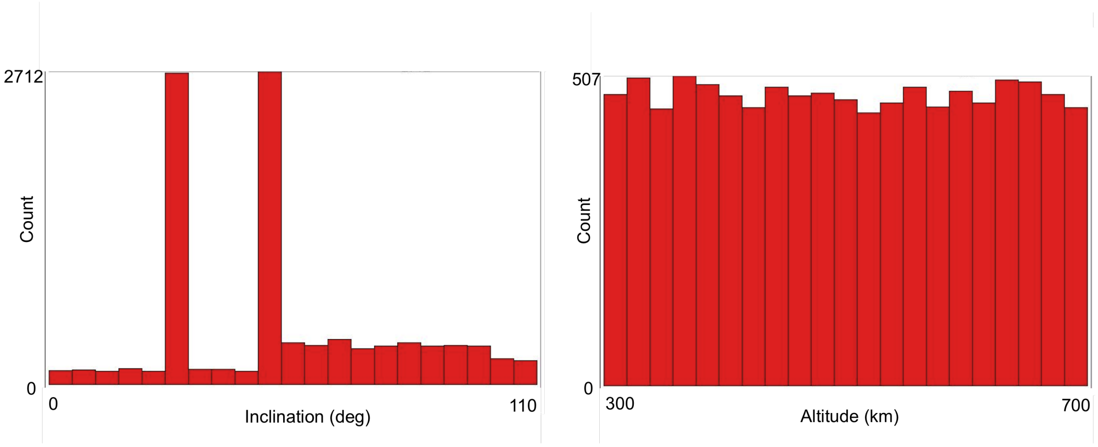


Figure 4.7: Model input variables distribution showing (a) inclination, and (b) altitude.

will experience more atmospheric drag during solar minimums resulting in a shorter lifetimes.

These variations are seen in Figure 4.8 and Figure 4.9, both lifetime plots showing total lifetimes of 100 and 25 years, respectively. Both plots show the expected exponential lifetime increase with time with variations occurring approximately every 11 years superimposed on the lifetime curve. In Figure 4.9 a peak occurring at about 3800 days (about 10 years from launch) and another around 7900 days (about 21 years from launch) showing this 11 year periodicity due to the solar cycle.

Lifetime is an important limiting parameter for the CubeSat mission design. The United Nations Inter-Agency Space Debris Coordination Committee states that a satellite must deorbit within 25 years of launch [33]. This rule was created in response to a growing concern over the impact that defunct satellites have on ongoing satellite operations. It is common for operational satellites to shift position in an effort to avoid existing debris. This is especially widely publicized when the International Space Station (ISS) needs a shift or the astronauts take shelter in escape pods during close encounters.

The deorbit constraint, therefore, imposes an upper limit to the altitude range for OSIRIS-3U. As can be seen from Figure 4.8, an altitude below 600 km meets this 25-year deorbit constraint.

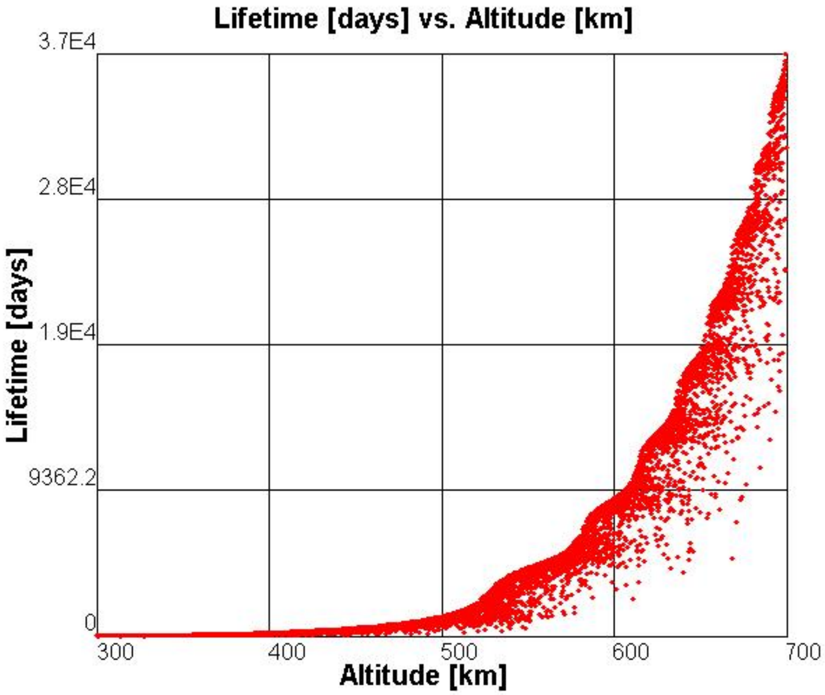


Figure 4.8: Spacecraft lifetime [days] vs. altitude showing the spacecraft lifetime increasing exponentially as the altitude increases.

A minimum mission lifetime of 6 months is required to meet the mission science goals and heater availability as discussed in Section 2.2. Figure 4.8 shows that an approximate cutoff altitude of 400 km limits many of the orbits to within the lifetime range.

After applying the 400-to-600 km altitude limits as shown in Figure 4.9, 3.4% of the remaining orbits fall within the 6-month-to-25-year lifetime restrictions.

4.4.3 Communications Time

Satellite access time to State College limits the acceptable inclinations for OSIRIS-3U due to the volume of generated data. Figure 4.10 shows the distribution of access times based on altitude, as shown in Figure 4.10a, and inclination, as shown in Figure 4.10b. Figure 4.10a shows that there is a general upward trend in available communications time with the expected increase in time that the ground station can be seen from a higher altitude.

Figure 4.10b shows that the outliers in Figure 4.10a come from low inclinations.

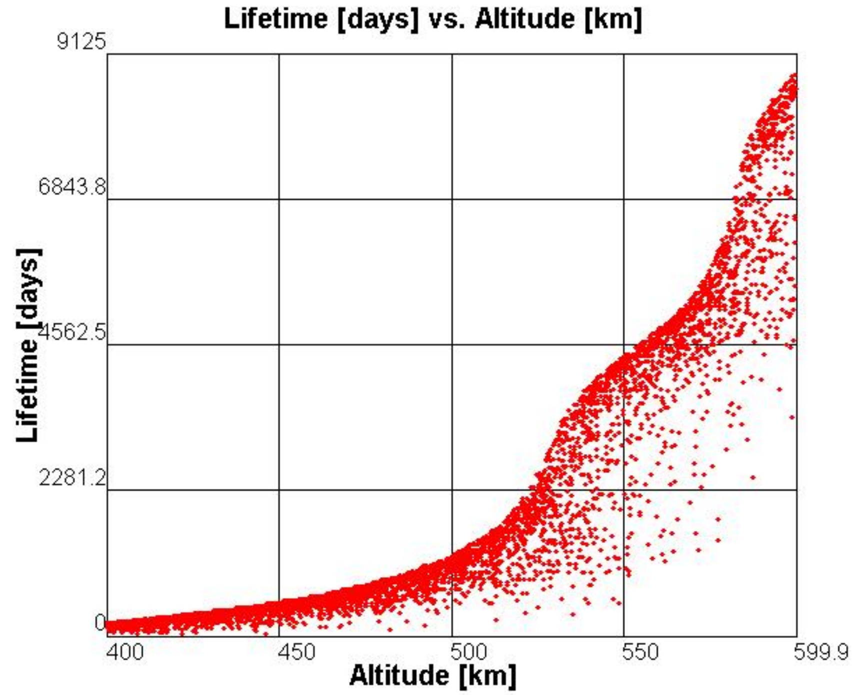


Figure 4.9: Spacecraft lifetime [days] vs altitude after altitude constraints have been applied showing that many of the remaining orbit fall within the 6 month to 25 year orbital lifetime constraints.

Once again, this is expected as a spacecraft orbiting around the equator does not have line-of-sight access to State College located around 47 degrees north latitude.

To facilitate additional science data collection and open up the range of acceptable inclinations slightly, the OSIRIS-3U science team agreed to a three-week window to downlink the data collected during the two-week science period. This allows a minimum threshold of 16800 seconds (4.67 hours) on the two-week downlink time. Thus, selecting an inclination threshold of 25 degrees inclination allows 100% of the remaining simulations to be acceptable for orbital scenarios.

4.4.4 Science Access

After the communications and lifetime limits have been applied, Figure 4.11 shows the profiles for number of science accesses to Arecibo. Figure 4.11a shows the distribution of science access counts versus altitude showing that there is an even distribution of science access counts over each altitude. Statistically analyzing the

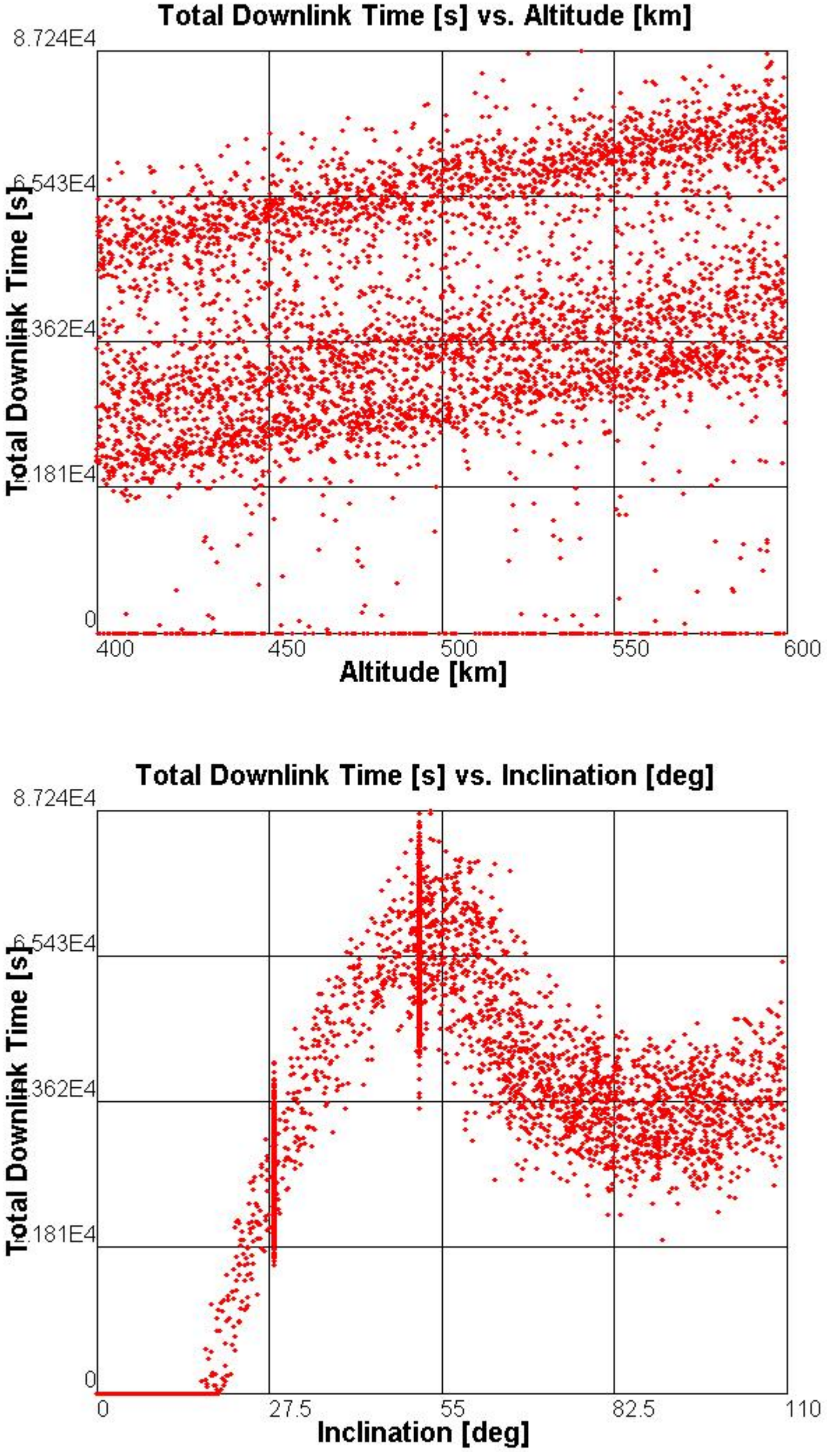


Figure 4.10: Amount of downlink time in a two-week period versus (a) altitude and (b) inclination.

data shows a mean of 5.2 accesses with a standard deviation of 3.2. This drives the requirement to consider eight science passes for a two-week period by including one standard deviation from the mean. While the science community requests as much data as possible, one standard deviation above the mean serves as a compromise point.

Figure 4.11b shows a decrease in the maximum number of science passes as inclination increases. This is due to a change in the geometry of the science pass as orbits at the higher inclinations cross more from north to south (or south to north) rather than diagonally or from west to east as would be done at lower inclinations and spend more time in the orbit at the Arecibo latitude.

Table 4.5 shows the breakdown of simulation results as they pertain to overall mission success. Of 10,000 simulated orbits, 4614, or 46%, meet the present altitude and inclination range requirements. Of these, some remain outside of the lifetime limits, thus reducing the acceptable total before analyzing the data based on science success. Defining the minimum acceptable success criteria based on completing two passes or more during a two-week science mission, there is a 90.2% likelihood that the mission can be completed as specified given any initial orbit within the limits of 25–110°inclination and 400–600 km altitude.

Table 4.5: Simulated orbits meeting science criteria levels.

	number	% remaining
Within bounds	4614	100
Meeting lifetime	4458	96.6
≥1 science pass	4223	91.5
≥2 science passes	4162	90.2
≥3 science passes	3969	86.0
≥4 science passes	3576	77.5
≥5 science passes	2831	61.4
≥6 science passes	2252	48.8
≥7 science passes	1740	37.7
≥8 science passes	1473	32.0
≥9 science passes	1255	27.2
≥10 science passes	975	21.1
≥11 science passes	618	13.4
≥12 science passes	322	7.0

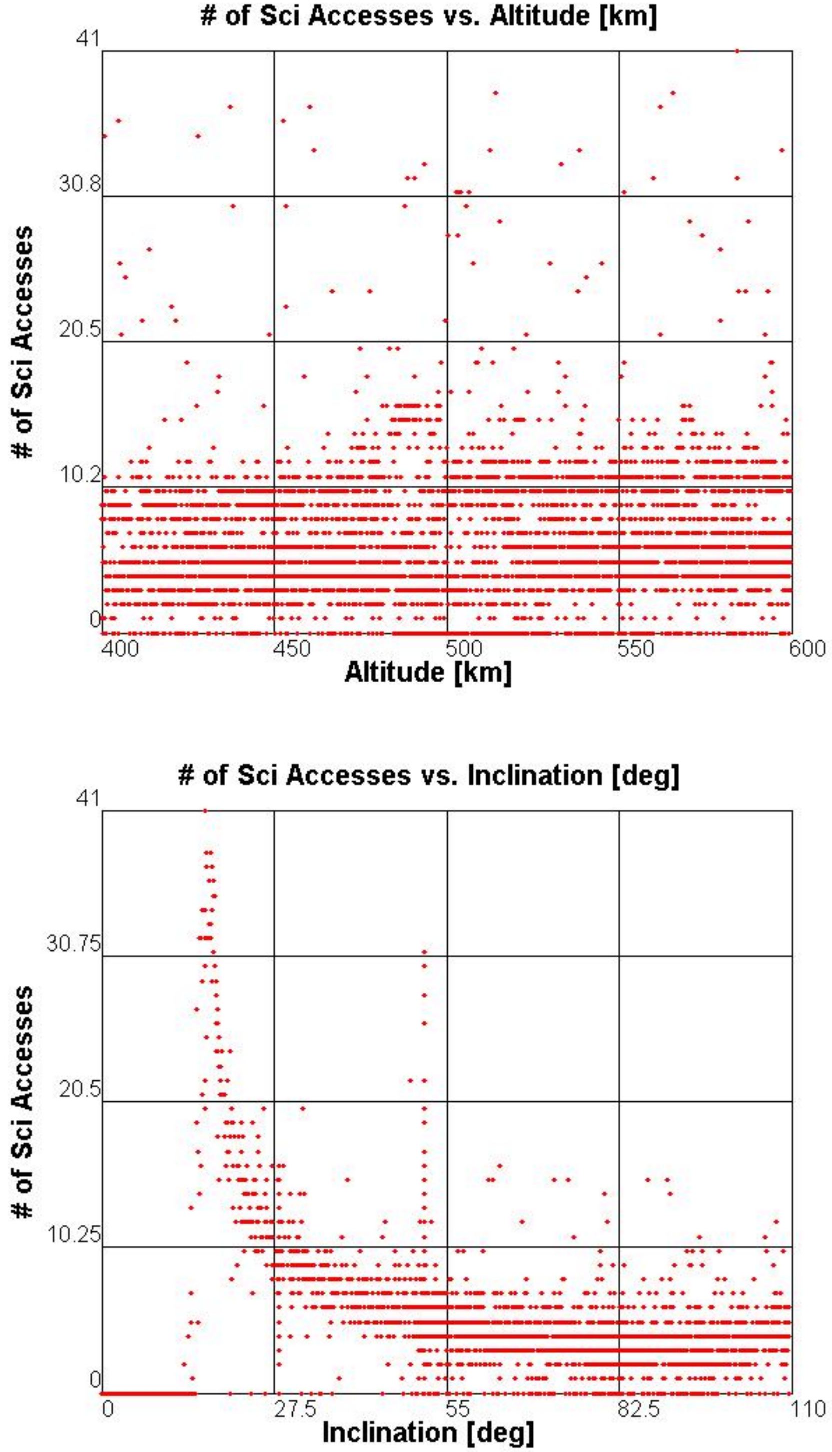


Figure 4.11: Number of science passes in a two-week period versus (a) altitude and (b) inclination.

Chapter 5

Plasma Impedance Probe Design

This work focuses on the development of a prototype of the Plasma Impedance Probe (PIP), or when implemented with a control loop, the Plasma Frequency Probe (PFP). Penn State's Langmuir probe designs have flown many times on rocket missions and already meet many of the requirements for flight on a CubeSat. Progress has been made in the development of Penn State's PIP technology and is detailed in [34][35][36]. The goal of this work was to develop a brassboard prototype to demonstrate a circuit to make the same measurement as a network analyzer while saving on power, mass, size, and cost.

5.1 Instrument Requirements

The probe is designed to support a wide range of atmospheric plasma diagnostics experiments. Basic instrument requirements are derived from the expected range of densities and magnetic field strength found on orbit. Table 5.1 summarizes the conditions considered for this design. Plugging these into Equation 2.19 yields the upper hybrid frequency for the four permutations of density and magnetic field strengths. This leads to a desired instrument frequency range between 750 kHz and 29 MHz.

An accuracy of 1% is desired across the density range. This requirement leads to 256 logarithmically spaced frequency points to acquire one full sweep and detect the resonance frequency within 1%. Other methods can be considered to reduce the number of frequency points while keeping the same accuracy with an *a priori*

Table 5.1: On-orbit environmental conditions.

Environmental Conditions		
Minimum Density	5.00×10^{9}	m^{-3}
Maximum Density	1.00×10^{13}	m^{-3}
Minimum Mag Field	0.15	G
Maximum Mag Field	0.68	G
Minimum f_{uh}	0.76	MHz
Maximum f_{uh}	28.46	MHz

estimate of the plasma frequency, or by actively tracking the plasma frequency via the implementation of a control loop.

A typical science mission requires a 10-m-or-better spatial resolution. A LEO satellite is travels at 7.5 km/s, which requires a density solution every 1.3 milliseconds. Operating in a sweeping mode, each solution requires 256 frequency points. Thus, the instrument needs to complete a source–measure cycle frequency of 200 kHz.

Additional instrument requirements are allocated from typical CubeSat accommodations including mass, volume, and power. The typical CubeSat will allocate 300 g to an instrument that is approximately the volume of a PC104 embedded computer board and about 1 W of continuous power while turned on. Table 5.2 summarizes the instrument design requirements targeting the OSIRIS-3U mission.

Table 5.2: Plasma Impedance Probe instrument requirements.

Minimum Frequency	750 kHz
Maximum Frequency	30 MHz
Measurement Accuracy	1%
Velocity	7.5 km/s
Spatial Resolution	10 m
Mass	300 g
Continuous Power	1 W
Volume	90 l × 90 w × 20 h mm
Data Rate	115200 baud

5.2 Instrument Design

Figure 5.1 shows the major components for the prototype plasma impedance probe. This first iteration interfaces with an external (off-board) microcontroller to allow a focus on the front-end electronics. The controller interfaces with a direct digital synthesizer (DDS), which is a function generator configured for operation between 750 kHz and 30 MHz. The DDS feeds into the electrometer, which provides a high impedance coupling to the probe, immersed in plasma. The electrometer also couples the DDS signal to a reference tank circuit that is representative of the probe impedance in free space. The return signal from both the probe and reference are compared for phase and amplitude variations, which are then sampled by an analog-to-digital converter (ADC) reporting back to the controller.

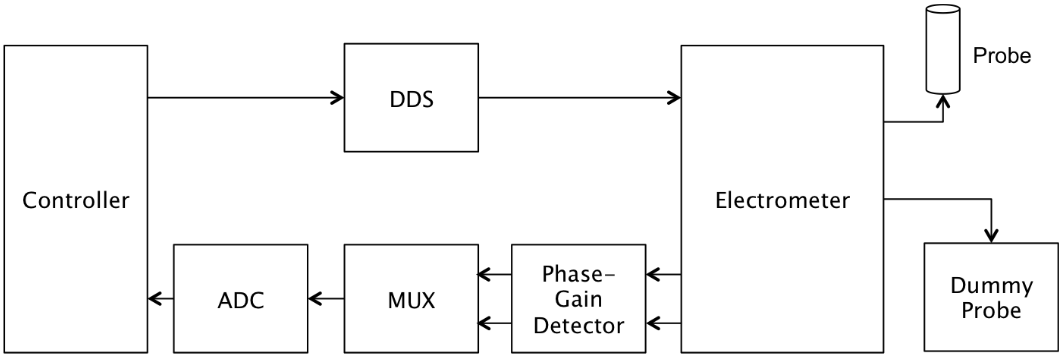
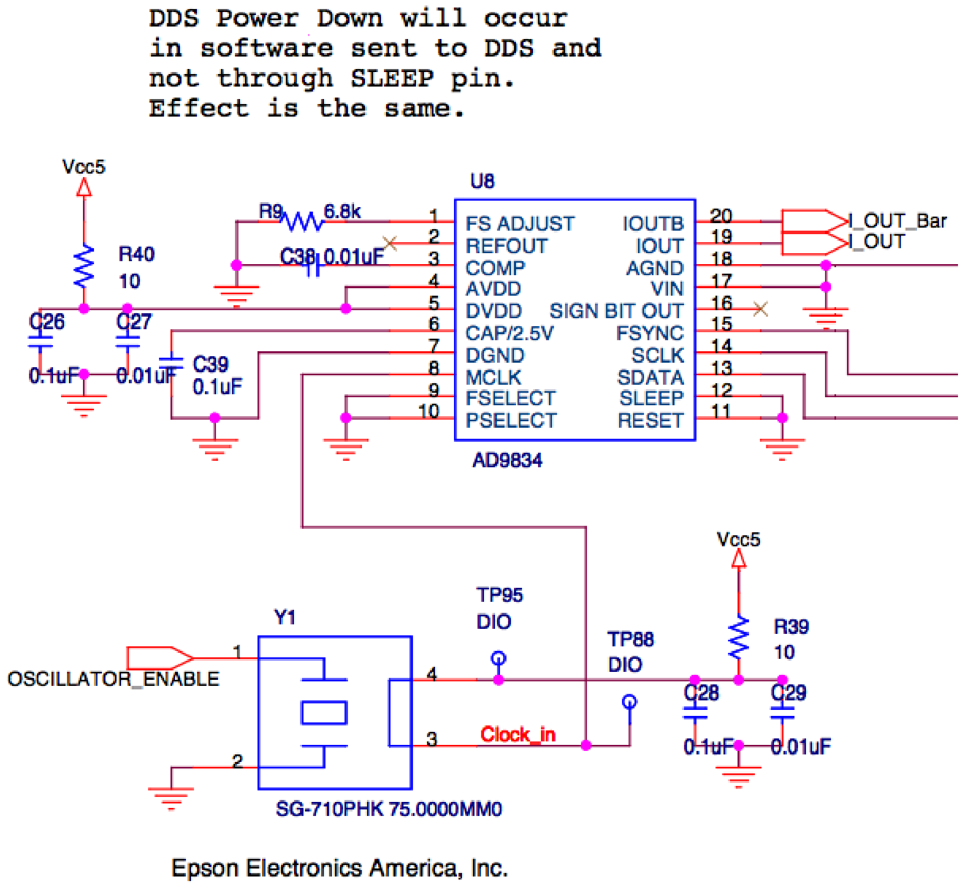


Figure 5.1: Plasma impedance probe block diagram.

The DDS selected for this design is the Analog Devices AD9834. The AD9834 can output frequencies up to 37.5 MHz when used in conjunction with a 75-MHz oscillator. This frequency range meets the requirement to operate up to 30 MHz. This DDS operates on the serial peripheral interface (SPI) bus, which provides for high speed command and control (required 4.8 Mbps minimum based on spatial requirements) of the DDS. Figure 5.2 shows a schematic of the implemented DDS circuit.

The electrometer couples the signal generated by the DDS to both the tank circuit and the probe itself. Figure 5.3 shows the schematic for the electrometer using an operational amplifier (op amp) to provide this coupling. Under the assumption that op amp errors are insignificant compared to the overall signal, the voltage on the op amp's inverting terminal can be assumed to be driven equal to



DDS Power Down will occur
in software sent to DDS and
not through SLEEP pin.
Effect is the same.

Figure 5.2: DDS schematic.

that on the positive terminal through the negative feedback loop. This allows the applied DDS signal to be driven to the inverting terminal and thus coupled to the tank circuit and probe. The reflected current is then seen back through the feedback loop and onto the phase-gain detector.

The tank circuit is composed of a capacitor and resistor that are set to match the impedance of the probe immersed in a plasma. The op amp is configured as a buffer, although the option is built into the circuit to add gain in the feedback.

The output of the electrometer is connected to the phase-gain detector, which is implemented using the Analog Devices AD8302. The AD8302 covers a signal range up to 2.7 GHz with a measurement of ±30 dB magnitude and ±90° phase difference, which is more than sufficient for this application. Figure 5.4 shows the schematic for the phase-gain detector. Using the phase-gain detector, the measured

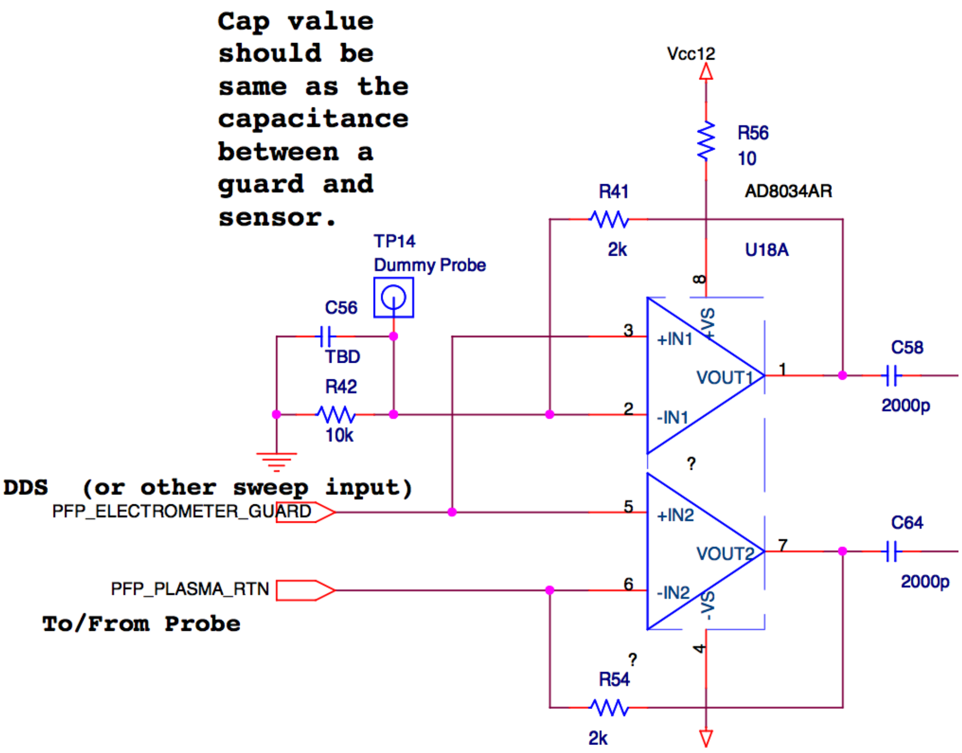


Figure 5.3: Electrometer schematic.

impedance is

$$Z = Ae^{j\theta},$$

where A is measured magnitude in volts and θ is the phase difference in radians.

Finally, the signal is measured by the ADC, in this case the Analog Devices AD1674. This part was included as part of the legacy design due to the availability of a radiation-tolerant version of the same chip. Since much of the rest of the circuit is not available in radiation-tolerant versions, future designs could include a different converter. The AD1674 is a 100-kHz dual-supply ADC; hence, it should be noted that this can only achieve a 20-m resolution when operated as the plasma impedance probe rather than having the additional software control to implement the plasma frequency probe frequency-tracking capability. When this part was initially selected, the focus of the development had been on the PFP and including the feedback loop. Figure 5.5 shows the ADC as implemented.

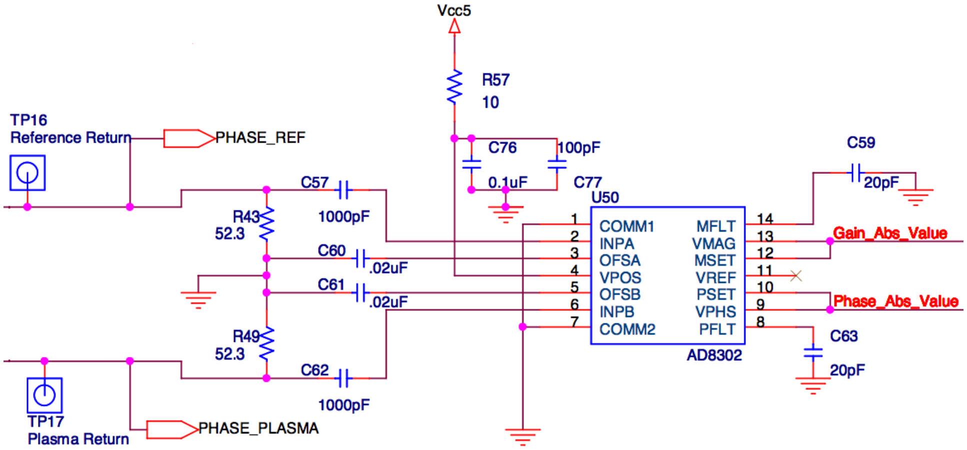


Figure 5.4: Phase–gain detector schematic.

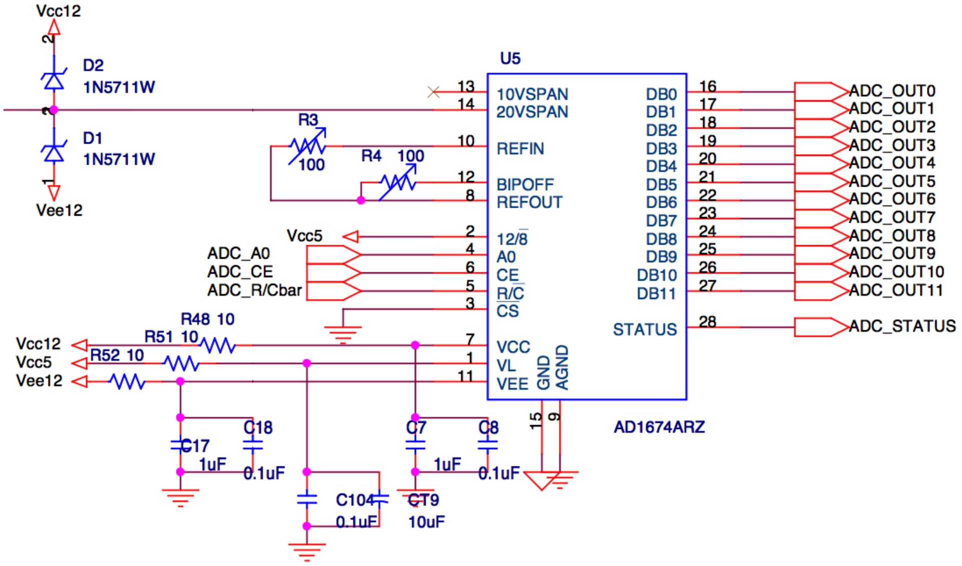


Figure 5.5: ADC schematic.

5.3 Test Circuit

The plasma can be modeled as an inductor, capacitor, and resister connected in parallel. By changing the inductor and capacitor, the resonance point can be changed to simulate differing plasma parameters.

Figure 5.6 shows the tank circuit board used during testing and developed as part of the work of Siegel [35]. There are 20 resonant circuits implemented on this board, of which 8 produce strong resonance structures that can be used in testing. Each circuit can be connected using a jumper that connects the circuit (along the outside of the board) to the main trace going to the instrument under test. Table 5.3 shows the realized circuit parameters for each of the available jumper positions.

These frequency parameters were verified by connecting the tank circuit to a network analyzer. Figure 5.7 shows the results of characterizing the tank circuit on the network analyzer. A feature to note is the phase lag seen above 10 MHz on the phase plot. This is a feature of interest when analyzing the data taken from the PIP in test.

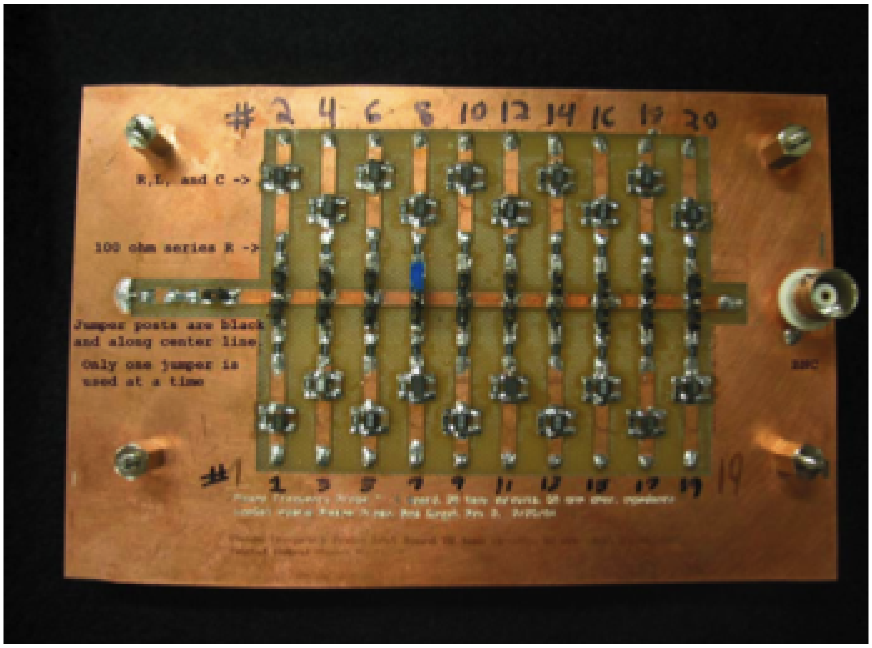


Figure 5.6: Test board with various tank circuits to simulate differing plasma conditions [35].

Table 5.3: Table summarizing the resonance frequencies of the tank circuit board jumper settings. Unlisted jumpers settings do not produce sharp resonance features [35].

Jumper Position	Resonance Frequency (MHz)
1	0.3739
2	1.4786
5	1.5443
12	4.4967
13	5.2793
15	6.2435
10	7.1750
14	10.795

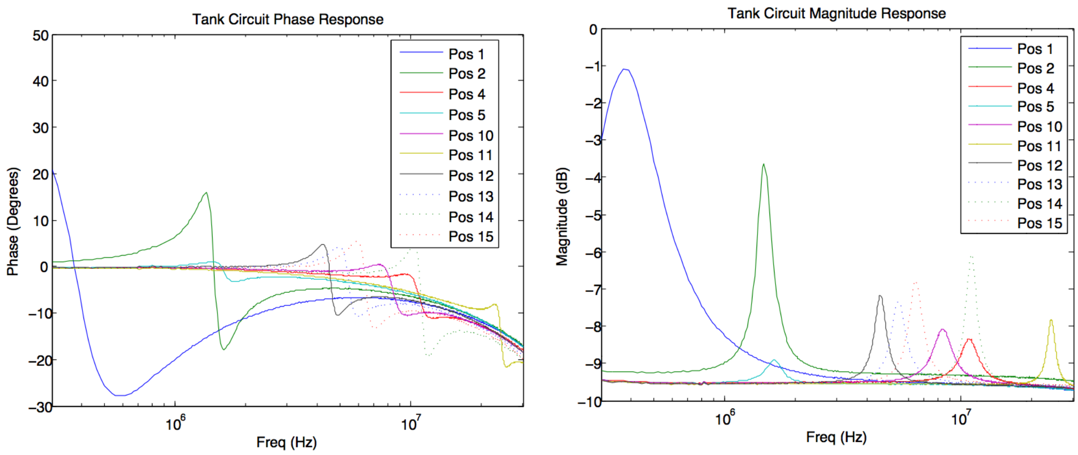


Figure 5.7: Tank circuit responses for each jumper position as measured by a network analyzer with internal calibration turned on: (a) phase response and (b) corresponding magnitude response.

5.4 Calibration

Modern network analyzers make use of the open–short–load (OSL) calibration method for one-port measurements. As the name implies, this method involves connecting three different terminators to the analyzer: an open circuit, a short circuit, and a load matched to the line impedance, which is typically 50 ohms.

Error in a network analyzer (note, as previously discussed, the PIP functions similar to a network analyzer) can be treated as a two-terminal device in series between the instrument port and the device under test (DUT). This error includes phase lag and signal attenuation due to the circuit parasitic capacitance, inductance, and resistance that is inherent in solder joints, circuit boards, and connectors. An OSL calibration allows the effects of these parasitics to be understood and thus calibrated out.

For low frequencies, the open, short, and load terminators can be assumed to be idea devices. With this assumption, the characterized errors can be removed using

$$Z_{\text{calibrated}} = \frac{Z_{\text{measured}} - b}{a - cZ_{\text{measured}}}, \tag{5.1}$$

where Z_{measured} is the measured impedance of the DUT, $Z_{\text{calibrated}}$ is the true impedance of the DUT with the errors removed, and a, b, and c are error terms that are determined from the OSL characterization. They are defined as follows

$$\begin{aligned} a &= \frac{Z_{\text{load}}(Z_{\text{open}} + Z_{\text{short}}) - 2Z_{\text{open}}Z_{\text{short}})}{Z_{\text{open}} - Z_{\text{short}}}, \\ b &= Z_{\text{load}}, \\ c &= \frac{2Z_{\text{load}} - Z_{\text{short}} - Z_{\text{open}}}{Z_{\text{open}} - Z_{\text{short}}}, \end{aligned}$$

where Z_{load} is the measured impedance with the load terminator connected, Z_{open} is the measured impedance with the open terminator connected, and Z_{short} is that with the short terminator connected.

Figure 5.8 shows the example of this calibration applied to data from the network analyzer with the automatic calibration turned off. First, the parameters for the OSL terminators were measured and then the tank circuit was connected with arbitrary jumper position two selected. As with the original calibration data, the phase can be seen dropping off as the frequency increases. This is likely due to stray capacitance and other manufacturing imperfections in the tank circuit board.

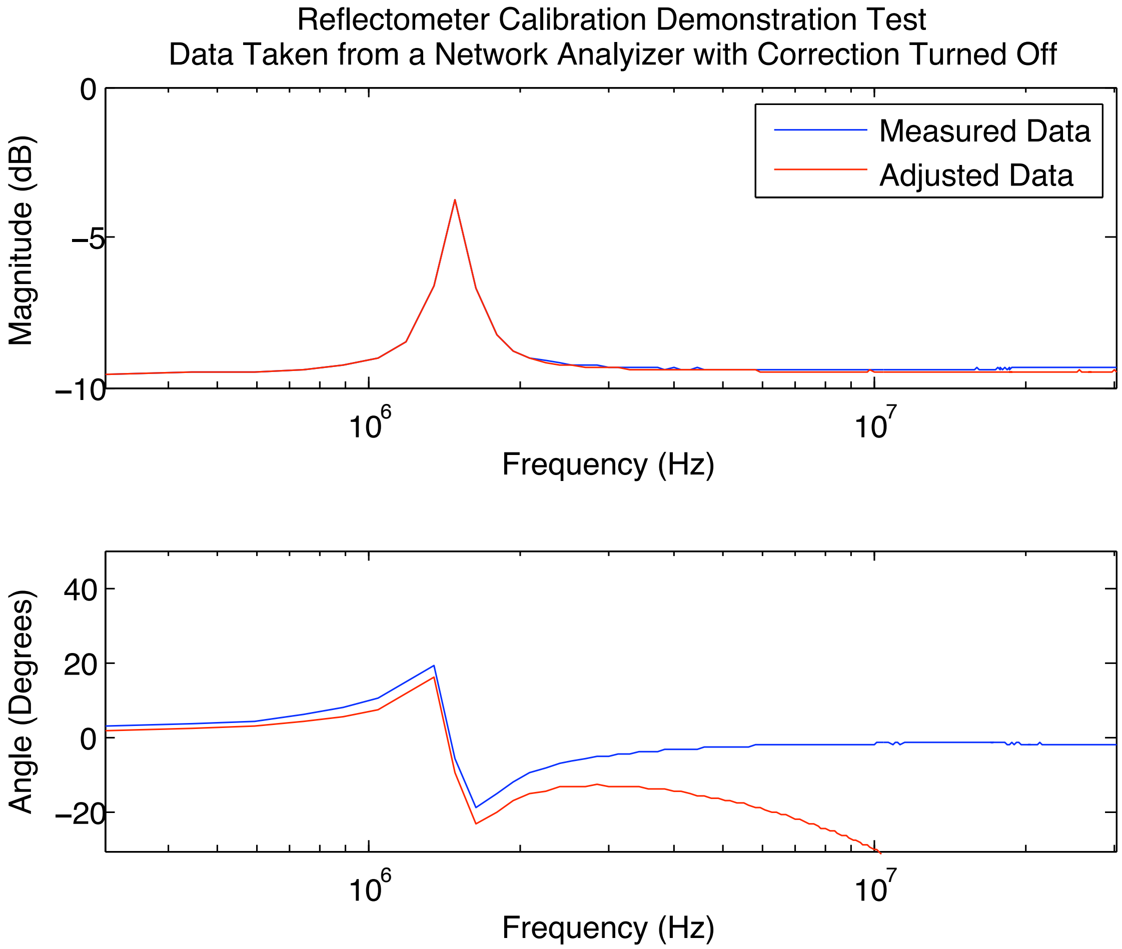


Figure 5.8: Example showing the effect of OSL calibration on data taken of the tank circuit connected to the network analyzer.

5.5 System Testing

Testing was broken up into two parts, component-level and then subsystem-level verification. Each major component was verified and characterized prior to testing the entire system with a tank circuit.

5.5.1 Component-Level Testing

5.5.1.1 Analog-to-Digital Converter Verification

The ADC was verified using the Omega CL8300 precision voltage source. The CL8300 was connected to the ADC and commanded to ten points across the range

−10 to 10 V. At each voltage step, the CL8300 output was confirmed by a handheld Agilent digital multimeter that was recently calibrated and agreement was consistently found down to the precision of the Agilent multimeter. At each step the digital code was read from the ADC and plotted as shown in Figure 5.9. The equation of a regression line through the ADC points is taken as calibration for all future ADC measurements.

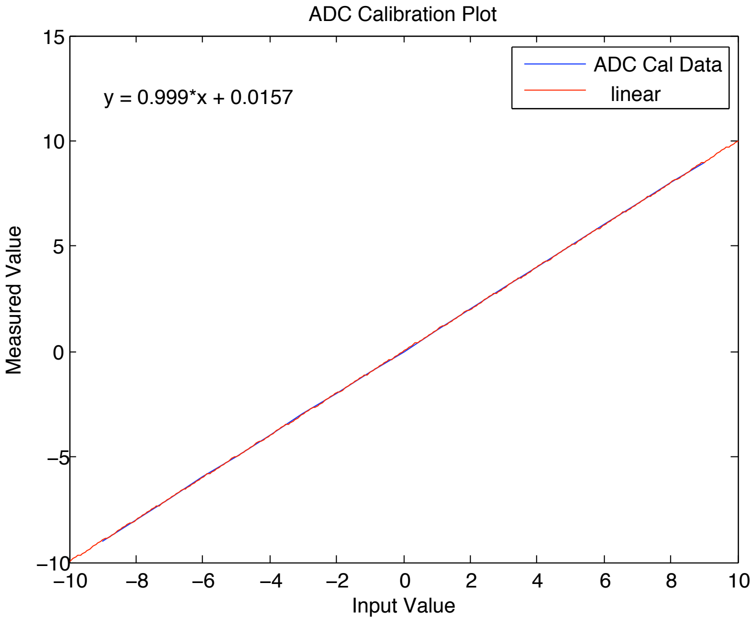


Figure 5.9: ADC calibration response.

5.5.1.2 Direct Digital Synthesizer Verification

The DDS was verified by commanding it to six different frequencies and measuring the spectrum at those frequencies. The DDS functions by down converting from the 75-MHz oscillator to the desired frequencies. This leaves the opportunity for spur frequencies between the desired frequency and 75 MHz to be generated. Figure 5.10 shows the output spectra for the six frequencies with the corresponding frequency spurs. Each frequency commanded has a corresponding image reflected about 37.5 MHz, the upper limit of the DDS. For example, the 30-MHz signal has an accompanying frequency spur at 45 MHz $(37.5 - 30 + 37.5)$.

These spur frequencies are filtered to minimize the potential contributions they will make to the overall measurement as shown in Figure 5.10b. Future designs should include a sharper filter to minimize the image frequency components above 37.5 MHz and testing should be preformed to understand the error contribution as a result of the spurious frequency component.

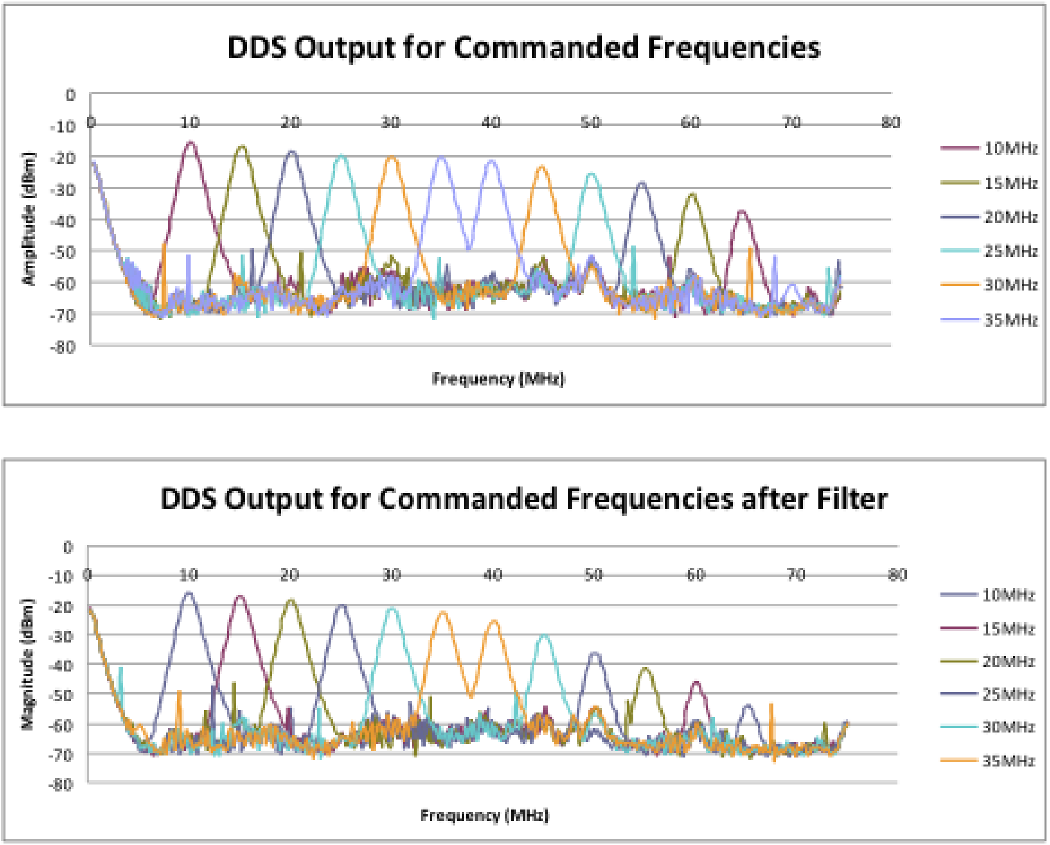


Figure 5.10: The frequency spectrum from the DDS for each of the frequencies commanded across the DDS's frequency range: (a) shows the DDS output before filtering and (b) shows the DDS after filtering to minimize spur frequencies.

5.5.1.3 Phase–Gain Detector Verification

To verify the phase–gain detector, a pair of Agilent 33220A function generators with a 10-MHz sync option were phase synchronized and connected to the circuit. To monitor the generated signal, the output also was split to an Agilent 54622D oscilloscope. The function generators were set to a frequency and then the phase of one of the devices was swept from –180 degrees to 180 degrees with respect to the other generator. This process was repeated for a number of frequencies to verify the performance for different frequencies. Note: care must be taken to ensure that cable lengths are matched for each length of cable used.

Every frequency step required manual offset adjustment to compensate for differences in the two function generators. An automated test was developed to step the frequency and measure the phase difference between the two devices with the oscilloscope. A computer then shifted the phase offset in one of the function generators to ensure that the phases were synchronized before executing the test.

Figure 5.11 shows the resulting plot of the output voltage of the DDS in response to the various phase inputs. The graph shows clipping occurring near the phase extremes, which is likely a result of the gain settings on the phase–gain detector. The gain should be adjusted in future testing.

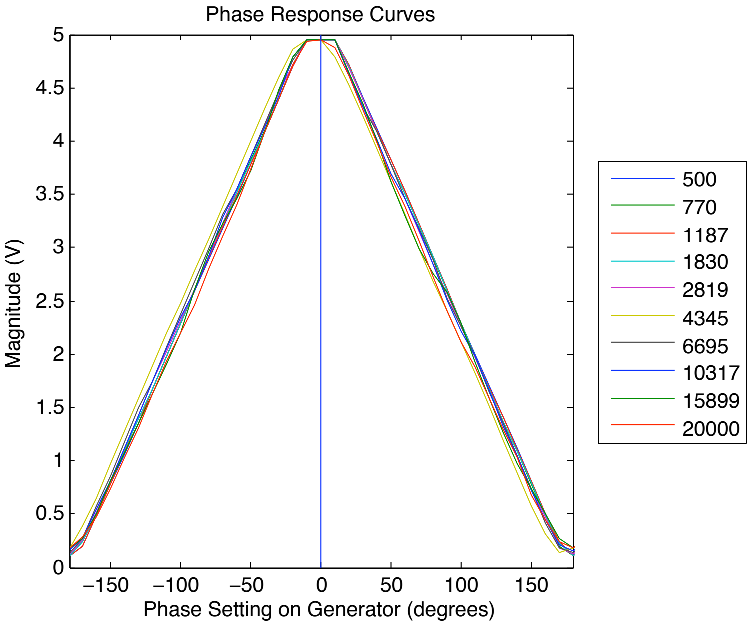


Figure 5.11: Overlaid plots of the phase–gain detector output voltage response to stimuli at different frequencies across the instruments 300 kHz to 30 MHz operating range.

5.5.2 Subsystem Level Testing

The instrument was interfaced with the tank circuit to verify the overall operation. Figures 5.12 and 5.13 show the response of the instrument when connected to the tank circuit with jumpers in positions one and two, respectively. In both plots, the measured data (blue) are compared to the network analyzer data (red). Measurements at both jumper positions show a well defined frequency resonance near the expected frequency. Averaging the phase and magnitude resonance frequency values shows a measurement error of 2.14% for jumper position one and 1.97% for jumper position two.

Both plots show a dramatic increase in phase above 1 MHz and an increase in magnitude response above 2 MHz. These effects are seen in all the jumper settings used showing that it is a systematic error across tank-circuit resonance settings. The increasing amplitude and phase delay causes errors in plasma frequency measurement above 2 MHz making the circuit ineffective for measuring plasma frequencies above 2 MHz. An Agilent bench top frequency generator was used in place of the DDS to eliminate the DDS extraneous output frequency content as the error source. The frequency generator was verified with a spectrum analyzer to generate a narrow output frequency at the commanded frequency.

The PIP phase and amplitude measurement error increases above 1 MHz and 2 MHz, respectively, are due to probe error sources yet to be characterized and

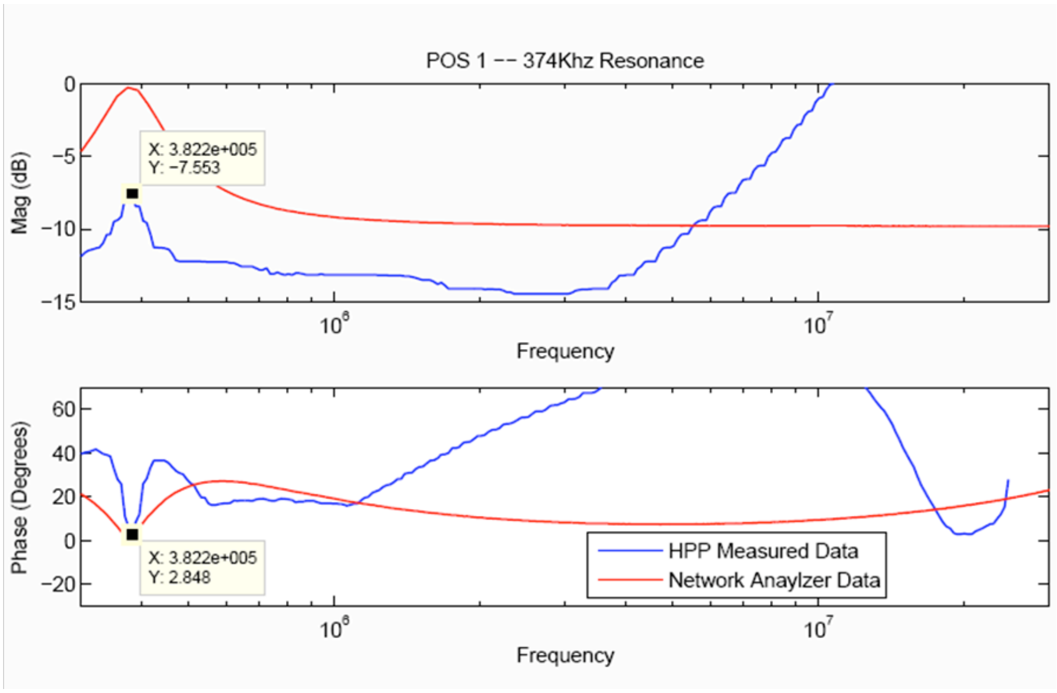


Figure 5.12: PIP measurement of tank circuit position 1.

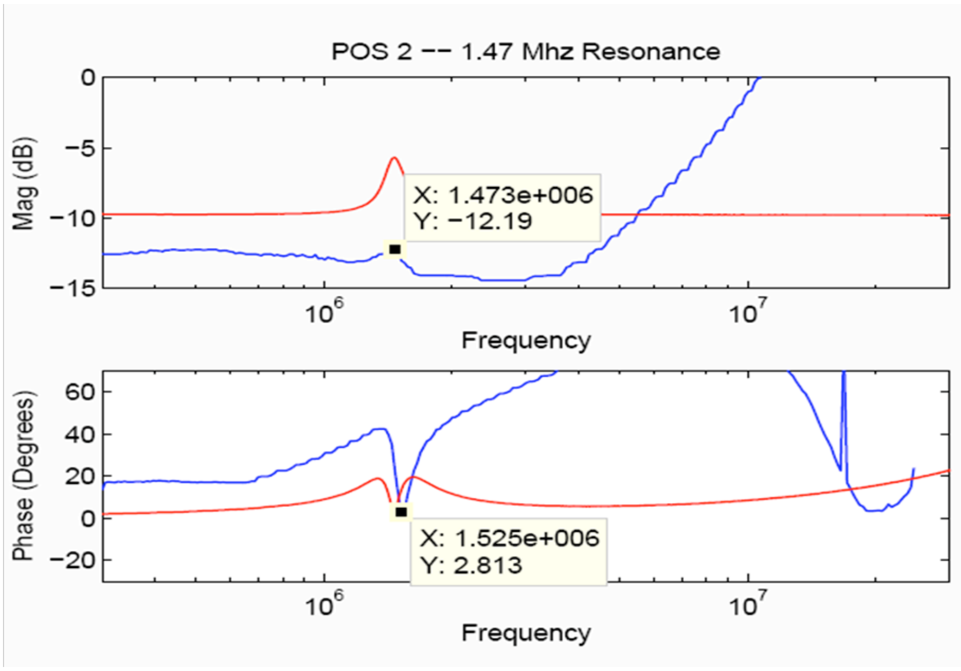


Figure 5.13: PIP measurement of tank circuit position 2.

understood. Some areas to investigate further include:

- investigate errors caused by mismatch between the “free space” impedance of the tank circuit and the reference probe circuit;
- revisit the assumption that the calibration terminators are ideal devices by incorporating the actual terminator impedance into the calibration equations; and
- simulate the test tank circuit and the effect of output frequency content on the measured value including frequency spurs and wide bandwidth frequency generation.

5.6 Frequency Tracking

Frequency tracking can be implemented to increase the temporal output of the plasma impedance probe. By actively tracking the resonant frequency, the total data volume can be greatly reduced while increasing measurement spatial resolution by decreasing the time required for each measurement. In addition, the overall measurement accuracy can be increased beyond the 1% accuracy specification for the plasma impedance probe. Control theory can be used to design an optimal frequency tracking system. A basic control system would need to first find the upper hybrid frequency and then track the frequency as the plasma parameters change. Such a system is described by Carlson [2004] where a plasma frequency probe was built and tested on a sounding rocket mission [37].

A simple system would perform a full frequency sweep from 300 kHz to 30 MHz and find the resonance frequency. The resonance frequency is where a peak in the magnitude response and zero phase crossing occur simultaneously. This point can then be tracked using a perturb-and-observe algorithm, in which a slight change in frequency is made and the effect on the phase measurement is observed. If the absolute value of the phase decreases with the new frequency, then the system keeps moving the frequency set point until the local minima is found. This perturb-and-observe algorithm can be implemented with either fixed or adaptive frequency step sizes depending on the acceptable frequency error. An adaptive control might use a gain proportional to the phase measurement determining how far the set frequency is from the phase zero crossing point.

Chapter 6

OSIRIS-3U System Development

The OSIRIS-3U program adopted a spiral model to develop the CubeSat. Lessons learned from previous SSPL projects, i.e., LionSat, NittanySat, SPIRIT, etc., showed that to ensure success of a multi-year spacecraft development effort, the projects need to be broken down into phases. Correlating these phases so that a full mission cycle can be completed over the course of an academic year provides the best results for both breaking the project into manageable elements and providing motivational learning environment for students.

The OSIRIS-3U program adopted a spiral development model with a build-a-little, test-a-little philosophy, thus breaking the CubeSat into two major development missions. The development missions were two high altitude balloon flights as part of NASA's High Altitude Student Platform (HASP). During each of these missions, the systems engineering "V" model was followed decomposing requirements from the OSIRIS-3U system into realizable subsystems and components and planning for verification.

The first mission focused on technology development and was called OSIRIS Lite (OLite). During this mission, each of the subsystems, command and data handling, guidance and navigation, power regulation and generation, and communications allocated a 5-inch-by-6-inch circuit board to demonstrate their systems. This allowed the students to begin developing the subsystems on a platform large enough to easily facilitate circuit-level testing, probing, and debug. The goal for this mission was to demonstrate the functionality of each of the subsystem components prior to requiring that they fit in a CubeSat form factor.

Following the OLite payload was the OSIRIS Lite 2 (OLite 2) payload, which implemented the same subsystems as the OLite payload, but added the requirement that they fit within the 1U CubeSat form factor. SSPL has been focusing on development of a 1U form factor spacecraft bus that allows significant room for hosted payloads and allows the flexibility to move to different CubeSat form factors per mission requirements. This section describes the system verification of the OLite 2 payload, integration with the HASP flight system, and the results of environmental tests.

6.1 OLite 2 Mission Level Requirements

The OLite 2 mission0level requirements are derived from the overall OSIRIS-3U requirements. This set of requirements incrementally builds the OSIRIS-3U system by demonstrating functionality with a subset of the CubeSat requirements applied to the system before applying the full set. These requirements are listed below.

1. The mission shall educate the student team developing the spacecraft bus in mission development, design, build, and verification to prepare for a CubeSat flight mission.

2. The communications subsystem shall data downlink over the main radio, beacon downlink, and data uplink in near satellite flight like conditions.

3. The guidance and navigation subsystem shall demonstrate the OSIRIS satellite GPS operation, attitude sensors, and attitude determination algorithm in a relevant near-space environment.

4. The power subsystem shall demonstrate power generation, storage, and supply for the satellite in near-space like conditions.

5. The thermal subsystem shall demonstrate modeling and operation of both passive and active heater control algorithms by maintaining temperature in the rarified gas high altitude balloon environment.

6. The command and data handling subsystem shall demonstrate command handling through flight software and control of each subsystem throughout

the flight.

6.2 OLite 2 System Composition

While HASP provides a good analog for satellite flight for such things as demonstrating operational control, attitude determination, and solar power generation, the thermal environment and system ConOps are vastly different. Two of the most noticeable differences are the necessity to include convection in thermal design for the balloon and the differences in sun/eclipse profiles since the balloon does not experience the same 90-minute cycles that the satellite will experience. To be able to keep the CubeSat design from being impacted by these differences and the differences in power and communications provided by HASP, the system was broken into two components, the CubeSat Simulator and the Support Box.

Figure 6.1 shows the completed OLite 2 balloon payload as built. The support box (SB) is the bottom portion of the system and is composed of power control and data interface handling. The CubeSat Simulator (CSS) is the top portion of the payload and is designed to meet the size constraints of a CubeSat. The CSS includes a power generation system, attitude determination capabilities, thermal control circuitry, a flight computer, and 430-MHz radio that make up the CubeSat spacecraft bus. An auxiliary payload, a cell phone, was added by the Penn State Lunar Lion team to demonstrate commercial technologies that could be used on a lunar landing system in a relevant environment. The phone is shown in the foreground, below the CSS bottom face, which is the command and data handling (CDH) board.

6.3 System Testing

6.3.1 Penn State Environmental Chamber Testing

The first system-level environmental test was in the Penn State Environmental Test Chamber. Figure 6.2 shows the OLite 2 system in the chamber, which is capable of achieving a vacuum of 1×10^{-6} Torr and a temperature range between –25 and 80 ° through the experiment mounting plate. The chamber is targeted to

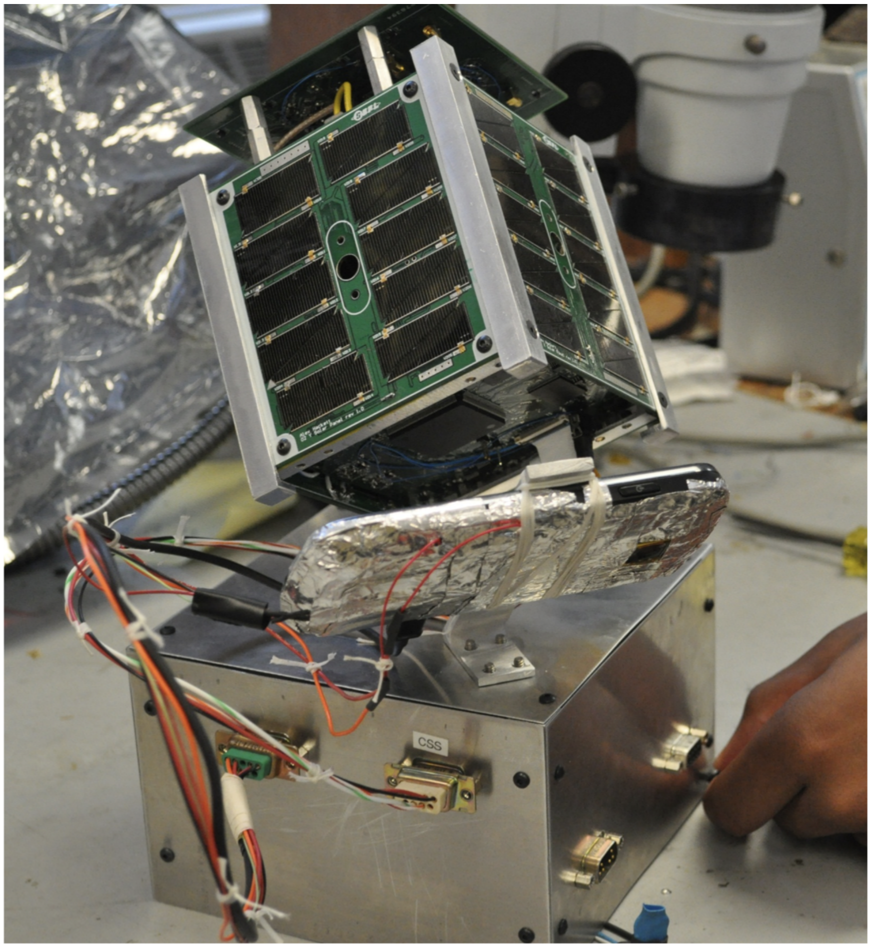


Figure 6.1: OSIRIS Lite 2 balloon payload.

simulate the conditions normally found in low earth orbit.

Thermal environmental analysis and testing for OLite 2 focused on testing as the payload would see at the maximum altitude (approximately 80 km) through radiative heat input from the Sun and Earth. As shown in Figure 6.3, the thermal team suggested a test profile bringing the chamber down to an operational pressure of 1 Torr (from 760 Torr at atmospheric pressure) and then cycling the chamber, between 50, –20, and 30 °C. This was targeted at simulating the temperatures expected within the internals of the CubeSat during orbit.

Testing in the SSPL environmental test chamber showed that many of the systems remained 10 °C above the ambient temperature due to internal self heating and limited heater usage. From this testing, the heaters were not expected to be required during balloon float.

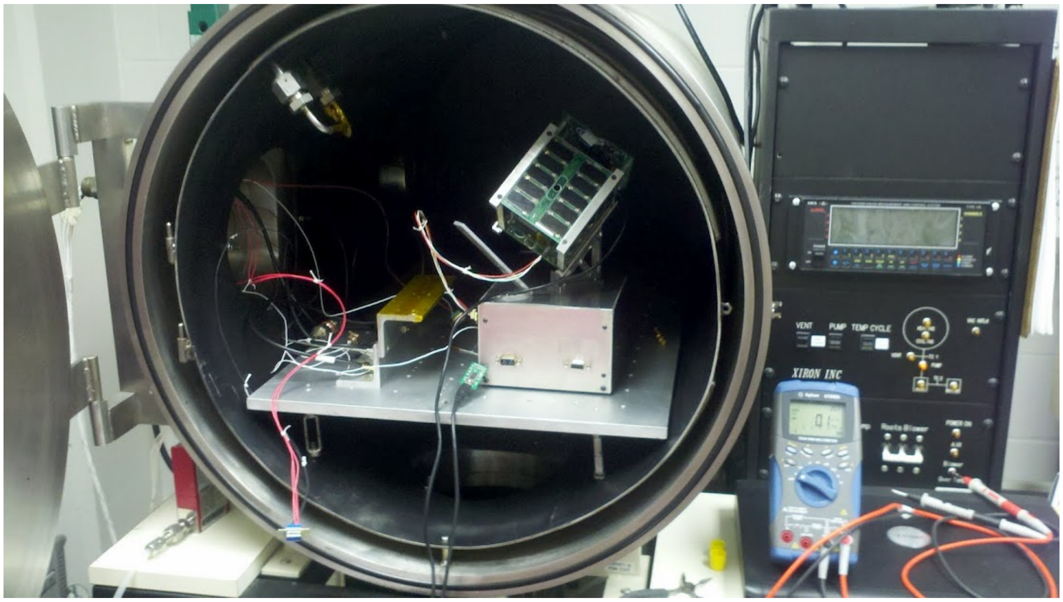


Figure 6.2: OLite 2 in the SSPL environmental test chamber.

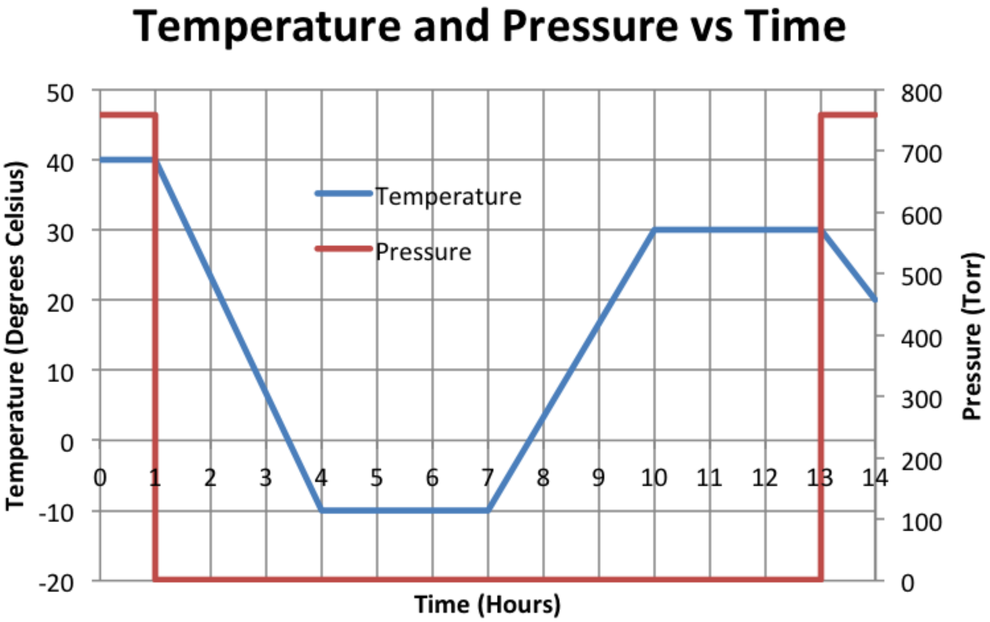


Figure 6.3: OLite 2 environmental test profile.

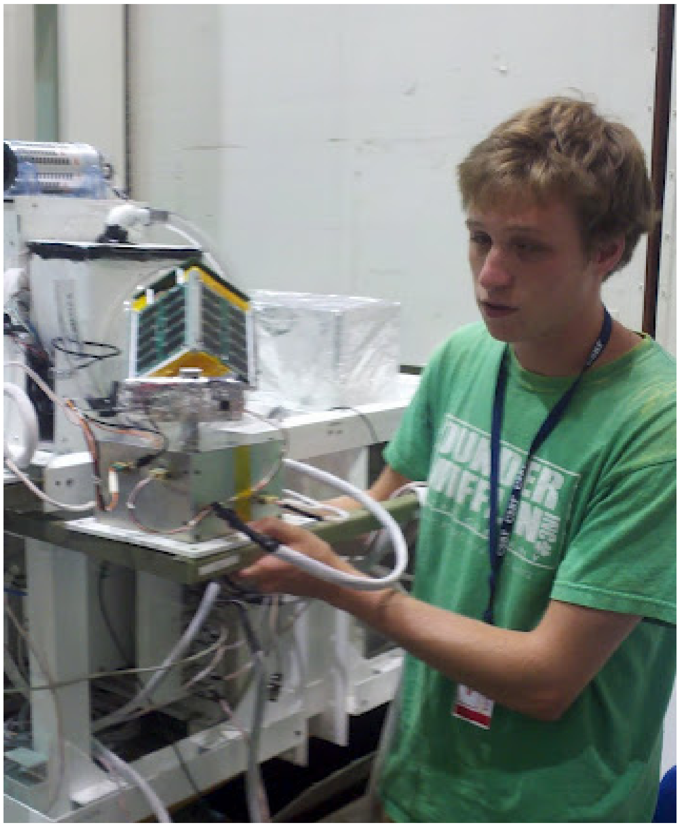


Figure 6.4: OLite 2 integrated on the HASP flight system

6.3.2 HASP Integration Testing

Integration on the HASP balloon involves verifying the mechanical properties by weighing the payload, mechanically mounting the OLite 2 payload to the HASP platform, and connecting HASP power and data to the payload. Once mounted, HASP powers on the system and verifies the system power consumption. Command sending and receiving is verified by sending a command and verifying data content change in the downlink stream as commanded.

Figure 6.4 depicts the OLite 2 payload after successfully integrating with the HASP system.

6.3.2.1 HASP Environmental Testing

Balloon testing uses a thermal test profile that differs from the testing done in the Penn State environmental test chamber. Whereas the Penn State test profile

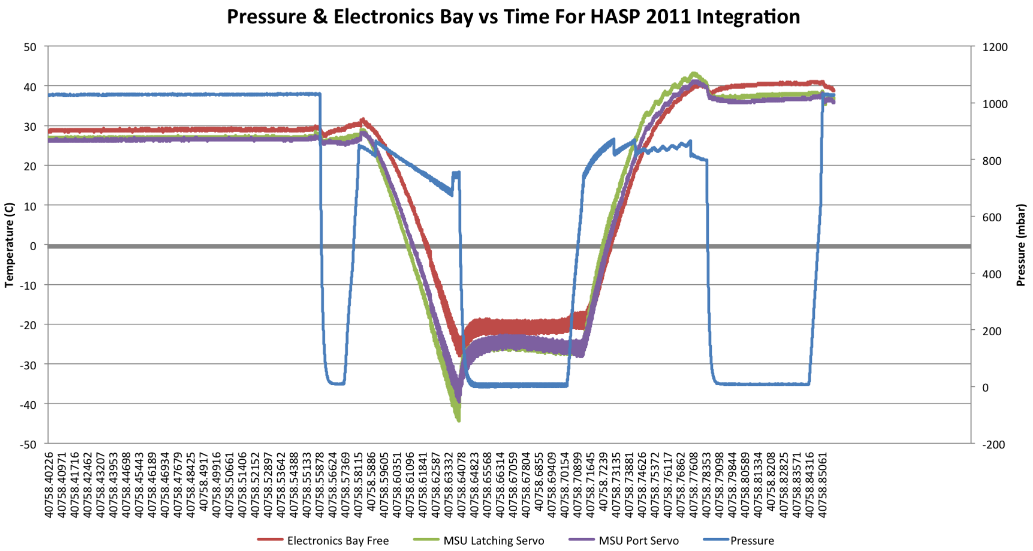


Figure 6.5: HASP environmental test profile.

involves pumping down the chamber to the pressure expected at altitude, and then cycling temperature through a radiative plate, the HASP test profile involves pulling a vacuum and then backfilling with nitrogen to use convection to control the environmental conditions. Figure 6.5 shows the temperature and pressure profile taken by the HASP system during the test. The temperature measurements were taken within the electronics bay and other electronics in an enclosed box, restricting air flow and dissipating an unspecified amount of power. Due to this, it is believed that the ambient temperature was actually much lower than that specified in this plot.

Figure 6.6 shows the temperature profile as measured by the OLite 2 system at well understood temperature sensors. The first temperature sensor, data channel BATTERY_A_TEMP, is in the battery monitoring integrated circuit and has proven in subsystem testing to track the battery temperature very well. The second sensor is the ADC_INTERNAL_HOUSEKEEPING_TEMP located on the CDH board and was shown to track the temperature of both the CDH processor and memory.

These plots show that the battery monitor tracked the temperature through –40 °C before failing, while the CDH system tracked the temperature through –50 °C before the CDH system failed and was unable to recover. The overall difference in minimum temperatures is due to additional heaters located on the battery to attempt to maintain the battery temperature, where the batteries were shown through test to fail below –20 °C.

6.3.2.2 Results

When OLite 2 was removed from the HASP environmental chamber, an attempt was made to power the system. This was met with the CDH system failing to boot from the NAND memory. The system NAND memory was a Micron Technologies MT29F4G08ABADAWP:D TR, which comes in a commercial and industrial package. Commercial rating specifies a tested range of 0 to 70 °C, while industrial has a range of –40 to 85 °C. The memory purchased for the OLite 2 system was inadvertently the commercial variety, thus resulting in corrupted data somewhere below 0 °C impeding the system from rebooting.

Back at Penn State, this chip was replaced with the industrial variety and tested to –45 °C in the lab thermal chamber (capable of –75 to 200 °C). Figure 2.1 shows the expected temperature profile through the ascent stage. Since focus through the development effort of OLite 2 had focused on the non-convective environment on orbit, the temperature profile through the atmosphere had been neglected. Along

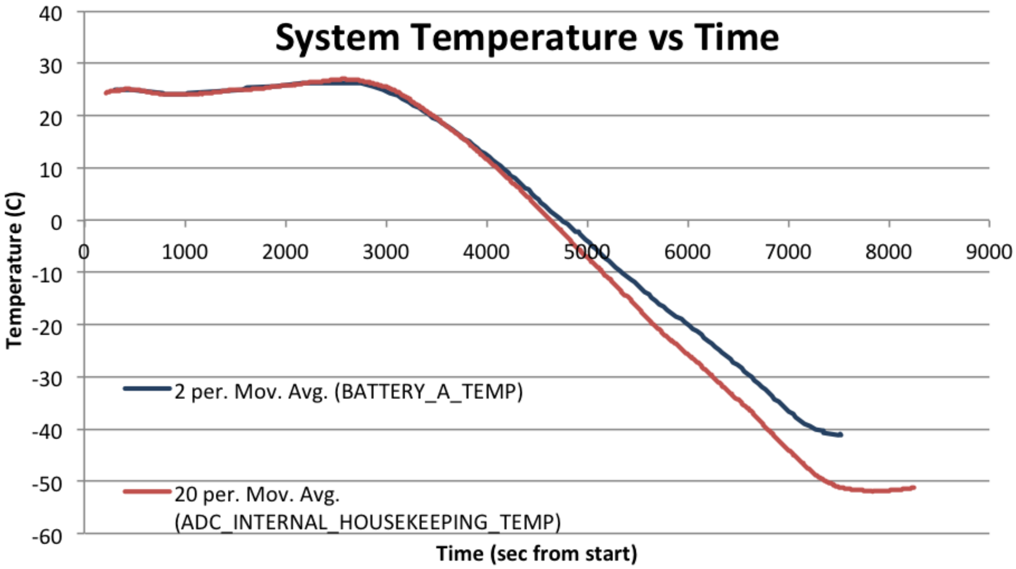


Figure 6.6: OLite 2 internal temperatures during HASP environmental test.

the ascent, the plot shows minima of –50 and –93 °C at 10 and 80 km, respectively. With the lower altitudes especially, convection dominates over radiative effects showing that the test profile during the HASP balloon environmental test were valid for the ascent stage.

To reduce the temperature effects, two actions were taken. The first was to add more heaters throughout the system to increase the overall system temperature. The second was to reduce the airflow through the electronics section. The airflow reduction was realized using blocks of foam insulation in the top and bottom of the CSS. This was a method observed during integration that many of the other payload teams implemented to protect their payloads. With these modifications, the system was tested at atmospheric pressure in the thermal chamber in SSPL following the HASP temperature profile and proved its ability to function nominally throughout the test.

6.3.3 HASP Flight Line Testing

6.3.3.1 Problem Description

Due to budgetary constraints on the OLite 2 project, the team relied on the HASP team members to interface OLite 2 on the flight system. Prior to shipping, the payload was fully verified to be functioning properly. This verification was in the form of a power on test prior to packaging the payload for shipping. This test consisted of powering the payload on, commanding the power on and off from the SB to the CSS, receiving and validating data from the CSS, and sending command to the CSS to verify commanding. With the completion of these tests, OLite 2 was packed up and shipped to the HASP flight line with instructions for the HASP team.

The general procedure for the HASP flight line testing was to make the mechanical and electrical connections as was done during initial HASP integration testing. When power was first applied to the CSS, it was observed that there was no data being received by the HASP balloon and that the system power draw was 10 mA on the 30-V power line. Initially, OLite 2 was expected to draw of 450 mA on the 30-V power line and automatically send data to the ground station. Additionally, the HASP team observed that the status LEDs within the CubeSat

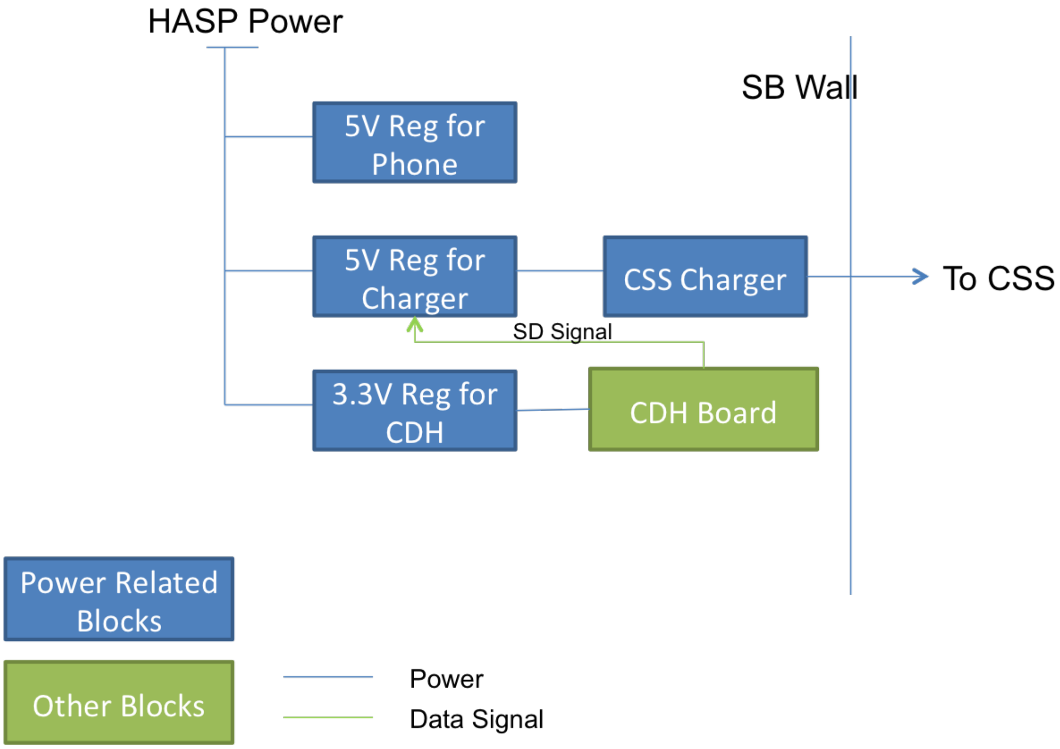


Figure 6.7: Support Box internal power connections.

were not lit, signifying that there was no external power getting to the CSS and the batteries were dead. When the umbilical connector on the outside of the SB was checked for power, it was found that the SB was not supplying power to the CSS.

For debugging purposes, Figure 6.7 shows the internal power and signal path within the SB. Power is provided to the CSS by means of a battery charger circuit targeted specifically to recharge the CSS battery during flight. This is, in turn, powered by a 5-V regulator drawing power from the 30-V HASP power line. The 5-V regulator also has a control signal line that allows the CDH system to turn off the 5-V regulator allowing the CSS to run off of internal power for a period of time.

6.3.3.2 Fault Analysis

A fault tree was developed to understand the possible conditions under which power would not be received by the CSS. Figure 6.8 shows the fault tree for the

CSS not receiving power. The fault tree analysis focused on system-level errors because these were the primary faults that could be understood and fixed in the field; board-level faults would have required shipment back to Penn State and a missed flight opportunity.

Two major categories of problems are evident from the fault tree, one being a CDH system malfunction and the other broken connections in the power distribution circuits. The CDH system controls the shutdown pin on the power supply to the battery charger. This pin could be pulled low to shut down the regulator by a CDH memory malfunction causing the CDH system not to boot, the CDH system to boot up into the CSS flight computer mode, or by CDH not receiving power. Likewise, broken wires or loose connections throughout the system could cause power to not be received where required.

From the information gathered by the HASP team, it is known that the umbilical cable is not causing the problem (since the connector on the outside of the SB does not have power), and enough power, 0.3 W, is being drawn to determine that the CDH system is powering up into the full Linux operating system eliminating the possibility of a 3.3-V power problem or the NAND being corrupted as these would cause zero to 0.1 W of power draw.

To continue debugging the system, the SB needs to be open. Once open, the following steps can be taken to isolate the problem. The order of steps was determined based on how invasive the procedure would be.

1. Check mechanical connection of power wires, inspect for damage to the wires
2. With SB power on, check CDH status LEDs and battery charger status LED
3. Remove CDH expansion board to isolate 5-V regulator shutdown pin
4. Probe power line voltages throughout the system

Many problems occur at interfaces, in this case they can be the mechanical interface between cables and boards. In the SB, a number of different types of connections including two types of screw terminals, one using ring lug crimps and the other a friction press on a wire between two plates, and solder joints. During transit, these can come loose and result in a broken or intermittent connection. A check of these connections is easily performed by using small tweezers to gently pull on the wire near the junction.

Checking the status LEDs on the CDH board shows the mode it is operating in. The CDH board is operating normally in SB mode when the LEDs are steadily lit;

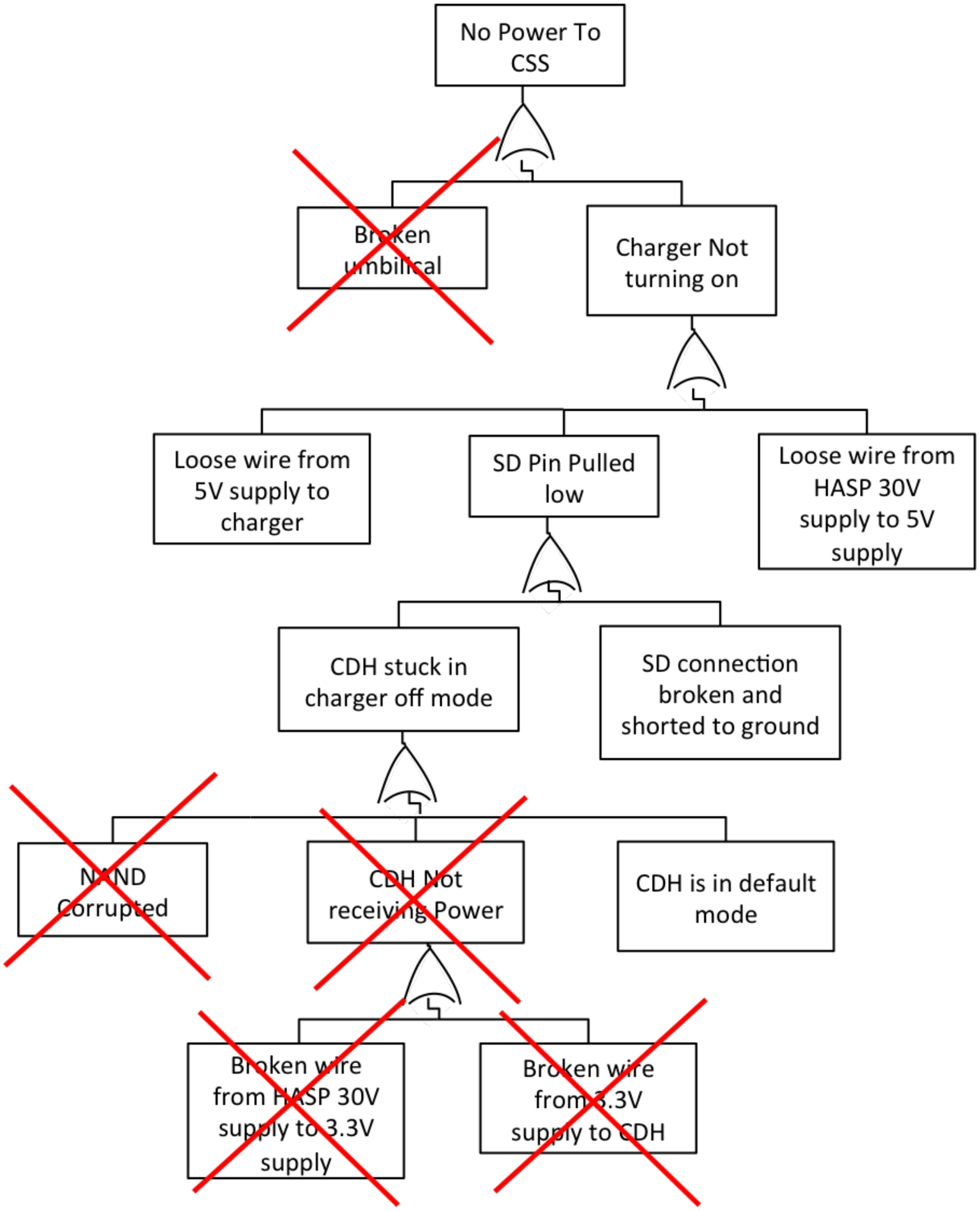

No Power To CSS
Broken umbilical
Charger Not turning on
Loose wire from 5V supply to charger
SD Pin Pulled low
Loose wire from HASP 30V supply to 5V supply
CDH stuck in charger off mode
SD connection broken and shorted to ground
NAND Corrupted
CDH Not receiving Power
CDH is in default mode
Broken wire from HASP 30V supply to 3.3V supply
Broken wire from 3.3V supply to CDH

Figure 6.8: Support Box power problem fault tree analysis.

flashing LEDs show that the CDH is in CSS mode or “default” mode and dimly lit LEDs show a problem with the NAND memory resulting in the CDH board being stuck while booting.

The shutdown pin on the 5-V power supply can be isolated by removing the expansion board from the CDH board. This would isolate the shutdown pin from any of the pull down that the CDH board may provide. Also, in conjunction with step one, this would eliminate the possibility of the control wire being shorted to ground by any damage.

While probing power lines with a digital multimeter is generally a good first step for system debugging, it is difficult to make these measurements within an integrated system. Deintegrating the system to get at component terminals to verify connections means reverifying the system as it is reconstructed. This can at cause more problems as components and connections get strained while removing the to access other parts.

6.3.3.3 Results

Without a Penn State team at the flight line to debug the system, OLite 2 was returned to Penn State and missed the flight. When OLite 2 returned, the problem was duplicated in bench testing. Removing the covers from the SB showed no immediate damage, but the 30-V line going to the 5-V power supply, which supplies power to the battery charger, came loose when tugged on. The power supply uses a screw terminal block with two plates to squeeze on a wire and use friction to hold the wire in place. The screw holding the plates together is believed to have come loose in transit to the launch site.

6.4 OLite 2 Project Conclusion

While the OLite 2 project experienced a number of problems during the system testing stage, system operation during environmental testing was able to meet many of the project objectives.

Student Education This goal was a success as students participating in all aspects of the projects were exposed to environmental testing and the effects different environments have on electronics. Additionally, students learned about and to appreciate the systems engineering process through experiencing the OLite 2 mission life cycle. Through the lessons learned during this testing effort, the students are prepared with the tools to continue development of the OSIRIS-3U system.

Communication The communications system has been demonstrated in bench testing to provide communications between remote terminals. Range testing around the State College, PA area has met the subsystem goals for the OLite 2 mission.

Guidance and Navigation The GPS and magnetometer systems preformed well through system testing. The GPS has demonstrated its ability to track satellites and provide timing and position data throughout system testing, and the magnetometer has performed flawlessly throughout. A proof-of-concept sun sensor prototype was developed and tested for application in the flight system. Integrating the sun sensors lead to an understanding of onboard signal integrity. With the concept design for the sun sensors completed and lessons learned documented, the guidance and navigation system meets the OLite 2 project objectives.

Power Power regulation, batteries, and solar panels worked as expected through system testing. While the environmental testing showed limitations of the batteries to operating in cold conditions, the concept was validated and will serve as the foundation for continued development. Solar panels revealed a high correlation between surface temperature and output power and the subsequent addition of a maximum power point tracking system (MPPT) to track the solar cell curve over temperature in future designs. Overall, the power system met its mission objectives through this development effort.

Thermal Through testing in the SSPL thermal vacuum chamber, the thermal subsystem proved its ability to model thermal effects in a convection-free vacuum environment. In doing so they have meet the thermal objectives for the OLite 2 mission as they pertain to an eventual orbital platform.

Flight Computer The flight computer has been able to demonstrate system operation and command handling throughout the test program meeting its overall mission objectives. Testing lead to thermal qualification of the CDH system provide appropriate thermal rated parts are selected.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Systems engineering is a broad subject with an even broader reach. For a systems engineer to complete his or her task, they need a broad understanding of the system at hand. For the OSIRIS-3U project, this has lead to the development of a detailed system model of the science mission to understand how system parameters affect the system's ability to meet mission requirements.

To this end, a model has been developed of the OSIRIS-3U system to understand the effects that a number of variables have on total science return. These variables include orbital lifetime, communications downlink time, power requirements, and science access count. This model lead to mission requirements on altitude and inclination, variables which most influence mission success, of 400–600 km and 25–110°, respectively.

Additionally, incremental development of the plasma impedance probe has lead to a better understanding of the probe system. The instrument theory has been demonstrated through bench testing with a tank circuit. Comparisons with network analyzer results show a strong correlation between the measurements taken by a network analyzer and the plasma impedance probe instrument at low frequencies and thus for low plasma densities.

Through the completion of this work, I have had the opportunity to act as the systems engineer for the OSIRIS-3U development program. This role has given me an appreciation for the systems engineering process, an understanding of the verification process, shown me the importance of testing early and often, and given me the tools to logically understand a problem and determine its root cause at many levels. Through work on the instrument, I have gained an appreciation for the understanding of the science mission and experienced the push–pull relationship between instrument and flight system requirements.

7.2 Lessons Learned

An important activity throughout a project is to gather the lessons learned along the way. These are important to ensure that future projects can learn from the efforts of those that came prior and avoid the same mistakes. Some lessons learned from the OLite 2 project are summarized below.

Make sure data and the computer program to access the data are stored together to allow retrieval of old data, and make sure all program files are revision controlled. While the importance of maintaining a backup of data is well known, it is easy to forget about the custom software that goes into making use of that data. In the case of the OLite 2 system, we found it extremely difficult to reanalyze the data nearly six months after project completion because development continued on the CubeSat, which resulted in changes to files supporting the analysis program. These files were not under revision control, making it impossible to revert to a working copy of the code to analyze the data.

Electrical connections should be made securely and with strain relief; solder joints, connectors, and ring lungs are preferred. The root cause of the OLite 2 mission failure came down to a loose wire, likely due to vibration during shipping loosening a screw holding a wire in place. Connections should be made to eliminate single points of failure by choosing methods that use redundant retention methods, e.g., a ring lug on a terminal block is held in place by the screw tightening down on top of the lug as well as the ring completely around the screw if the screw loosens. Additionally, strain relief should be used to prevent fatigue at the connections to reduce the likelihood of wires breaking.

Keep in mind the mission objectives. As a good systems engineering practice, it is important to focus on the primary mission objectives throughout the mission development, else students focus on adding features without fully testing the mission-critical aspects. Additionally, when crunch time comes, and it inevitably will, the mission objectives help prioritize the remaining work to focus on getting really needs to be completed.

Be aware of component temperature ratings and actively choose the appropriate device. The NAND flash memory used on the flight computer was available in two varieties, the commercial temperature range part was selected for the OLite 2 flight computer board without noticing the temperature range and this caused days of lost time. While the component appeared to meet system needs during testing, it failed when integrated during flight system testing.

Understand the test environments and how it represents the mission profile. The temperature profile during ascent was overlooked while focusing on thermal temperature for float altitudes. This lead to the test environment in the SSPL environmental chamber to differ from the integration test environment, and after further review, the actual launch profile.

In addition to normal digital communication methods, include externally visible indicators for important components. While OLite 2 was on the flight line, LED indicators showing power state at various points and communication activity would serve to help the debugging process to by indicating which power supplies were active and allow easier system debug.

7.3 Future Work

7.3.1 OSIRIS-3U System Work

The next step for the OSIRIS-3U system is to develop a functional hardware-in-the-loop (HIL) test model taking inputs from the system model previously developed and driving functionality on the CubeSat. While many of the students have worked to understand the operation of a given subsystem, there is much learning to occur regarding the operation of the OSIRIS-3U system as a whole. The HIL simulation will allow students to understand how changing system parameters influences the system as a whole. This would include the impact of changing downlink rates and overhead on accumulated data downlinked, monitoring power reserves through an accurate representation of the battery's inputs and outputs on orbit, and allowing students to begin to understand the system response to faults that may occur.

Much of the ground work has been laid, but the student engineers picking up this work will have the responsibility of bringing the hardware systems together into a robust, operational system. Student engineers will need to understand how CDH transitions between operational states in normal operation and what data needs to be relayed as part of the beacon packet to understand when an error has occurred. This will include building many more system fault trees and then building a response plan to determine when CDH can handle an anomaly and when it requires ground intervention.

The importance of a long term data management program became apparent while preparing this work. The OSIRIS-3U development program has been a multiyear development effort with two prototypes, OLite and OLite 2, having been developed. Over the course of these development efforts, data have been generated from each of the projects using custom data formats fore each project. Future work should develop a common data handling methodology to facilitate data recall and analysis from past missions and allow easy integration for future projects.

Overall, the OSIRIS-3U program is on track in its efforts to build a robust system. With many of the subsystems understood, it is now time for the SSPL student engineers to turn their attention towards how these systems work together to complete the mission.

7.3.2 Instrument Development

While much work has been done on the science instrument, it has been left with many open questions. Specifically, the understanding of the instrument response at high frequencies. To understand the problem better, some more detailed circuit modeling should be done to characterize the errors generated within the circuit including op amp bias currents, the effect of the dummy probe circuit, and frequency spurs near the desired stimulus frequency. Additional verification of the open–short–load method should be considered including using a non-idealistic model for the terminators and investigate the phase variation between the instrument and manual calibration on data taken from the network analyzer. Once basic operation is established, and functionality is demonstrated, the system should be optimized using a digital control system to track the resonance frequency. This can lead to improvements in measurement accuracy and spatial resolution while decreasing total data volume, making the resulting plasma frequency probe a desirable instrument for use on small satellite missions.

Appendix A

Monte Carlo Simulation Code

```
%% SSPL OSIRIS Monte Carlo Simulator
%
% Authors:
%   Allen Kummer    atk5025@psu.edu
%   Andrew Palski   amp5500@psu.edu
%
% NOTE: STK must be running and have the scenario open before running this script
%
% STK Link Command Reference:
%       http://www.agi.com/resources/help/online/stk/index.html
%
% Important Variable Definitions Column Definitions
% ---------------------
% numOrbits Number of Orbits in the simulation period
%
% a Orbit semimajor axis in meters referenced from the center of the Earth
% e Orbit ecentricity
% incl Orbit inclination
% lowOmega Argument of Perigee [Degrees]
% capOmega RAAN Right Ascension of the ascending node [Degrees]
% theta Mean Anomaly [Degrees]
%
% countTimesInScience  Number of times entering into the science region
% timeInScience  Sum total of time in the science region per simulation period
% meanScienceAccess  Average time in the science region per simulation period
% maxScienceAccess  Maximum time in the science region per simulation period
%
% downlinkEventCount  Number of downlink events per simulation period
% totalDown Sum total of downlink event access times per simulation period
% meanDown Average downlink event access time per simulation period
% maxDown maximum downlink event access time per simulation period
% minDown minimum downlink event access time per simulation period
%
```

```
% energyAvg Average power gathered during the simulation period
% lifetime              Spacecraft lifetime

%% User Variables
n=10000; %number of iterations
outputFileName = 'csvlist.csv';
DurationLimit = 99999;
minAltitude = 300;
maxAltitude = 700;

%% Configure Link
uiapp=actxGetRunningServer('STK9.application'); %sets up the connection
root=uiapp.Personality2; %gives control over the connection
scen=root.CurrentScenario; %sets up the scenario
root.UnitPreferences.SetCurrentUnit('DateFormat','EpSec');
start=scen.StartTime;
stop=scen.StopTime;

Satellite=scen.Children.Item('Sat');
Sensor=root.GetObjectFromPath('*/Facility/Propogated_point').Children.Item('Sensor1');
Sensor2=root.GetObjectFromPath('*/Facility/State_College').Children.Item('Sensor2');
root.ExecuteCommand(['SetLifetime */Satellite/Sat DragCoeff 2.2' ...
    [' ReflectCoeff 1 DragArea .03 SunArea .01 Mass' ...
    [' 4 DecayAltitude 0 FluxSigmaLevel 1 2ndOrder' ...
    ' On Rotate On Graphics Off DensityModel Jacchia71']);

%% Initialize Variables
description=zeros(n,6); %a description of the orbit
timeInSun=zeros(n,1); %total time in sun
meanSunAccess=zeros(n,1);
maxSunAccess=zeros(n,1);
minSunAccess=zeros(n,1);

meanSunOfOrbit=zeros(n,1); %time as a percentage of the orbit
maxSunOfOrbit=zeros(n,1);
minSunOfOrbit=zeros(n,1);

totalDown=zeros(n,1); %Access to groundstation
meanDown=zeros(n,1);
maxDown=zeros(n,1);
minDown=zeros(n,1);

countTimesInScience=zeros(n,1); %total time in science regions
timeInScience=zeros(n,1); %total time in science regions
meanScienceAccess=zeros(n,1);
maxScienceAccess=zeros(n,1);

lifetime=zeros(n,2); %columns for altitude and lifetime

energyTot=zeros(n,1);
```

```
energyAvg=zeros(n,1); %per orbit
numOrbits=zeros(n,1);

timestep=60;
tic

%% Open Simulation Datafile to Save To
outputFileHeader = { ...
        'Simulation Number',        ...
        '# of Orbits',              ...
        'Altitude [km]',            ...
        'Ecentricity',              ...
        'Inclination [deg]',        ...
        'Argument of Perigee [deg]',...
        'RAAN [deg]',               ...
        'Mean Anomaly [deg]',       ...
        'Orbit Time [s]',           ...
        '# of Sci Accesses',        ...
        'Total Sci Time [s]',       ...
        'Mean Sci Time [s]',        ...
        'Max Sci Time [s]',         ...
        '# of PSU Accesses',        ...
        'Total Downlink Time [s]',  ...
        'Mean Downlink Time [s]',   ...
        'Max Downlink Time [s]',    ...
        'Min Downlink Time [s]',    ...
        'Mean Orbit Energy [Whr]',  ...
        'Lifetime [days]'           ...
    };

csvFun = @(str)sprintf('%s,',str);
xchar = cellfun(csvFun, outputFileHeader, 'UniformOutput', false);
xchar = strcat(xchar{:});
xchar = strcat(xchar(1:end-1),'\n');

fid=fopen(outputFileName,'wt');
fprintf(fid,xchar);
fclose(fid);

%% Create Wait Bar
progressBar = waitbar(i/n,sprintf(['Orbit Simulation 0 of' ...
' %i',n), 'Name', 'Orbit Monte Carlo');

%% Start Monte Carlo Simulation
for i=1:n
waitbar(i/n,progressBar,sprintf('Orbit Simulation %i of %i',i,n));

    a = ((maxAltitude-minAltitude) * rand(1) + minAltitude + 6371)*1000;
    T = 2*pi*sqrt((a/1000)^3/3.986E5);
    lifetime(i,1) = a/1000-6371;
```

```
   e = abs(.01*randn(1)); %normal distribution with stddev of .01...closely
       %packs random numbers to 0

   check=rand(1);
   if check<.25
       incl=28.5;%cape
   elseif check<.5
       incl=(110-50)*rand(1)+50;%vandenburg
   elseif check<.75
       incl=51.6;%iss
   else
       incl=100*rand(1);%random
   end
   lowOmega=360*rand(1);
   capOmega=360*rand(1);
   theta=360*rand(1);
   des=[num2str(a) ’ ’ num2str(e) ’ ’ num2str(incl) ’ ’ num2str(lowOmega) ...
’ ’ num2str(capOmega) ’ ’ num2str(theta)];
   description(i,:)=[a e incl lowOmega capOmega theta];
   command=[’SetState */Satellite/Sat Classical J4Perturbation’ ...
    ’ UseScenarioInterval 10 J2000 "10 Oct 2011 16:00:00.00"’ des];
   root.ExecuteCommand(command);

   try
       %Access for primary science area
       access=Satellite.GetAccessToObject(Sensor);
       access.ComputeAccess;
       accessDP=access.DataProviders.Item([’Access’ ...
   ’ Data’).Exec(scen.StartTime,scen.StopTime);
       accessDurations=accessDP.DataSets.GetDataSetByName(’Duration’).GetValues;

       dur=zeros(length(accessDurations),1);
       for j=1:length(accessDurations)
           dur(j)=accessDurations{j};
       end
       countTimesInScience(i)=length(dur);
       timeInScience(i)=nansum(dur);
       meanScienceAccess(i)=nanmean(dur);
       maxScienceAccess(i)=max(dur);
   catch
       countTimesInScience(i)=0;
       timeInScience(i)=0;
       meanScienceAccess(i)=0;
       maxScienceAccess(i)=0;
   end

   try
       %Access for state college
       access=Satellite.GetAccessToObject(Sensor2);
       access.ComputeAccess;
```

```
        accessDP=access.DataProviders.Item(['Access' ...
    ' Data').Exec(scen.StartTime,scen.StopTime);
        accessDurations=accessDP.DataSets.GetDataSetByName('Duration').GetValues;

        dur=zeros(length(accessDurations),1);
        for j=1:length(accessDurations)
            dur(j)=accessDurations{j};
        end

        downlinkEventCount(i)=length(accessDurations);
        totalDown(i)=nansum(dur);
        meanDown(i)=nanmean(dur);
        maxDown(i)=max(dur);
        minDown(i)=min(dur);
    catch
        downlinkEventCount(i)=0;
        totalDown(i)=0;
        meanDown(i)=0;
        maxDown(i)=0;
        minDown(i)=0;
    end

    %Access to sun
    access=Satellite.GetAccessToObject(root.GetObjectFromPath('*/Planet/Sun'));
    access.ComputeAccess;
    accessDP=access.DataProviders.Item(['Access' ...
    ' Data').Exec(scen.StartTime,scen.StopTime);
    sunDurations=accessDP.DataSets.GetDataSetByName('Duration').GetValues;
    SunstartTime=accessDP.DataSets.GetDataSetByName('Start Time').GetValues;
    SunstopTime=accessDP.DataSets.GetDataSetByName('Stop Time').GetValues;

    dur=zeros(length(sunDurations)-2,1);
%     durPercent=zeros(length(accessDurations)-2,1);
    for j=2:length(sunDurations)-1
        dur(j-1)=sunDurations{j};
        durPercent(j-1)=dur(j-1)/T;
    end
    sunStart=zeros(length(SunstartTime),1);
    sunStop=zeros(length(SunstopTime),1);
    for j=1:length(SunstartTime)
        sunStart(j)=SunstartTime{j};
        sunStop(j)=SunstopTime{j};
    end

    %Satellite Lifetime
    root.ExecuteCommand([sprintf(['setLifetime */Satellite/Sat' ...
    [' DurationLimit %i DecayAltitude 100 Rotate On' ...
     ' LimitType Duration DragArea .03 Mass 4', ...
     DurationLimit)]);
    result = root.ExecuteCommand(['Lifetime */Satellite/Sat']);
    split_result = regexp(result.Item(0), ' ', 'split');
try
        if length(split_result) == 9
            lifetime(i,2)=DurationLimit;
        elseif strcmp(split_result{18}, 'days.') % Check to see if time is < 1 year
            lifetime(i,2)=str2num(split_result{17}); %days
        elseif strcmp(split_result{18}, 'years.') % Check to see if time > 1 year
            lifetime(i,2)=str2num(split_result{17})*365;
        else
            lifetime(i,2)=NaN;
        end

    catch
lifetime(i,2) = 0;
end

    %calculating power
    elems={'x';'y';'z'};
    satDP=Satellite.DataProviders.Item(['Sun' ...
    ' Vector').Group.Item('J2000').ExecElements(start, ...
  stop, timestep, elems);
    sun1= cell2mat(satDP.DataSets.Item(cast(0,'int32')).GetValues);
    sun2= cell2mat(satDP.DataSets.Item(cast(1,'int32')).GetValues);
    sun3= cell2mat(satDP.DataSets.Item(cast(2,'int32')).GetValues);
    sun=[sun1,sun2,sun3];

    elems = {'q1';'q2';'q3';'q4'};
    satDP = Satellite.DataProviders.Item(['Attitude' ...
    ' Quaternions').ExecElements(start, stop, timestep, ...
 elems);

    q2 = cell2mat(satDP.DataSets.Item(cast(0,'int32')).GetValues);
    q3 = cell2mat(satDP.DataSets.Item(cast(1,'int32')).GetValues);
    q4 = cell2mat(satDP.DataSets.Item(cast(2,'int32')).GetValues);
    q1 = cell2mat(satDP.DataSets.Item(cast(3,'int32')).GetValues);
    q=[q1,q2,q3,q4];
    dcm=Quat2vector(q);

    check=false;
    count=1;

    for j=1:(stop-start)/timestep
        if count<=length(sunStart) && abs((j-1)*timestep-sunStart(count)) < timestep
            check=true;
        elseif count<=length(sunStart) && abs((j-1)*timestep-sunStop(count)) < timestep
            check=false;
            count=count+1;
        end

        if check
            x  =dcm(:,:,j)'*[ 1; 0; 0];
            xn =dcm(:,:,j)'*[-1; 0; 0];
            y  =dcm(:,:,j)'*[ 0; 1; 0];
            yn =dcm(:,:,j)'*[ 0;-1; 0];
            z  =dcm(:,:,j)'*[ 0; 0; 1];
            zn =dcm(:,:,j)'*[ 0; 0;-1];

            % +X solar panel in STK is the +X in OSIRIS
            angle=acos(dot(x,sun(j,:))/norm(x)/norm(sun(j,:)));
            if angle<(pi/2)
                energyTot(i)=energyTot(i)+(aoi2power(angle,'+x')*timestep/3600);
            end

            % -X solar panel in STK is the -X in OSIRIS
            angle=acos(dot(xn,sun(j,:))/norm(xn)/norm(sun(j,:)));
            if angle<(pi/2)
                energyTot(i)=energyTot(i)+(aoi2power(angle,'-x')*timestep/3600);
            end

            % +Y solar panel in STK is the +Y in OSIRIS
            angle=acos(dot(y,sun(j,:))/norm(y)/norm(sun(j,:)));
            if angle<(pi/2)
                energyTot(i)=energyTot(i)+(aoi2power(angle,'+y')*timestep/3600);
            end

            % -Y solar panel in STK is the -Y in OSIRIS
            angle=acos(dot(yn,sun(j,:))/norm(yn)/norm(sun(j,:)));
            if angle<(pi/2)
                energyTot(i)=energyTot(i)+(aoi2power(angle,'-y')*timestep/3600);
            end

            % +Z solar panel in STK is the +Z in OSIRIS
            angle=acos(dot(z,sun(j,:))/norm(z)/norm(sun(j,:)));
            if angle<(pi/2)
                energyTot(i)=energyTot(i)+(aoi2power(angle,'+z')*timestep/3600);
            end

            % -Z solar panel in STK is the -Z in OSIRIS
            angle=acos(dot(zn,sun(j,:))/norm(zn)/norm(sun(j,:)));
            if angle<(pi/2)
                energyTot(i)=energyTot(i)+(aoi2power(angle,'-z')*timestep/3600);
            end
        end

    end

    numOrbits(i)=floor((stop-start)/T);
    energyAvg(i)=energyTot(i)/numOrbits(i);

    dataToWrite = [...
```

```
        i, ...
        numOrbits(i), ...
        lifetime(i,1), ...
        e, ...
        incl, ...
        lowOmega, ...
        capOmega, ...
        theta, ...
        T, ...
        countTimesInScience(i), ...
        timeInScience(i), ...
        meanScienceAccess(i), ...
        maxScienceAccess(i), ...
        downlinkEventCount(i), ...
        totalDown(i), ...
        meanDown(i), ...
        maxDown(i), ...
        minDown(i), ...
        energyAvg(i), ...
        lifetime(i,2)
        ];

    if(length(dataToWrite) == length(outputFileHeader))
        dlmwrite(outputFileName, dataToWrite, '-append', 'delimiter', ',');
    else
        error('MonteCarlo:dataToWriteLengthCheck', ['Length of file' ...
    ' header and data to be written do not match');
    end

end

toc
%% Close progress bar
close(progressBar)

%% AOI2POWER
%  Alex Hackett
%  Fall 2011

function pOut = aoi2power( aoi, panel, age )
% Computes power output of a solar panel, given angle-of-incidence and a
%    panel label.
%
%     pOut:  output power in Watts
%      aoi:  input angle-of-incidence in radians, in the interval [0, pi/2]
%    panel:  input panel label, must match one of the following:
%            '-x', '+x', '-y', '+y', '-z', or '+z'
%      age:  age of the panel (optional), if included, must match one of
%            the following:
%            'BOL', 'EOL'
%           'EOL' is the default value

    % Argument parsing
    if( nargin == 2 )
        age = 'EOL';
    elseif ( nargin == 3 )
    else
        disp( 'Incorrect number of arguments.  See aoi2power help.' );
        pOut = 0;
        return;
    end

    % Set age parameters appropriately
    if( strcmp( age, 'BOL' ) )
        cellEff = .24;              % Average solar cell efficiency BOL
        cellVtempCoeff = -6.7;      % Average solar cell voltage
                                    % temperature coefficient
                                    % [mV/C] BOL
        cellJtempCoeff = 6;         % Average solar cell current
                                    % density temperature
                                    % coefficient [uA/cm^2/C] BOL
    else
        if( ~strcmp( age, 'EOL' ) )
            disp( 'Invalid age.  Defaulting to ''EOL''.' );
        end

        cellEff = .21;              % Average solar cell efficiency EOL
        cellVtempCoeff = -7.2;      % Average solar cell voltage
                                    % temperature coefficient
                                    % [mV/C] EOL
        cellJtempCoeff = 14;        % Average solar cell current
                                    % density temperature
                                    % coefficient [uA/cm^2/C] EOL

    end

    % Solar Panel parameters
    nXcells = 34;               % # of cells on -X panel
    pXcells = 52;               % # of cells on +X panel
    nYcells = 52;               % # of cells on -Y panel
    pYcells = 52;               % # of cells on +Y panel
    nZcells = 18;               % # of cells on -Z panel
    pZcells = 0;                % # of cells on +Z panel
    cellArea = 2.277;           % Area of solar cell [cm^2]
    cellVmp = 2.19;             % Solar cell voltage (at max power) [V]
    cellImp = 0.028;            % Solar cell current (at max power) [A]
    cellTemp = 80;              % Average solar cell temperature [C]
    irrSun = 1350;              % Average solar irradiance [W/m^2]
    itoEff = 0.9;               % Transmission efficiency of ITO + glass/acryllic combo
```

```
% Variables
nXarea = nXcells*cellArea/10000;       % -X solar area [m^2]
pXarea = pXcells*cellArea/10000;       % +X solar area [m^2]
nYarea = nYcells*cellArea/10000;       % -Y solar area [m^2]
pYarea = pYcells*cellArea/10000;       % +Y solar area [m^2]
nZarea = nZcells*cellArea/10000;       % -Z solar area [m^2]
pZarea = pZcells*cellArea/10000;       % +Z solar area [m^2]

pOutMax = 0;

% Intermediate calculation
pTempLoss = -cellVtempCoeff/1000 * ( cellTemp - 25 ) * cellImp;
pTempGain = cellJtempCoeff/1e6 * ( cellTemp - 25 ) * cellVmp;

% Panel specific
if ( strcmpi( panel, '-x' ) )
    pTempLoss = pTempLoss * nXcells;
    pTempGain = pTempGain * nXarea;
    pOutMax = irrSun * nXarea - pTempLoss + pTempGain;
elseif ( strcmpi( panel, '+x' ) )
    pTempLoss = pTempLoss * pXcells;
    pTempGain = pTempGain * pXarea;
    pOutMax = irrSun * pXarea - pTempLoss + pTempGain;
elseif ( strcmpi( panel, '-y' ) )
    pTempLoss = pTempLoss * nYcells;
    pTempGain = pTempGain * nYarea;
    pOutMax = irrSun * nYarea - pTempLoss + pTempGain;
elseif ( strcmpi( panel, '+y' ) )
    pTempLoss = pTempLoss * pYcells;
    pTempGain = pTempGain * pYarea;
    pOutMax = irrSun * pYarea - pTempLoss + pTempGain;
elseif ( strcmpi( panel, '-z' ) )
    pTempLoss = pTempLoss * nZcells;
    pTempGain = pTempGain * nZarea;
    pOutMax = irrSun * nZarea - pTempLoss + pTempGain;
elseif ( strcmpi( panel, '+z' ) )
    pTempLoss = pTempLoss * pZcells;
    pTempGain = pTempGain * pZarea;
    pOutMax = irrSun * pZarea - pTempLoss + pTempGain;
else
    fprintf( 'Unknown panel: %s', num2str( panel ) );
end

% Calculate output final output power, taking efficiencies and AOI into
% account
pOut = cos( aoi ) * itoEff * pOutMax * cellEff;

% If we end up with a negative value (loss > generated power or an
% incorrect angle is given), we don't have any power out.
if( pOut < 0 )
```

```
        pOut = 0;
    end

end
```

Appendix B

Solar Panel Power Calculator

```
%% AOI2POWER
%  Fall 2011

function pOut = aoi2power( aoi, panel, age )
% Computes power output of a solar panel, given angle-of-incidence and a
%    panel label.
%
%     pOut:  output power in Watts
%      aoi:  input angle-of-incidence in radians, in the interval [0, pi/2]
%    panel:  input panel label, must match one of the following:
%            '-x', '+x', '-y', '+y', '-z', or '+z'
%      age:  age of the panel (optional), if included, must match one of
%            the following:
%            'BOL', 'EOL'
%            'EOL' is the default value

    % Argument parsing
    if( nargin == 2 )
        age = 'EOL';
    elseif ( nargin == 3 )
    else
        disp( 'Incorrect number of arguments.  See aoi2power help.' );
        pOut = 0;
        return;
    end

    % Set age parameters appropriately
    if( strcmp( age, 'BOL' ) )
        cellEff = .24;              % Average solar cell efficiency BOL
        cellVtempCoeff = -6.7;      % Average solar cell voltage
                                    % temperature coefficient
                                    % [mV/C] BOL
        cellJtempCoeff = 6;         % Average solar cell current
```

```
                                % density temperature
                                % coefficient [uA/cm^2/C] BOL
else
    if( ~strcmp( age, 'EOL' ) )
        disp( 'Invalid age.  Defaulting to ''EOL''.' );
    end

    cellEff = .21;              % Average solar cell efficiency EOL
    cellVtempCoeff = -7.2;      % Average solar cell voltage
                                % temperature coefficient
                                % [mV/C] EOL
    cellJtempCoeff = 14;        % Average solar cell current
                                % density temperature
                                % coefficient [uA/cm^2/C] EOL

end

% Solar Panel parameters
nXcells = 34;               % # of cells on -X panel
pXcells = 52;               % # of cells on +X panel
nYcells = 52;               % # of cells on -Y panel
pYcells = 52;               % # of cells on +Y panel
nZcells = 18;               % # of cells on -Z panel
pZcells = 0;                % # of cells on +Z panel
cellArea = 2.277;           % Area of solar cell [cm^2]
cellVmp = 2.19;             % Solar cell voltage (at max power) [V]
cellImp = 0.028;            % Solar cell current (at max power) [A]
cellTemp = 80;              % Average solar cell temperature [C]
irrSun = 1350;              % Average solar irradiance [W/m^2]
itoEff = 0.9;               % Transmission efficiency of ITO + glass/acryllic combo

% Variables
nXarea = nXcells*cellArea/10000;       % -X solar area [m^2]
pXarea = pXcells*cellArea/10000;       % +X solar area [m^2]
nYarea = nYcells*cellArea/10000;       % -Y solar area [m^2]
pYarea = pYcells*cellArea/10000;       % +Y solar area [m^2]
nZarea = nZcells*cellArea/10000;       % -Z solar area [m^2]
pZarea = pZcells*cellArea/10000;       % +Z solar area [m^2]

pOutMax = 0;

% Intermediate calculation
pTempLoss = -cellVtempCoeff/1000 * ( cellTemp - 25 ) * cellImp;
pTempGain = cellJtempCoeff*1e4/1e6 * ( cellTemp - 25 ) * cellVmp;

% Panel specific
if ( strcmpi( panel, '-x' ) )
    pTempLoss = pTempLoss * nXcells;
    pTempGain = pTempGain * nXarea;
    pOutMax = irrSun * nXarea - pTempLoss + pTempGain;
    elseif ( strcmpi( panel, '+x' ) )
        pTempLoss = pTempLoss * pXcells;
        pTempGain = pTempGain * pXarea;
        pOutMax = irrSun * pXarea - pTempLoss + pTempGain;
    elseif ( strcmpi( panel, '-y' ) )
        pTempLoss = pTempLoss * nYcells;
        pTempGain = pTempGain * nYarea;
        pOutMax = irrSun * nYarea - pTempLoss + pTempGain;
    elseif ( strcmpi( panel, '+y' ) )
        pTempLoss = pTempLoss * pYcells;
        pTempGain = pTempGain * pYarea;
        pOutMax = irrSun * pYarea - pTempLoss + pTempGain;
    elseif ( strcmpi( panel, '-z' ) )
        pTempLoss = pTempLoss * nZcells;
        pTempGain = pTempGain * nZarea;
        pOutMax = irrSun * nZarea - pTempLoss + pTempGain;
    elseif ( strcmpi( panel, '+z' ) )
        pTempLoss = pTempLoss * pZcells;
        pTempGain = pTempGain * pZarea;
        pOutMax = irrSun * pZarea - pTempLoss + pTempGain;
    else
        fprintf( 'Unknown panel: %s', num2str( panel ) );
    end

    % Calculate output final output power, taking efficiencies and AOI into
    % account
    pOut = cos( aoi ) * itoEff * pOutMax * cellEff;

    % If we end up with a negative value (loss > generated power or an
    % incorrect angle is given), we don't have any power out.
    if( pOut < 0 )
        pOut = 0;
    end

end
```

Bibliography

[1] Futron (2002) *Space transporation costs: trends in price per pound to orbit 1990-2000*, *Tech. rep.*, Futron Corporation.

[2] Fortescue, P., G. Swinerd, and J. Stark (2011) *Spacecraft Systems Engineering*, 4 ed., John Wiley and Sons.

[3] Heidt, M. H., P. J. Puig-Suari, P. A. S. Moore, P. S. Nakasuka, and P. R. J. Twiggs (2000) "CubeSat: a new generation of picosatellite for education and industry low-cost space experimentation," in *14th Annual/USU Conference on Small Satellites*.

[4] Cal Poly (Accessed 2012) *Cal Poly website*.
URL `http://calpoly.edu`

[5] Anonymous (2012), "NORAD two-line element sets current data," .
URL `http://celestrak.com/NORAD/elements/`

[6] Oltrogge, D. L. and K. Leveque (2011) "An evaluation of CubeSat orbital decay," in *25th Annual AIAA/USU Conference on Small Satellites*.

[7] Wallace, J. M. and P. V. Hobbs (2006) *Atmospheric Science: An Indtroductory Survey*, Elsevier, Inc.

[8] Odenwald, S. (2009), "The day the sun brought darkness," .
URL `http://www.nasa.gov/topics/earth/features/sun_darkness.html`

[9] Bernhardt, P. A., R. P. McCoy, K. Dymond, J.M.Picone, R. R. Meier, F. Kamalabadi, D. Cotton, S. Charkrabarti, T. A. Cook, J. S. Vickers, A. W. Stephan, L. Kersley, S. E. Pryse, I. K. Walker, C. N. Mitchell, P. R. Straus, H. Na, C. Biswas, G. R. Kronschnabl, and T. D. Raymund (1998) "Two-dimensional mapping of the plasma density in the upper atmosphere with computerized ionospheric tomography (CIT)," *Physics of Plasmas*, **5**, pp. 2010–2021.

[10] BERNHARDT, P. A. and C. L. SIEFRING (2006) "New satellite-based systems for ionospheric tomography and scintillation region imaging," *Radio Science*, **41**, p. 14.

[11] BERNHARDT, P. A., C. L. SIEFRING, T. F. R. IVAN J. GALYSH, D. E. KOCH, T. L. MACDONALD, M. R. WILKENS, and G. P. LANDIS (2006) "Ionospheric applications of the scintillation and tomography receiver in space (CITRIS) mission when used with the DORIS radio beacon network," *Journal of Geodesy*, **80**, pp. 473–485.

[12] RAYMUND, T. D., J. R. AUSTEN, S. J. FRANKE, C. H. LIU, J. A. KLOBUCHAR, and J. STALKER (1990) "Application of computerized tomography to the investigation of ionospheric structures," *Radio Science*, **25**, pp. 771–789.

[13] HAJJ, G. A., E. R. KURSINSKI, L. J. ROMANS, W. I. BERTIGER, and S. S. LEROY (2002) "A technical description of atmospheric sounding by GPS occultation," *Journal of Atmospheric and Solar-Terrestrial Physics*, **64**, pp. 451–469.

[14] STRAUS, P. R. (2007) "Ionospheric climatology derived from GPS occultation observations made by the ionospheric occultation experiment," *Advances in Space Research*, **39**, pp. 793–802.

[15] KREHBIEL, J. P., L. H. BRACE, R. F. THEIS, W. H. PINKUS, and R. B. KAPLAN (1981) "The Dynamics Explorer Langmuir probe instrument," *Space Science Instrumentation*, **5**, pp. 493–502.

[16] MOTT-SMITH, H. M. and I. LANGMUIR (1926) "The theory of collectors in gaseous discharges," *Physics Review*, **28**, pp. 727–763.

[17] ESCOBAR, A. (2009) *A Langmuir probe instrument for research in the terrestrial ionosphere*, Master's thesis, The Pennsylvania State University.
URL `https://etda.libraries.psu.edu/paper/9629/4814`

[18] CHEN, F. F. (2006) *Introduction to Plasma Physics and Controlled Fusion*, vol. 1: Plasma Physics, 2 ed., Springer Science+Business Media, LLC.

[19] SPENCER, E., S. PATRA, T. ANDRIYAS, C. SWENSON, J. WARD, and A. BARJATYA (2008) "Electron density and electron neutral collision frequency in the ionosphere using plasma impedance probe measurements," *Journal of Geophysical Research*, **113**, p. 9.

[20] NASA (2007), "NASA/SP-2007-6105 NASA Systems Engineering Handbook," .

[21] (2012), "What is systems engineering?" .
URL `http://www.incose.org/practice/whatissystemseng.aspx`

[22] INCOSE (2012), "Systems Engineering Body of Knowledge," .
URL `http://www.sebokwiki.org/`

[23] Hooks, I. (1994) *Managing requirements*, *Tech. rep.*, Requirements Experts.

[24] Boehm, B., R. Valerdi, and E. Honour (2008) "The ROI of systems engineering: some quantitative results for software-intesnse systems," *Systems Engineering*, **11**, p. 221234.

[25] Boehm, B. (1986) "A sprial model of software development and enhancement," *ACM SIGSOFT Software Engineering Notes*, **11**, pp. 14–24.

[26] Bilén, S. G. (2011) "Introduction to systems design," Class Lecture.

[27] Mankins, J. C. (1995) *Technology readiness levels*, *Tech. rep.*, NASA Advanced Concepts Office.

[28] (2012), "Fault tree analysis programs," .
URL `http://www.enre.umd.edu/tools.htm`

[29] Spectrolab Inc. "Datasheet: 25.1% GaInP2/GaAs/Ge Triple Junction Solar Cells," .

[30] ——— "Datasheet: Triangular Advanced Solar Cells (TASC)," .

[31] King, J. (2003), "AMSAT-IARU link budget spreadsheet rev 1," .
URL `http://www.amsat.org/amsat/ftp/software/spreadsheet/AMSAT-IARU_Link_Budget_Rev1.xls`

[32] CalPoly (2009), "CubeSat design specification," .

[33] IADC (2002) "IADC space debris mitigation guidelines," in *United Nations Inter-Agency Space Debris Coordination Committee, Document IADC-02-01.*

[34] Siegel, R. D. (2002) *Design of digitally controlled langmuir swept and fixed biased probes*, Honor's thesis, The Pennsylvania State University.

[35] ——— (2004) *Design of a hybrid plasma probe system*, Master's thesis, The Pennsylvania State University.

[36] Schratz, B. (2006) *Design of a sounding rocket plasma frequency probe*, Honor's thesis, The Pennsylvania State University.

[37] Carlson, C. G. (2004) *Next generation plasma frequency probe instrumentation technique*, Master's thesis, Utah State University.