2015

Design and Development of the WVU Advanced Technology Satellite for Optical Navigation

Miranda Straub

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

Recommended Citation

Straub, Miranda, "Design and Development of the WVU Advanced Technology Satellite for Optical Navigation" (2015). *Graduate Theses, Dissertations, and Problem Reports*. 6737.
<https://researchrepository.wvu.edu/etd/6737>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

Design and Development of the WVU Advanced Technology Satellite for Optical Navigation

Miranda Straub

Thesis submitted to the
Benjamin M. Statler
College of Engineering and Mineral Resources
at West Virginia University

in partial fulfillment of the requirements for the degree of
Master of Science in Aerospace Engineering

John Christian, Chair, Ph.D.
Alfred Lynam, Ph.D.
John Kuhlman, Ph.D.

Department of Mechanical and Aerospace Engineering

Morgantown, West Virginia
2015

Keywords: cubesat, optical navigation, extended Kalman filter

Copyright 2015 Miranda Straub

Abstract

Design and Development of the WVU Advanced Technology Satellite for Optical Navigation

Miranda Straub

In order to meet the demands of future space missions, it is beneficial for spacecraft to have the capability to support autonomous navigation. This is true for both crewed and uncrewed vehicles. For crewed vehicles, autonomous navigation would allow the crew to safely navigate home in the event of a communication system failure. For uncrewed missions, autonomous navigation reduces the demand on ground-based infrastructure and could allow for more flexible operation. One promising technique for achieving these goals is through optical navigation. To this end, the present work considers how camera images of the Earth's surface could enable autonomous navigation of a satellite in low Earth orbit. Specifically, this study will investigate the use of coastlines and other natural land-water boundaries for navigation. Observed coastlines can be matched to a pre-existing coastline database in order to determine the location of the spacecraft. This paper examines how such measurements may be processed in an on-board extended Kalman filter (EKF) to provide completely autonomous estimates of the spacecraft state throughout the duration of the mission.

In addition, future work includes implementing this work on a CubeSat mission within the WVU Applied Space Exploration Lab (ASEL). The mission titled WVU Advanced Technology Satellite for Optical Navigation (WATSON) will provide students with an opportunity to experience the life cycle of a spacecraft from design through operation while hopefully meeting the primary and secondary goals defined for mission success. The spacecraft design process, although simplified by CubeSat standards, will be discussed in this thesis as well as the current results of laboratory testing with the CubeSat model in the ASEL.

Acknowledgments

I would first like to thank Dr. Christian for his continuous guidance and input regarding my research over the past few years. In addition, I would like to thank the members of the WVU Applied Space Exploration Lab for their willingness to help solve any problems I encountered. I especially thank Andrew Liounis and Dr. John Woods for their valuable feedback on my AIAA publication. Outside of the ASEL, I would like to acknowledge Dr. Jason Gross and Ryan Watson of WVU for their collaboration on the CubeSat Survey as well as Allison Willingham of NASA Goddard Space Flight Center for her support. I also thank the NASA West Virginia Space Grant Consortium (WVSGC) for their funding for this research through the 2014-2015 Research Initiation Grant. Finally, I would like to thank my parents for their constant love and support which has gotten me to where I am today.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Objectives	3
1.3	Thesis Organization	3
2	Background	4
2.1	Migration to Small Satellites	4
2.2	CubeSats	7
2.3	CubeSat Survey	10
2.3.1	Methodology	11
2.3.2	Results of Survey & Discussion	12
2.4	Flight Processors	14
2.5	Operating Systems	16
2.6	Discussion of Survey Results	16
3	CubeSats in the Applied Space Exploration Lab	17
3.1	Spacecraft Design	19
3.2	Current Progress of WATSON Mission	21
4	Methods of Optical Navigation	25
4.1	Optical Navigation	25
4.2	Navigation Using Landmarks	26
4.3	Coastline Determination	27
4.4	Coastline Matching with a Georeferenced Database	29
5	Extended Kalman Filter	33
5.1	Overview of the Kalman Filter	33
5.2	Derivation of Propagation Equations	35
5.3	Derivation of Measurement Step and Update Equations	39
5.3.1	State Update Equations	39
5.3.2	Covariance of the State Update Equations	41
5.3.3	Derivation of the Optimal Kalman Gain	42
6	Coastline Optical Navigation Filter Implementation	45
6.1	Spacecraft Mathematical Models	45
6.2	State Propagation using an Extended Kalman Filter	50
6.3	Measurement Model for Coastline Points in Images	51
7	LEO Simulation Results	53
7.1	Coastline Visibility	53
7.2	Assessment of Navigation Filter Performance	54
7.3	Discussion of Filter Results	59
8	Outreach Opportunities	59
9	Conclusion	61

10 Appendix	63
10.1 Outreach Presentation	63

List of Figures

1	Example of a microsatellite: FASTRAC developed by the University of Texas at Austin (University of Texas at Austin)	5
2	Example of a nanosatellite: WVU Advanced Technology Satellite for Optical Navigation (1U CubeSat)	6
3	Example of a femtosatellite: Sprite ChipSat developed by Cornell University (image: Cornell University)	7
4	Example of CubeSat Configurations (image: University of Colorado Boulder)	8
5	Poly PicoSatellite Orbital Deployer (P-POD) (image: CalPoly) . . .	9
6	6U Nanosatellite Launch Adapter System (NLAS) dispenser developed by NASA Ames (image: NASA Ames)	9
7	Results of survey showing the size and number of CubeSats launched per year as of August 2014 (based on 116 CubeSats with applicable data)	13
8	Results of survey showing the number of U.S. CubeSats launched per year, separated by class, as of August 2014 (based on 116 CubeSats with applicable data)	13
9	Comparison of CubeSat class percentages based on data from all information collected and the sample set in which processor information was known	14
10	Summary of processors used in surveyed CubeSat missions	15
11	Results of survey showing the processors used for non-university missions (Total: 14)	15
12	Results of survey showing the percentage of OS used in CubeSat missions (Total: 33)	16
13	WATSON concept of operations to meet secondary goals	20
14	Coordinate frame for CubeSat power analysis	22
15	Results for CubeSat power analysis	24
16	Current assemblies of the Interorbital Systems Arduino PCB (bottom left), solar panel (top), and power management PCB (bottom right)	24
17	Camera footprints (gray boxes) from a spacecraft in a 56° inclination, 1,000 km circular orbit with images taken once every 5 minutes. The camera has a FOV of 30°	32
18	Frame image (a) to be matched to template image (b) with data from GSHHG database	33
19	Normalized cross correlation results from template matching	33
20	Observed image frame found in template image based on normalized cross correlation values	34
21	Flow chart summarizing the extended Kalman filter algorithm. . . .	44
22	Visual representation of orbital elements	49
23	Coordinate frames used for spacecraft dynamics modeling	50
24	Line-of-sight vectors for measurement model	51
25	Simulated view of coastlines and the corresponding footprint based on a 1,000 km circular polar orbit	53

26	Orbit-specific contour plot showing the percentage of time coastline is observable based on inclination and FOV at a 1,000 km circular orbit with images taken once every five minutes	54
27	Monte Carlo position and velocity errors for a 1,000 km equatorial (0° inclination) orbit	56
28	Monte Carlo position and velocity errors for a 1,000 km, 45° inclination orbit	56
29	Monte Carlo position and velocity errors for a 1,000 km polar (90° inclination) orbit	57
30	Monte Carlo position and velocity errors for a 500 km, 45° inclination orbit	57
31	Monte Carlo position and velocity errors for a 500 km polar (90° inclination) orbit	58
32	Discussing what different types of engineers do on a daily basis with Mrs. Maloney's 7th grade science classes at Suncrest Middle School	60
33	Racing balloon rockets to demonstrate Newton's 3rd law of motion in Mrs. Maloney's 7th grade science classes at Suncrest Middle School	61

List of Tables

1	Small Satellite classifications as categorized by mass	5
2	Summary of Data Collected for CubeSat Survey	12
3	Processor Specifications	17
4	Proposed mass budget for the WATSON mission	22
5	Summary of steady state position and velocity errors (1σ) for specified orbits	55

Nomenclature

τ	integration variable
Ω	argument of periapsis
ω	right ascension of the ascending node
ν	true anomaly
μ	Earth's gravitational parameter
σ	standard deviation
a	semi-major axis
$\mathbf{e}_k$	state error
$\mathbf{e}_i$	unit vector in LOS direction
e	eccentricity of orbit
h	altitude of orbit
$\mathbf{h}$	specific angular momentum
i	inclination of orbit
m	mass
$\mathbf{o}_i$	vector from center of Earth to landmark
r	scalar norm of vector $\mathbf{r}$
$\mathbf{r}$	spacecraft position vector
$\dot{\mathbf{r}}$	spacecraft velocity vector
$\ddot{\mathbf{r}}$	spacecraft acceleration vector
$\mathbf{s}$	vector from spacecraft to landmark
$\mathbf{v}$	measurement noise
$\mathbf{w}$	zero-mean white noise vector
$\mathbf{x}$	state vector
$\mathbf{y}$	measurement vector
$\tilde{\mathbf{y}}$	noisy measurement vector
$\mathbf{A}$	rotation matrix from spacecraft PQW to IJK coordinate frame
$\mathbf{D}$	satellite degradation
E	expected value operator
$\mathbf{F}$	Jacobian of state vector
$\mathbf{H}$	measurement mapping matrix
$\mathbf{I}$	identity matrix
$\mathbf{I}_d$	inherent degradation of solar cells
$\mathbf{K}$	Kalman Gain
L	satellite lifetime
$\mathbf{P}$	covariance of state
P	power
$\mathbf{Q}$	process noise
$\mathbf{R}$	measurement covariance
$\mathbf{T}$	rotation matrix from camera to inertial frame
$+$	<i>a priori</i> (before update) estimate
$-$	<i>a posteriori</i> (after update) estimate

Acronyms

ASEL	Applied Space Exploration Lab
COTS	Commercial-Off-The-Shelf
EKF	Extended Kalman Filter
FOV	Field Of view
KF	Kalman filter
LEO	Low Earth Orbit
WATSON	WVU Advanced Technology Satellite for Optical Navigation
WVSGC	West Virginia Space Grant Consortium

1 Introduction

The purpose of this thesis is to perform an initial investigation of a small satellite mission to be developed in the Applied Space Exploration Lab (ASEL) at West Virginia University (WVU). Currently, this is planned to be one of the first satellites launched by WVU and therefore will be used widely as an educational tool to obtain experience with both mission and hardware design and development. Additionally, it will allow for navigation methods researched by students in the ASEL to be implemented and analyzed on an actual mission.

Within the ASEL, research is being conducted primarily in the fields of optical navigation. This research applies to both Earth orbits and Lunar orbits, rendezvous and docking of spacecraft, and even landing on asteroids. Optical navigation allows one to determine the position and orientation of a spacecraft relative to an object through information obtained by camera images. Some of these methods require manual operation which is not always ideal. For this reason, methods are being studied that will allow for autonomous navigation through optical methods.

For this thesis, the focus will be on designing and analyzing a CubeSat mission to implement an algorithm for autonomous navigation for an earth observing mission. This requires an understanding of the trend towards miniaturized satellites and the limitations this poses on the mission, as well as an understanding of navigation filters and their results. These topics will be discussed thoroughly in subsequent sections.

1.1 Motivation

When designing a spacecraft for any mission, it is important to consider the physical limitations of a spacecraft. In order to reduce the cost of a launch, the mass and volume of the spacecraft both must be minimized, in turn reducing the amount of available on-board power. This has caused a significant migration

towards the development of small satellites (SmallSats) which have led to the design and development of CubeSats, or small 10 cm x 10 cm x 10 cm satellites. Developed by California Institute of Technology in 1999, CubeSats have become standardized platforms allowing for increased ride-sharing launch opportunities, thus providing access to affordable educational tools for use in university settings.

Typically, CubeSats are launched into a low Earth orbit (LEO) with an altitude that can vary from approximately 160 km to 2,000 km; however a satellite in an orbit below 300 km will experience severe atmospheric drag and is often considered impractical. Depending on altitude and other orbital parameters, the lifespan of a CubeSat can vary greatly. The satellite's orbit is often based solely on the ride-sharing opportunity and launch vehicle available at the time of launch. On average, a CubeSat will remain in orbit for approximately 1 year before experiencing significant orbital degradation effects.

As far as navigation is concerned, with the many recent technological developments in the aerospace industry and the desire to conduct more ambitious space missions, there is an increasing need for fully autonomous spacecraft. Autonomous spacecraft navigation systems are important for both crewed and uncrewed missions. In the case of crewed missions, autonomous navigation systems are crucial in bringing the crew home safely in case of a communication system failure. For robotic missions, autonomous navigation reduces demands on ground-based infrastructure and increases mission flexibility when far away from the Earth. Recent advancements in technology have provided autonomous opportunities through the use of Global Position System (GPS) receivers, however highly accurate receivers can become costly and are still prone to error. For this reason, various forms of autonomous navigation have been studied thoroughly. In this case, we consider optical navigation using images of Earth. Using these images, it is possible to use identified coastlines to determine information about the spacecraft position, thus providing the motivation for this thesis.

1.2 Objectives

This thesis will provide a means of describing the process involved in designing a new mission to eventually be flown as a CubeSat mission from WVU. This process involves computing estimates of mass, power, communication link budgets, orbital maneuvers, etc. as well as creating 3D CAD models to estimate the volume available. Lab testing will also be done through the assembly of a CubeSat prototype in the lab.

In addition, MATLAB will be used to generate computer simulations that provide visual interpretations of what a satellite will see during various orbits. This will allow for visual line-of-sight vectors to be generated and used in a navigation filter to determine the location of the spacecraft.

1.3 Thesis Organization

This thesis is organized as follows: Chapter 2 provides an introduction to CubeSats and describes the recent trend towards miniaturization of spacecraft, with Chapter 3 detailing work with CubeSats to date in the WVU Applied Space Exploration Lab. Chapter 4 explains various methods of optical navigation and presents a proposed method that uses images of coastlines from a satellite camera in LEO to determine the location of a spacecraft. Chapter 5 presents a derivation of the state propagation equations and the measurement step and update equations used in an extended Kalman filter. Chapter 6 shows the mathematical models needed to use coastlines as landmarks for navigation in an extended Kalman filter where Chapter 7 analyzes filter performance from various LEO simulations. Chapter 8 provides results from outreach opportunities in which information about aerospace engineering was shared with local students. Chapter 9 will then provide a conclusion of this thesis.

2 Background

This chapter provides an overview of the trend towards small satellites as well as a description of CubeSats and the CubeSat standards developed by CalPoly. In addition, it provides the results from a CubeSat survey investigating the number and types of CubeSat missions launched from the United States. This survey concludes with presenting the most widely used processor and operating system for the surveyed CubeSat missions. This will help the ASEL determine what components to purchase for the future mission.

2.1 Migration to Small Satellites

The launch of Sputnik I by the Soviet Union in 1957 gave rise to what we now know in the United States as the Space Era.¹ Sputnik was the world's first artificial satellite and spurred a response from the U.S. which led directly to the creation of the National Aeronautics and Space Administration, or NASA, in 1958.¹ In that same year, the U.S. launched their first successful satellite known as Explorer I.² During this time period, satellites were often large and massive (which is often still the case to date). For example, Sputnik I weighed 83.6 kg while Explorer I weighed 13.9 kg, but it was considered a great achievement that something of such size and mass could be launched into orbit.¹

In recent years, technological advancements in electronics and sensor technology have allowed components to be made with smaller footprints and increased performance. In turn, it is now possible to miniaturize not only electrical components, but satellites as well. Missions that once required large and expensive satellite components are now much more affordable and able to be launched via various launch vehicles. In addition, many of these satellites are able to be fabricated much more easily than traditional satellites due to their simplified infrastructure and the availability of commercial off-the-shelf (COTS) components. These characteristics allow for a shorter development time as well as more accessible

launch opportunities. Such satellites have been appropriately categorized as small satellites, or “SmallSats”, and can be broken down into subcategories (dependent on mass) as described in Table 1.

Table 1: Small Satellite classifications as categorized by mass

Mass	Classification
10-100 kg	Microsatellites
1-10 kg	Nanosatellites
0.1-1 kg	Picosatellites
10-100 g	Femtosatellites

Within the microsatellite category (10-100 kg) is the University of Texas at Austin’s Formation Autonomy Satellite with Thrust ReNav Attitude and Crosslink, or FASTRAC.³ This satellite was the winner of the Air Force’s University Nanosatellite Program (UNP) Nanosatellite-3 competition in 2005. Since 1999, the UNP has provided university students the opportunity to take part in a four year concept-to-flight-ready spacecraft design and development competition to enable small satellite research and development.⁴ Upon the design review conducted by top members of the aerospace community, one satellite is chosen for funding to complete and launch the proposed mission. FASTRAC was the winner of the competition in 2005 and was UT Austin’s first student-built satellite. Launched in November 2010, the purpose of this mission was to investigate relative navigation solutions using crosslinked data measurements between the two satellites.³

Figure 1: Example of a microsatellite: FASTRAC developed by the University of Texas at Austin (University of Texas at Austin)

It should be noted that within the category of nanosatellites (1-10 kg) are CubeSats, which will later become the focus of this thesis. Briefly, CubeSats are small 10 cm x 10 cm x 10 cm cubes that have a maximum mass of 1.3 kg. Alternate configurations exist, such as the 3U CubeSat, where the satellites can be stacked along one axis with maximum dimensions of 30 cm x 10 cm x 10 cm and a maximum mass of 10 kg. A more elaborate discussion of CubeSats will be provided in later sections.

Figure 2: Example of a nanosatellite: WVU Advanced Technology Satellite for Optical Navigation (1U CubeSat)

One example of a picosatellite (0.1-1 kg) is the PICOSAT1.0 developed by the Aerospace Corporation. This mission, launched in February of 2000, consisted of two tethered picosatellites in order to demonstrate formation flying of tethered picosatellites in addition to successfully communicating results from testing of MEMS (microelectromechanical system) components to the ground station. The two PICOSATs weighed approximately 0.275 kg each and had dimensions of 25 mm x 75 mm x 100 mm.^{5,6}

Finally, an example of a femtosatellite (10-100 g, also known as a satellite on a chip) is given by the KickSat mission developed at Cornell University. The KickSat mission plans to demonstrate the deployment of 128 Sprites (see Figure 3) from a CubeSat-based mechanism while monitoring their performance in space. The Sprite ChipSat has been in development since 2008, providing an excellent example of a femtosatellite. Typically, these satellites include the most essential features of

a traditional satellite scaled down to fit on a single chip. More information about this mission can be found in [7] in addition to their open source software used for the mission.

Figure 3: Example of a femtosatellite: Sprite ChipSat developed by Cornell University (image: Cornell University)

2.2 CubeSats

As mentioned previously, within the nanosatellite category of small satellites is the increasingly popular CubeSat. The CubeSat Project, developed in 1999 by Professor Jordi Puig-Suari of California Polytechnic State University (Cal Poly), San Luis Obispo and Professor Bob Twiggs of Stanford University's Space Systems Development Laboratory (SSDL), was an initial attempt at standardizing design parameters for small satellites in order to reduce cost and development time for missions. The CubeSat Project outlines the standards required for CubeSat development while describing specific parameters (i.e. dimensions, mass, deployment mechanisms, etc.) that must be met for the spacecraft to qualify as a CubeSat.

The CubeSat Project states that a 1U (1 unit) CubeSat must have the dimensions of 10 cm x 10 cm x 10 cm. Alternate configurations exist and include 2U and 3U CubeSats with dimensions of 10 cm x 10 cm x 20 cm and 10 cm x 10 cm x 30 cm, respectively (see Figure 4). In addition, mass limitations prohibit CubeSats from exceeding 1.3 kg per unit (U), meaning a 2U CubeSat must not exceed 2.6 kg while a 3U must not exceed 4.0 kg. These standards also limit the

amount of power available for use on the spacecraft. The specified dimensions pose limitations regarding the surface area available for solar panels. Previous studies have shown that the average power of a CubeSat can vary from 1W (1U) to 5W (3U).⁸ However, assuming the spacecraft still meets the mass requirements, deployable solar panels can be attached to significantly increase the amount of available power (i.e. a 3U CubeSat with deployable panels can have up to 20W of available power).⁸

Figure 4: Example of CubeSat Configurations (image: University of Colorado Boulder)

CubeSats that meet dimension and mass requirements can be launched from the Poly Picosatellite Orbital Deployer (P-POD) developed by CalPoly (see Figure 5). The P-POD serves as a standardized deployment system and an interface between CubeSats and the launch vehicle for up to three 1U CubeSats or any equivalent combination. The P-POD is a rectangular aluminum box that contains a spring mechanism and a door. Most CubeSats require a deployment switch in which a signal is sent from the launch vehicle to the switch that releases a set of springs on the door to deploy the satellites. Any CubeSat being deployed from a P-POD must adhere to additional specific guidelines in the CubeSat Design Specifications document in order to ensure compatibility for a successful deployment.⁹

It should be noted that some recently proposed experiments have been designed for 6U CubeSat structures, resulting in the need for an alternative deployer. For-

tunately, NASA Wallops, NASA Ames, as well as other institutions have come up with designs for 6U deployers, all of which have been tested and have received a Test Readiness Level (TRL) of at least 6, with TRL 9 being that the object has been flown and tested successfully in space.¹⁰ Figure 6 shows the Nanosatellite Launch Adapter System (NLAS) developed by NASA Ames.

Figure 5: Poly PicoSatellite Orbital Deployer (P-POD) (image: CalPoly)

Figure 6: 6U Nanosatellite Launch Adapter System (NLAS) dispenser developed by NASA Ames (image: NASA Ames)

Pumpkin, Inc. provides a relatively inexpensive and comprehensive CubeSat kit available in 1U, 1.5U, 2U, and 3U configurations. All components included in this kit conform to the CubeSat specifications. Most components can be purchased (specifically from Clyde Space) that have undergone thermal and radia-

tion hardened testing.¹¹ The Pumpkin CubeSat kit includes the following:¹² A CubeSat structure of the specified size, a Development Board (DB) and Pluggable Socketed Processor Module (PSPM) for in-lab development and testing, a Motherboard (MB) and Pluggable Processor Module (PPM) for the Flight Model, low-power, high-performance electronics based on your choice of PPM (choose between TI's 16-bit MSP430 ultra-low-power microcontroller, Silicon Labs® C8051 8-bit mixed-signal MCU, Microchip® PIC24 16-bit high-performance microcontroller, or Microchip® dsPIC33 16-bit digital signal controller), Pumpkin's Salvo Pro RTOS software and CubeSat Kit software libraries, Pumpkin's processor-specific software libraries, HCC-Embedded's EDFS-THIN FAT file system library for use with the CubeSat Kit's SD card socket, example projects, plug-in modem/transceiver support and built-in USB 2.0, USB debugging/Flash emulation tool (FET) for programming and debugging, power supplies, programming adapters, cables and tools. These and other related kits make it easy and affordable to launch a satellite while providing access to space and educational experiences to a large audience.

Based on the size and task of a CubeSat mission, the cost of missions can range from approximately \$10,000 for a simple 1U mission to over \$100,000 for a 3U mission. This significant price increase is often due to the high cost of solar panels and the amount necessary to encompass the structure.¹³ The cost of launches, however, can vary greatly depending on the launch service. Some sources estimate that 1U CubeSat launches cost an average of \$85,000 [14] while others estimate \$40,000 per kilogram [15]. To help alleviate these costs, NASA developed the CubeSat Launch Initiative (CSLI) to provide free launch opportunities for CubeSats to be flown as secondary payloads on previously planned missions.¹⁶

2.3 CubeSat Survey

As briefly mentioned before, the standards developed by Cal Poly have resulted in a significant increase in the number of small satellite missions launched over the

past few years.¹⁷ This is partially due to the development of commercial off-the-shelf (COTS) components specifically designed to meet all CubeSat requirements while providing a variety of options readily available for various processors, operating systems, attitude control systems, navigation systems, etc. Because of this, CubeSats have become relatively inexpensive, making them ideal educational tools in university and classroom settings. This section surveys nearly all CubeSat missions to date and provides insight into the types of missions being carried out (i.e. university, government, private, etc.) as well as commonly used hardware and software and whether the mission was successful. Specifically, processors and operating systems were researched to determine which combinations were most reliable and how they affected mission results. It is hoped that this information will contribute to the success of WVU's first CubeSat mission in the near future.

2.3.1 Methodology

A comprehensive survey was conducted on past, present, and future CubeSat missions. This survey initially consisted of every CubeSat mission that provided accessible information, totaling slightly over 250 missions. For simplicity, it was decided that the focus would be on domestic (launched from within the United States) missions, narrowing the search to approximately 150 missions. Some of these missions had been previously launched, yet others were still proposed concepts. This list was consolidated once more to focus on already launched missions as well as missions with launch services confirmed in upcoming years. As a result, a comprehensive list of approximately 115 CubeSat missions has been developed.

To gather applicable information, the first step involved creating a list of desired mission details. This information can be categorized as listed in Table 2. A first pass of obtaining data was accomplished by thoroughly searching the internet, however a significant amount of data was ultimately left incomplete. As a second pass, the project managers of each mission were emailed and asked specifically for information regarding data listed in Table 2 as well as any other applicable

Table 2: Summary of Data Collected for CubeSat Survey

Mission Information	Launch Information	Other Information
Name of Satellite	Launch Date	Processor
Mission Objective	Launch Vehicle/Mission	Orbit
Sponsor	Mission Status	Uplink/Downlink
Manufacturer	Deployer	Attitude Determination
Domestic	Program	Attitude Control
Size	Orbit Parameters	Orbit Determination
Mass		Miscellaneous

information that could be provided or further references. This helped fill in a significant amount of missing data which was used to create a spreadsheet that can be provided electronically upon request and includes all applicable data collected.

It should be noted that despite all efforts made in this project, the results of this survey only provide estimates used to identify trends in the presented data. This is due to the fact that many of the surveyed missions were conducted by universities with limited budgets while others were sponsored by private companies or government agencies with higher budgets. This should be considered when analyzing the numerical data obtained from this survey and will hopefully be made prominent throughout the discussion of results.

2.3.2 Results of Survey & Discussion

A market assessment conducted by SpaceWorks Enterprises, Inc. in 2014 showed that of the nanosatellites launched between 2009 and 2013, 47% can be categorized as 1U CubeSats.¹⁷ Although not enough information is given to determine exactly how many 2U and 3U CubeSats have been launched, our data seems to show a significantly increasing trend towards 3U CubeSat missions based on the information presented in Figure 7. These results show the increase in overall CubeSat missions and gives reason to believe this trend will continue in the near future. There also seems to be an increase in 3U missions and a decrease in 1U missions based on this data set.

Figure 8 shows the total number of U.S. CubeSats launched per year, sep-

Figure 7: Results of survey showing the size and number of CubeSats launched per year as of August 2014 (based on 116 CubeSats with applicable data)

arated by class (i.e. private, government, university, military, etc.). From this graph, it is evident that a majority of launches have been for university missions and educational purposes; however there has been a significant increase seen in government missions as well as privately funded missions.

Since the applicable processor data does not encapsulate the total CubeSat

Figure 8: Results of survey showing the number of U.S. CubeSats launched per year, separated by class, as of August 2014 (based on 116 CubeSats with applicable data)

data set, it is important to point out that data may be skewed. Due to the educational nature of university missions, these project managers were often willing to provide mission information to other organizations or this information was made freely available online. However, military and government missions typically have some restrictions as to what information they can share with the public. For this reason, it was assumed that the sample data set (missions providing processor and operating system information) may be slightly biased by representing mainly university missions. However, upon further investigation, it was found that the sampled data set was similar to the total data set as represented in Figure 9.

Figure 9: Comparison of CubeSat class percentages based on data from all information collected and the sample set in which processor information was known

2.4 Flight Processors

Based on the 55 missions in which processor data was attainable, almost 40% of these missions utilized PIC processors. Of these processors, 52% used the PIC18. These results are summarized in Figure 10. Upon further investigation, it was concluded that 90% of the PIC18 processors were used in 1U missions. It is also important to note that Pumpkin, Inc. currently provides Pluggable Processor Modules (PPM) available with their CubeSat Kit, making it possible to incorporate the following processors into the flight module: SiLabs C805, PIC24, PIC33, and TI MSP430. The convenience of purchasing processors with a Pumpkin CubeSat Kit may play a key role in processor trends, especially for 1U Cube-

Sats, although many factors were not accounted for in this analysis (i.e. budget, development time, cost of materials, etc.) which could have played a key role in determining the processors as well.

Figure 10: Summary of processors used in surveyed CubeSat missions

To counter the potential skewness presented by the large percentage of university missions when considering processors, Figure 11 shows the percentage of processors used on all non-university missions. However, the results were once again consistent such that PIC processors totaled over 50% of all processor data recorded.

Figure 11: Results of survey showing the processors used for non-university missions (Total: 14)

2.5 Operating Systems

Salvo and Linux were the two most commonly used operating systems in the CubeSat missions surveyed. Salvo is available with the purchase of any Pumpkin PPM and is included with the purchase of a Pumpkin CubeSat Kit. Most processors provide various options for programming and therefore no concrete conclusion can be made as to which operating system should be used.

Figure 12: Results of survey showing the percentage of OS used in CubeSat missions (Total: 33)

2.6 Discussion of Survey Results

Based on the information collected from this survey, it can be concluded that a majority of CubeSat missions have used PIC processors (specifically PIC18) with either Linux or Salvo operating systems. The availability of Pumpkin CubeSat kits is likely a major contributor to these results, although this was not confirmed in this survey. Some crucial aspects to consider when choosing a processor are cost, user interface, and power requirements, each of which will be determined by the nature of the mission. It is worth noting that the results of this survey lead us to believe that successful missions can be achieved through the use of PIC processors and Linux or Salvo operating systems, however a variety of other

Table 3: Processor Specifications

Processor	Max. Operating Frequency (MHz)	Max. Program Memory (KB)	Clock Speed
PIC18	40	32	32 kHz
PIC24	32	256	32 kHz
PIC32	50	256	32 kHz
PIC33	16	536	32 kHz
TI MPS430	24	512	32 kHz
SiLabs 8051	50	32	16 MHz
Freescale HC12	50	512	32 kHz
Atmel AtxMega (Arduino)	32	128	32 kHz
Atmel AT91SAM9 (Arduino)	210	8	32 kHz
Phytec phyCORE- LPC3250	208	32	13 MHz
Taskit Stamp9G20	396	32	32 kHz

options are available and may be better suited to certain missions. Table 3 provides specifications on some of the processors used for missions in this survey. Although different PIC models are comparable to each other, it can be seen that significantly faster and more powerful processors do exist and have been tested and flown in space.

3 CubeSats in the Applied Space Exploration Lab

As of 2015, West Virginia University has not yet launched a CubeSat mission; however, a 1U mission is currently being designed and developed by the Applied Space Exploration Lab and is known as the WVU Advanced Technology Satellite for Optical Navigation, or WATSON. This mission is being developed with primary

goals (those required to achieve mission success) consisting of the following:

1. To develop a university built CubeSat program at WVU,
2. To provide undergraduate and graduate students with hands-on experience in designing, fabricating, and operating a spacecraft, and
3. To enhance the education of students at WVU to benefit both the state of West Virginia and the nation.

These primary goals of the WATSON project are supported by a number of goals identified in the NASA Strategic Plan,¹⁸ which are summarized below.

- **NASA Strategic Goal 2:** Expand scientific understanding of the Earth and the universe in which we live
 - *Outcome 2.1:* Advance Earth system science to meet the challenges of climate and environmental change
- **NASA Strategic Goal 3:** Create the innovative new space technologies for our exploration, science, and economic future
 - *Outcome 3.1:* Sponsor early stage innovation in space technologies in order to improve the future capabilities of NASA, other government agencies, and the aerospace industry
 - *Outcome 3.4:* Facilitate the transfer of NASA technology and engage in partnerships with other government agencies, industry, and international entities to generate U.S. commercial activity and other public benefits
- **NASA Strategic Goal 6:** Share NASA with the public, educators, and students to provide opportunities to participate in our mission, foster innovation, and contribute to a strong national economy
 - *Outcome 6.1:* Improve retention of students in STEM disciplines by providing opportunities and activities along the education pipeline

Secondary goals, although desirable, are often considered “stretch” goals and are not necessarily required for mission success, but would be beneficial towards future endeavors if met. For WATSON, the secondary goal is to capture images of Earth from LEO in support of on-going optical navigation research at WVU. These

images would be used for image processing in which coastlines are identified and matched to some georeferenced database for implementation with an extended Kalman filter (discussed in Chapters 4-6) to autonomously estimate the state and covariance of the spacecraft. Accomplishing these secondary goals means that WATSON could contribute to two technical areas (TA) of the NASA Space Technology Roadmap:¹⁹

- **TA05:** Communication and Navigation Systems
 - Reduce mass and power requirements, advanced autonomous control
- **TA08:** Science Instruments, Observatories, and Sensor Systems
 - Remote sensing, environmental conditions

3.1 Spacecraft Design

One of the first steps in the design process of a spacecraft consists of providing a concept of operations, or ConOps. A ConOps is essentially a description of the overall mission and its operations from launch to deorbit. Figure 13 shows the primary concept of operations of the WATSON mission. In this case, after the satellite is launched into orbit via some launch vehicle, it will then be deployed from a P-POD. After reaching the desired orbit, communication will be established with the ground. Images will then be captured by the on-board camera and downlinked to the ground for processing. The mission is planned to last approximately 6 months before the spacecraft is subject to atmospheric drag effects causing it to deorbit and burn up on reentry to Earth's atmosphere.

When designing a spacecraft mission, there are several factors that must be considered such as mass, power, cost, schedule, lifetime, and reliability. Specifically, mass is a fundamental driver in the spacecraft design process. This directly influences cost because of the effect of mass on the launch vehicle; the larger the mass of the spacecraft, the more energy it will take to launch into orbit. Power also plays a key role in the design of a spacecraft. In addition, schedule can ultimately

Figure 13: WATSON concept of operations to meet secondary goals

affect the design of a spacecraft. For example, more relaxed schedules could allow the designers ample time to optimize the design and consider other options as opposed to a shorter time line that allows for only a few designs. Finally, lifetime and reliability can drive the design process of a spacecraft in the sense that longer missions will likely require more durable and radiation-hardened components or parts.

It is also important to consider the orbit and propulsion requirements for a mission. The orbit will affect the amount of sunlight available to the solar panels and can require various thermal design constraints. If the spacecraft is to maneuver to a different orbit, a change in velocity (ΔV) budget will need to be designed. This will also affect the propulsion system and the mass of the spacecraft.

Ultimately, it is the payload that will drive a majority of the design of the spacecraft. The payload will determine what orbit the spacecraft must be in to obtain applicable data, which in turn determines whether propulsion will be necessary. The payload also determines how much power will be needed to operate the components.²⁰

As mentioned before, CubeSats have greatly simplified the spacecraft design process. The size and mass limitations make it feasible for CubeSats to be launched via rideshare opportunities, often at little or no cost to the designer. Typically, the launch vehicle has a predetermined orbit (often LEO) which limits the orbit determination portion of typical spacecraft design procedures. CubeSats in LEO will most likely not have a lifetime over one year and will be subject to orbital degradation, eliminating the need for recovering the spacecraft. Because of this, it is common to use relatively inexpensive COTS components produced by various manufacturers such as Pumpkin, Inc. or Clyde Space. These components are designed to meet CubeSat standards, therefore alleviating the need to design, assemble, and test new PCBs and other electrical components.

3.2 Current Progress of WATSON Mission

To design a successful mission, it is necessary to consider constraints that may limit the size or functionality of a spacecraft. In the case of a CubeSat mission, it is crucial to ensure that the satellite meets the mass limitations imposed by the CubeSat standards. Since WATSON is proposed to be a 1U CubeSat, it must weigh less than 1.3 kg to be deployed by a P-POD. Because of this, a mass budget was created (as seen in Table 4) where CBE represents the current best estimate. A 15% contingency is added since these are preliminary mass estimates, but this contingency percentage will decrease as the project matures. This budget accounts for a payload (camera) and the spacecraft subsystems. These subsystems include an attitude determination and control system (ADCS), communications, command and data handling (C&DH), power, and structure. The values for the subsystems were obtained from data sheets provided by the Pumpkin CubeSat Kit manufacturers while the payload is based on an Edmund Optics camera and lens. It can be seen that there is a margin of approximately 23 g.

Currently, testing in the ASEL is being carried out with an Interorbital Systems 1U CubeSat kit donated to the ASEL from NASA IV&V (located in Fairmont,

Table 4: Proposed mass budget for the WATSON mission

Element	CBE (g)	15% Contingency (g)	Allocated (g)	Total (g)
1.0 Payload	200.0	30.0	230.0	230.0
2.0 Spacecraft Bus (dry)	650.0	97.5	747.5	747.5
2.1 ADCS	100.0	15.0	115.0	
2.2 Communications	100.0	15.0	115.0	
2.3 C&DH	100.0	15.0	115.0	
2.4 Power	100.0	15.0	115.0	
2.5 Structure	200.0	30.0	230.0	
3.0 Margin				22.5
4.0 Total Allocated CubeSat Mass				1,000.0

WV). Since the kit was found to be incompatible with a P-POD, it is being used primarily as an educational tool for students. Part of this learning experience involves assembling and integrating various subsystems, specifically the solar panels, power distribution system, and the command and data handling system.

Figure 14: Coordinate frame for CubeSat power analysis

An initial power analysis was performed to determine how much power to expect from a 1U CubeSat with solar panels on four of the six faces as specified by the Interorbital Systems design. Using the geometry shown in Figure 14, the orientation of the CubeSat with respect to the sun could be used to calculate the amount of power available from the solar panels. This was done by approximating the beginning-of-life power (P_{BOL}) available per unit area for each solar panel

using Equation 1 where P_o is the power output from the solar cells as specified by the provider (usually in terms of power per area), I_d is the inherent degradation dependent on the composition of solar cell materials, and $\cos \theta$ is the loss when the panel is not positioned perpendicularly to the sun.²⁰

$$P_{BOL} = P_o I_d \cos \theta \quad (1)$$

It is also important to take into consideration the overall degradation during the mission lifecycle. We can describe this as follows, where D is the degradation per year and L is the satellite lifetime in years.²⁰

$$L_d = (1 - D)^L \quad (2)$$

When calculating the amount of available power, it is crucial to ensure that end-of-life power is used in calculations, otherwise due to degradation effects, there may not be enough available power at the end of the mission to reach completion.²⁰ For this reason, P_{EOL} is calculated and used for the final results.

$$P_{EOL} = P_{BOL} L_d \quad (3)$$

These results, shown in Figure 15, show that we can expect approximately 0.5W-0.8W of power from the solar panels provided by the Interorbital Systems CubeSat kit. This is consistent with 1U CubeSat power recommendations (utilizing six solar panels) for the literature in [8].

Currently, the CubeSat is still being assembled. Figure 16 shows an assembled solar panel, the Arduino PCB, and the power management PCB provided with the Interorbital Systems CubeSat kit. Testing in the lab will be done to ensure the solar panels are able to recharge the battery pack through programming of an Arduino Mini in the near future.

Figure 15: Results for CubeSat power analysis

Figure 16: Current assemblies of the Interorbital Systems Arduino PCB (bottom left), solar panel (top), and power management PCB (bottom right)

4 Methods of Optical Navigation

This chapter will introduce optical navigation and some commonly used methods, while later describing navigation using landmarks and coastline determination for Earth observing satellites using coastlines as landmarks. Simulation results for a model depicting a satellite orbiting Earth will be provided in addition to a detailed description of how this simulation incorporates a coastline database into the filter used for spacecraft navigation.

4.1 Optical Navigation

The first mission to incorporate optical navigation (OPNAV) methods in orbit determination was the Mariner 9 mission to Mars, launched in 1971.²¹ In this case, as was the case for many missions in this time period, optical measurements were obtained from an image of a known object or target body. Typically, target bodies include planets, moons, asteroids, comets, and stars. In general, OPNAV often consists of two different problems: 1) determining where the image of the target body should appear if the position of the spacecraft, the position of the target, and the camera's attitude and optical parameters are known and 2) determining the coordinates of targets in a digital image, often found using image processing methods and image matching techniques.²² These coordinates are then used to estimate the state of the spacecraft. In the case of the Mariner 9 mission, any measurements had to be processed on the ground before the updated estimate could be uplinked back to the spacecraft. This was not only time consuming, but also left no room for error in the case of a communications system failure. To alleviate this issue, various types of on-board optical navigation methods have been designed and tested which can also be used for deep space missions or as backup navigation methods in case of a crewed mission system failure.

At long ranges, it is common to use horizon-based OPNAV techniques, such as those described in [23, 24, 25]. Christian and Lightsey in [23] show how two types

of horizon-based OPNAV measurements are used: centroid-apparent diameter and angle between a horizon and reference star measurements. The centroid-apparent diameter method is fairly straightforward and is used if a significant portion of a target body can be seen in the image. Assuming this is the case, then the direction to the center of the body can be determined with image processing techniques. If the physical dimensions of the target body are known, these values can be used relative to the size of the body in the image to estimate the distance between the spacecraft and the body. The second method involves using the angle between the horizon of the body and a reference star to determine the line-of-sight (LOS) vector. Although this measurement does not produce a fixed position vector, it will instead create a cone of possible locations of the spacecraft relative to the target body. However, using additional star locations can provide a line in space along which the spacecraft must lie.²³

4.2 Navigation Using Landmarks

At close ranges, surface features or landmarks may be observed on the target bodies. Many studies have been carried out concerning the use of optical navigation with various types of surface landmarks.^{26,27,28,29,30,31} For example, craters commonly found on the surface of planets, satellites, asteroids, and other solar system bodies are often proposed as landmarks for optical navigation. Generally, a crater can be identified in an image by looking for an elliptical rim and a bright to dark shading pattern depending on the lighting at the time of observation. Because of these commonalities, most crater detection algorithms consist of a series of steps that include edge detection and ellipse fitting. Once the ellipses are identified as landmarks, the size, shape, and position of that landmark relative to other landmarks are used to match the ellipses with a known database of craters for that body.²⁸ It should be noted, however, that this process can be difficult because the craters often have different appearances when viewed from different directions and with different sun angles. Also, the age of the crater may contribute to the sharp-

ness of the crater edges on the surface which can make detection easier or more difficult.²⁶ Techniques such as image cross-correlation, context based matching, and projective conic invariants can be used to make up for these differences but are beyond the scope of this paper. They are discussed in detail in [26] and [30].

On Earth, there are a number of different naturally occurring features that could be used for navigation. Of special interest here are coastlines, islands, and other inland water boundaries such as lakes and rivers due to the sharp contrast able to be observed by a satellite in low-Earth orbit (LEO).³² Coastlines are used due to the visible contrast in colors between land and water as seen by the spacecraft. These bodies of water can include coastlines, islands, rivers, and lakes. Additional features can be detected as well (e.g. volcanoes, mountains, snow, and even urban areas); however these are often changing over time and can be more difficult to distinguish and map.³³

By using coastlines as landmarks, a line-of-sight (LOS) vector can be used in an extended Kalman filter (EKF) to autonomously estimate the state and the state covariance of a spacecraft. To do this, it is assumed that the only source of external measurements will be from a star-tracker (for attitude) and from an Earth-observing camera. The state of the spacecraft will then be propagated on-board using assumed dynamics and Inertial Measurement Unit (IMU) data. In addition, a camera measurement model was developed to utilize the measurements obtained from the landmark observations for processing through the on-board EKF. These models will be discussed in later chapters.

4.3 Coastline Determination

Numerous methods of extracting coastlines from images have been studied. Common solutions include using conventional image processing boundary detection methods or classifying illumination levels of an image. Generally, edge or boundary detection refers to being able to identify sharp discontinuities in an image and determining whether they represent boundaries between regions of the

image. Typically, these discontinuities are found as abrupt changes in pixel intensity. Various edge detection methods are available such as the Roberts, Sobel, or Prewitt edge detectors which are based on first order derivative (or gradient) operators.³⁴ Others, such as the Laplacian of a Gaussian (LOG) are based on second order derivative operators.³⁴ First order derivative edge detectors function by taking the first derivative of the image and observing the maximum and minimum intensity points. A pixel is declared an edge point if the value of the gradient exceeds some specified threshold. The Laplacian method takes the second derivative of an image and searches for zero crossings which it then labels as an edge point.^{35,36}

One especially popular choice for edge detection is the Canny edge detector.³⁷ This algorithm was created by John Canny as a way to enhance the many edge detection methods already known. Specifically, he wanted to minimize the amount of “missed” edge points while identifying non-edge points in addition to minimizing the distance between the detected edge points and the actual edge. His final criterion was to only have one response to each edge point. Based on these criteria, Canny implemented a detection algorithm that begins by smoothing the image to eliminate noise. The image gradient, or the directional change in the intensity of the image, is then found and used to separate the image into regions. The algorithm then suppresses pixels within these regions that are not maximums. The gradient is further reduced by hysteresis using two threshold values. If the magnitude is above the threshold, it is considered an edge and if it is below the threshold, it is set to zero and not labeled as an edge. If the magnitude is between the two thresholds, it is set to zero unless there is a surrounding pixel above the higher threshold.³⁵

Although it is considered an optimal edge detector, previous work by Liu and Jezek³² has shown Canny edge detection to be insufficient for coastline extraction which will be necessary for a successful WATSON mission. This is due to the lack of consistent intensity contrast between land and water regions, often resulting

in discontinuous coastline data. However, it is possible to alleviate this problem using image segmentation and post-segmentation processing as seen in [32]. Other authors, such as those in [38], use two masking steps in addition to the Canny edge detection: an edge focusing step and a closing step used as an input for the object-oriented matching process. The usage of these masking steps in [38] showed significant improvements over Canny edge detection.

One other problem that arises with coastline extraction and feature selection is the potential for misidentification of non-coastline boundaries as coastlines. For example, many edge detection techniques use contrast as a determining factor to differentiate boundaries and edges. If the satellite image contains cloud coverage, it is possible for clouds to be considered boundaries and therefore identified as a possible coastline. A solution to this problem is discussed in [39] using a global ocean color sensor which measures radiances in eight visible and near-infrared bands. By creating scatter plots of radiance observed from two different bands, it was possible to normalize the data and specify thresholds to classify observed regions as land, water, clouds, or ice.

4.4 Coastline Matching with a Georeferenced Database

After coastlines are determined from an image, it is necessary to match observed coastline points to some coastline database. This process is called image registration. In general, there are two main categories of image registration: manual registration and automated registration. Manual registration often uses manually chosen ground control points (GCPs) or recognizable features in an image. The process of choosing these points in both the sensed image as well as the reference image is both time consuming and inaccurate. Automated registration techniques can also be divided into two categories: area-based and feature-based. Area-based methods compare a small window from the reference image with the same size window in the sensed image and the normalized cross correlation between the two are compared. This method is relatively straightforward, but only

works well for gray scale images. Feature-based methods extract and match either region boundaries or sharp edges from the two images. Eugenio, et al., propose in [40] a feature-based approach that consists of estimating an image gradient energy map of the sensed image. Corresponding features were found by optimizing reference image contours on the sensed image which is then re-sampled using the coefficients of the energy mapping functions.⁴⁰

When dealing with a digital image, the image can be thought of as a matrix containing light intensity values for each pixel. At times it can be helpful to adjust these intensity values in relation to the intensity values of neighboring pixels, a process known as filtering. After the image is filtered, the new intensity of each pixel is determined using a weighted sum of the surrounding pixels. This weighting is defined by a second matrix, $\mathbf{G}$, frequently called the "kernel". The process of calculating the weighted sum of the pixels is called cross-correlation.⁴¹ Mathematically, this is described by

$$\mathbf{J}(u, v) = \sum_{i=-n}^n \sum_{j=-m}^m \mathbf{G}(i + n + 1, j + m + 1) \mathbf{I}(u + i, v + j) \quad (4)$$

where $\mathbf{J}$ is the filtered digital image, (u, v) represents the image coordinate of the pixel, and $2n + 1$ and $2m + 1$ are the number of rows and columns of $\mathbf{G}$, respectively. This can be applied to all pixels using

$$\mathbf{J} = \mathbf{G} \star \mathbf{I} \quad (5)$$

where $\star$ represents the cross correlation operator.

Cross correlation can also be thought of in terms of a dot product. If the columns of $\mathbf{G}$ are stacked into a single column vector $\mathbf{g}$ and the columns of the image patch (i.e. the small area to be matched within larger template image) are stacked to form column vector $\mathbf{p}$, then the cross correlation can be calculated by

$$\mathbf{J}(u, v) = \mathbf{g}^T \mathbf{p} \quad (6)$$

When $\mathbf{g}$ and $\mathbf{p}$ are in the same direction (i.e the patch is similar to the template), the dot product will be a maximum. This means that a filter may be used as a template. Typically, to prevent mis-registration due to varying brightness and intensity values within the image, the image patch vector may be normalized at each pixel using

$$\mathbf{J}'(u, v) = \frac{\mathbf{g}^T \mathbf{p}}{\|\mathbf{p}\|} \quad (7)$$

This process is called normalized cross correlation. It is important to note that values of the kernel are usually normalized to sum to 1 to ensure the correlation process does not brighten or dim the image. It is this normalization that allows this operation to be more robust than others, however it becomes computationally expensive as the coefficient is calculated for each point in the image.⁴² In addition, it should be noted that this algorithm is not scale and rotation invariant, meaning results are only feasible if the observed image frame is able to be converted to the same scale and rotation as the template image. It is assumed that data obtained from a startracker and IMU on-board the satellite will allow for determination of appropriate rotations to transform the captured image to the same coordinate frame as the template image.

Several options are available to provide a reference of coastlines for optical navigation. One of the most well-known databases is provided by the National Geophysical Data Center (NGDC), part of the National Oceanic and Atmospheric Administration (NOAA). World shorelines are provided through the Global Self-Consistent, Hierarchical, High-resolution Geography database (GSHHG) in the form of shapefiles that store geometrical characteristics of datasets as vector coordinates, i.e. coastlines and other water/land boundaries.⁴³ These files are able to be read and written using a variety of programs. The NGDC provides a software called GEODAS-NG that has the capability of reading the coastline data from the GSHHG or writing shapefiles from a provided input image.

It is also possible to use MATLAB as a georeferenced database in conjunction with the GSHHG database. MATLAB is capable of plotting GSHHG data with

respect to latitude and longitude coordinates. Using this data, it was possible to determine both a model of the Earth in order to simulate the coastlines that would be observed from a satellite in LEO, as well as correlation results using normalized cross correlation. The first simulation was modeled under the assumption that the on-board camera is always nadir-pointing (with a small amount of random noise added). With this assumption, the spacecraft orbit was propagated using two-body orbital mechanics.⁴⁴ Figure 17 provides a visual representation of a spacecraft in a 1,000 km orbit with a 56° inclination. The solid line represents the ground track of the spacecraft while the gray boxes show the area visible in the camera FOV assuming an image is taken once every 5 minutes.

Figure 17: Camera footprints (gray boxes) from a spacecraft in a 56° inclination, 1,000 km circular orbit with images taken once every 5 minutes. The camera has a FOV of 30° .

Results from the previously mentioned correlation algorithms were implemented using North America coastlines as the template image while trying to locate the Great Lakes. The frame and template images generated using high resolution GSHHG data can be seen in Figure 18. The results showing the correlation coefficients can be seen in Figure 19 where (a) shows the correlation results within the template image and (b) shows a more detailed view of the maximum correlation values. Figure 20 shows the observed image located within the template based on the highest calculated correlation coefficients.

(a) Observed image frame after satellite image edge detection

(b) GSHHG high resolution database template

Figure 18: Frame image (a) to be matched to template image (b) with data from GSHHG database

(a) Results from normalized cross correlation of high resolution GSHHG data template matching

(b) Detailed view of correlation results using high resolution GSHHG data

Figure 19: Normalized cross correlation results from template matching

5 Extended Kalman Filter

This chapter derives the propagation equations and state and covariance update equations for the extended Kalman filter. These equations will be used later in an assessment of filter performance based on coastline measurements obtained from a satellite in LEO.

5.1 Overview of the Kalman Filter

The Kalman filter (KF) is commonly used in navigation of aerospace vehicles because it allows for autonomous estimates of the vehicle's state and covariance.

Figure 20: Observed image frame found in template image based on normalized cross correlation values

Introduced by Rudolf Kalman in 1960,⁴⁵ the KF is a sequential filter consisting of two steps: a propagate step and an update step. The classic-discrete continuous KF assumes continuous linear dynamics with discrete updates from a linear measurement model,

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t) \mathbf{x}(t) + \mathbf{G}(t) \mathbf{w}(t) \quad (8)$$

$$\mathbf{y}_k = \mathbf{h}_k \mathbf{x}_k + \mathbf{v}_k \quad (9)$$

where $\mathbf{x}$ is the state vector, $\mathbf{w}$ is zero-mean white noise, $\mathbf{y}$ is the measurement vector, $\mathbf{H}$ is the measurement mapping matrix, and $\mathbf{v}$ is the measurement noise. The $\mathbf{G}(t) \mathbf{w}(t)$ term describes the process noise perturbing the state expressed as $\dot{\mathbf{x}} = \mathbf{F}(t) \mathbf{x}(t)$. If it is possible to perfectly predict the dynamics of the system, it can be assumed that $\mathbf{w} = 0$; however this is rarely the case in practice. In addition, the absence of process noise will cause numerical complications in the solution resulting in the filter “over converging”.

Most aerospace applications typically involve non-linear problems. Because the KF is derived for linear systems, these cases require the use of a new variation of the KF, called the Extended Kalman filter (EKF).^{45, 46, 47}

5.2 Derivation of Propagation Equations

Due to the non-linearity of an EKF, the model must be linearized about a reference state. The non-linear dynamics can be expressed as

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, t) + \mathbf{w}(t) \quad (10)$$

where $\mathbf{w}(t)$ is again zero-mean white noise, further defined as a stationary stochastic process with a constant power spectral density function (PSD). In this case, this means that the random variables changing with time (i.e. the stochastic process) are invariant under a translation in time. Therefore,

$$E[\mathbf{w}(t)] = 0 \quad (11)$$

Referring back to Equation 10, integration determines the state as

$$\mathbf{x}(t) = \mathbf{x}(t_{k-1}) + \int_{t_{k-1}}^t \mathbf{f}(\mathbf{x}, \tau) d\tau + \int_{t_{k-1}}^t \mathbf{w}(\tau) d\tau \quad (12)$$

Taking the expected value of the original differential equation gives

$$\begin{aligned} E[\dot{\mathbf{x}}(t)] &= E[\mathbf{f}(\mathbf{x}, t)] + E[\mathbf{w}(t)] \\ \frac{d}{dt}E[\mathbf{x}(t)] &= E[\mathbf{f}(\mathbf{x}, t)] \\ \dot{\hat{\mathbf{x}}}(t) &= \hat{\mathbf{f}}(\mathbf{x}, t) \end{aligned} \quad (13)$$

In addition, we can show that the expected value of Equation 12 is

$$E[\mathbf{x}(t)] = E[\mathbf{x}(t_{k-1})] + \int_{t_{k-1}}^t E[\mathbf{f}(\mathbf{x}, \tau)] d\tau + \int_{t_{k-1}}^t E[\mathbf{w}(\tau)] d\tau \quad (14)$$

$$\hat{\mathbf{x}}(t) = \hat{\mathbf{x}}(t_{k-1}) + \int_{t_{k-1}}^t \hat{\mathbf{f}}(\mathbf{x}, \tau) d\tau \quad (15)$$

We also know that covariance is defined as

$$\mathbf{P}(t) = E \left[(\mathbf{x}(t) - \hat{\mathbf{x}}(t)) (\mathbf{x}(t) - \hat{\mathbf{x}}(t))^T \right] \quad (16)$$

We can rewrite this in its expanded form as

$$\mathbf{P}(t) = E \left[\mathbf{x}(t) \mathbf{x}(t)^T \right] - \hat{\mathbf{x}}(t) \hat{\mathbf{x}}(t)^T \quad (17)$$

In order to propagate the covariance, we need a differential equation which is obtained by taking the derivative of the above equation.

$$\dot{\mathbf{P}}(t) = E \left[\dot{\mathbf{x}}(t) \mathbf{x}(t)^T \right] + E \left[\mathbf{x}(t) \dot{\mathbf{x}}(t)^T \right] - \dot{\hat{\mathbf{x}}}(t) \mathbf{x}(t)^T - \hat{\mathbf{x}}(t) \dot{\hat{\mathbf{x}}}(t)^T \quad (18)$$

Focusing on the first term, we find that

$$\begin{aligned} E \left[\dot{\mathbf{x}}(t) \mathbf{x}(t)^T \right] &= E \left[(\mathbf{f}(\mathbf{x}, t) + \mathbf{w}(t)) \mathbf{x}(t)^T \right] \\ &= E \left[\mathbf{f}(\mathbf{x}, t) \mathbf{x}(t)^T \right] + E \left[\mathbf{w}(t) \mathbf{x}(t)^T \right] \\ &= E \left[\mathbf{f}(\mathbf{x}, t) \mathbf{x}(t)^T \right] + \\ &\quad E \left[\mathbf{w}(t) \left[\mathbf{x}(t_{k-1}) + \int_{t_{k-1}}^t \mathbf{f}(\mathbf{x}, \tau) d\tau + \int_{t_{k-1}}^t \mathbf{w}(\tau) d\tau \right]^T \right] \\ &= E \left[\mathbf{f}(\mathbf{x}, t) \mathbf{x}(t)^T \right] + E \left[\mathbf{w}(t) \mathbf{x}(t_{k-1})^T \right] + \\ &\quad \int_{t_{k-1}}^t E \left[\mathbf{w}(t) \mathbf{f}(\mathbf{x}, \tau)^T \right] d\tau + \int_{t_{k-1}}^t E \left[\mathbf{w}(t) \mathbf{w}(\tau)^T \right] d\tau \\ &= E \left[\mathbf{f}(\mathbf{x}, t) \mathbf{x}(t)^T \right] + \mathbf{Q}(t) \int_{t_{k-1}}^t \delta(t - \tau) d\tau \end{aligned} \quad (19)$$

where $\mathbf{Q}(t)$ represents process noise.

We know that by definition,

$$\int_{-\infty}^{\infty} \delta(t - \tau) d\tau = 1 \quad (20)$$

If $t_{k-1} < t < t_k$, we can say

$$\int_{t_{k-1}}^t \delta(t - \tau) d\tau + \int_t^{t_k} \delta(t - \tau) d\tau = 1 \quad (21)$$

Since the impulse is at t , each side of the integral must contribute equally to the outcome; therefore, we can say

$$E \left[\dot{\mathbf{x}}(t) \mathbf{x}(t)^T \right] = E \left[\mathbf{f}(\mathbf{x}, t) \mathbf{x}(t)^T \right] + \frac{1}{2} \mathbf{Q}(t) \quad (22)$$

Returning to Equation 18 and substituting in Equation 13,

$$\dot{\mathbf{P}}(t) = E \left[\dot{\mathbf{x}}(t) \mathbf{x}(t)^T \right] + E \left[\mathbf{x}(t) \dot{\mathbf{x}}(t)^T \right] - \hat{\mathbf{f}}(\mathbf{x}, t) \mathbf{x}(t)^T - \hat{\mathbf{x}}(t) \mathbf{f}(\hat{\mathbf{x}}, t)^T \quad (23)$$

Consequently, substituting Equation 22 gives

$$\begin{aligned} \dot{\mathbf{P}}(t) &= E \left[\mathbf{f}(\mathbf{x}, t) \mathbf{x}(t)^T \right] + E \left[\mathbf{x}(t) \mathbf{f}(\hat{\mathbf{x}}, t)^T \right] + \mathbf{Q}(t) - \hat{\mathbf{f}}(\mathbf{x}, t) \hat{\mathbf{x}}(t)^T - \hat{\mathbf{x}}(t) \mathbf{f}(\hat{\mathbf{x}}, t)^T \\ &= E \left[\mathbf{f}(\mathbf{x}, t) (\mathbf{x}(t) - \hat{\mathbf{x}}(t))^T \right] + E \left[(\mathbf{x}(t) - \hat{\mathbf{x}}(t)) \mathbf{f}(\hat{\mathbf{x}}, t)^T \right] + \mathbf{Q}(t) \end{aligned} \quad (24)$$

noting that $\mathbf{Q}(t) = \mathbf{Q}(t)^T$.

Because $\mathbf{f}(\mathbf{x}, t)$ is non-linear, we proceed by linearizing about the state estimate, $\hat{\mathbf{x}}$ using a Taylor Series expansion. Neglecting higher order terms, this can be expressed as

$$\mathbf{f}(\mathbf{x}, t) \approx \mathbf{f}(\hat{\mathbf{x}}, t) + \left. \frac{\partial \mathbf{f}(\mathbf{x}, t)}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}} (\mathbf{x} - \hat{\mathbf{x}}) \quad (25)$$

where we can define the Jacobian as

$$\mathbf{F} = \left. \frac{\partial \mathbf{f}(\mathbf{x}, t)}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}} \quad (26)$$

The expected value of this approximation becomes our state propagation equation,

$$\begin{aligned}
\dot{\hat{\mathbf{x}}} &= \hat{\mathbf{f}}(\mathbf{x}, t) \\
&= E[\mathbf{f}(\mathbf{x}, t)] \\
&\approx E[\mathbf{f}(\hat{\mathbf{x}}, t) + \mathbf{F}(t)(\mathbf{x}(t) - \hat{\mathbf{x}}(t))] \\
&\approx \mathbf{f}(\hat{\mathbf{x}}, t) + \mathbf{F}(t)[\hat{\mathbf{x}}(t) - \hat{\mathbf{x}}(t)] \\
&\approx \mathbf{f}(\hat{\mathbf{x}}, t)
\end{aligned} \tag{27}$$

To derive the covariance propagation equation, we begin by returning to Equation 23 to get

$$\dot{\mathbf{P}}(t) = E[\mathbf{f}(\mathbf{x}, t)(\mathbf{x}(t) - \hat{\mathbf{x}}(t))^T] + E[(\mathbf{x}(t) - \hat{\mathbf{x}}(t))\mathbf{f}(\mathbf{x}, t)^T] + \mathbf{Q}(t) \tag{28}$$

where we know from Equation 27 that

$$\mathbf{f}(\mathbf{x}, t) \approx \mathbf{f}(\hat{\mathbf{x}}, t) + \mathbf{F}(t)(\mathbf{x}(t) - \hat{\mathbf{x}}(t)) \tag{29}$$

Direct substitution gives our final propagation equation for the covariance as

$$\begin{aligned}
\dot{\mathbf{P}}(t) &= E[\mathbf{f}(\hat{\mathbf{x}}, t)(\mathbf{x}(t) - \hat{\mathbf{x}}(t))^T + \mathbf{F}(t)(\mathbf{x}(t) - \hat{\mathbf{x}}(t))(\mathbf{x}(t) - \hat{\mathbf{x}}(t))^T] \\
&\quad + E[(\mathbf{x}(t) - \hat{\mathbf{x}}(t))\mathbf{f}(\hat{\mathbf{x}}, t)^T + (\mathbf{x}(t) - \hat{\mathbf{x}}(t))(\mathbf{x}(t) - \hat{\mathbf{x}}(t))^T \mathbf{F}^T] + \mathbf{Q}(t)
\end{aligned}$$

Distributing the expected value operator gives

$$\begin{aligned}
\dot{\mathbf{P}}(t) &= \mathbf{f}(\hat{\mathbf{x}}, t)(\mathbf{x}(t) - \hat{\mathbf{x}}(t))^T + \mathbf{F}(t)E[(\mathbf{x}(t) - \hat{\mathbf{x}}(t))(\mathbf{x}(t) - \hat{\mathbf{x}}(t))^T] \\
&\quad + (\mathbf{x}(t) - \hat{\mathbf{x}}(t))\mathbf{f}(\hat{\mathbf{x}}, t)^T + E[(\mathbf{x}(t) - \hat{\mathbf{x}}(t))(\mathbf{x}(t) - \hat{\mathbf{x}}(t))^T] \mathbf{F}^T \\
&\quad + \mathbf{Q}(t)
\end{aligned}$$

Recalling Equation 17, we can further simplify this to the final form of the covari-

ance propagation equation,

$$\dot{\mathbf{P}}(t) = \mathbf{F}(t) \mathbf{P}(t) + \mathbf{P}(t) \mathbf{F}(t)^T + \mathbf{Q}(t) \quad (30)$$

5.3 Derivation of Measurement Step and Update Equations

After propagating the initial state and covariance, we can use sensor measurements to improve our estimates using the update equations. We first begin with the non-linear measurement equation where $\tilde{\mathbf{y}}$ represents the measurement in terms of a measurement mapping matrix $\mathbf{H}_k$ with added zero-mean white noise $\mathbf{v}$.

$$\tilde{\mathbf{y}}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \quad (31)$$

In these equations, the subscript k denotes the measurements obtained at time t_k . Recalling the properties for zero-mean white noise, we know that

$$E[\mathbf{v}_k] = 0 \quad (32)$$

We can also define the measurement covariance $\mathbf{R}_k$, as

$$\mathbf{R}_k = E[(\mathbf{v}_k - \hat{\mathbf{v}}_k)(\mathbf{v}_k - \hat{\mathbf{v}}_k)^T] \quad (33)$$

5.3.1 State Update Equations

To determine the state update equations, we begin by taking the expected value of the measurement $\tilde{\mathbf{y}}_k$. This gives

$$E[\tilde{\mathbf{y}}_k] = \hat{\mathbf{y}}_k = \mathbf{H}_k \mathbf{x}_k + E[\mathbf{v}_k] = \mathbf{H}_k \mathbf{x}_k \quad (34)$$

where the measurement sensitivity matrix $\mathbf{H}_k$ can be defined as

$$\mathbf{H}_k = \left. \frac{\partial \mathbf{h}(\mathbf{x}_k)}{\partial \mathbf{x}_k} \right|_{\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^-} \quad (35)$$

We can also define the errors between the estimated and actual state as

$$\mathbf{e}_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_k^- \quad (36)$$

$$\mathbf{e}_k^+ = \mathbf{x}_k - \hat{\mathbf{x}}_k^+ \quad (37)$$

where the superscript “ $-$ ” indicates the *a priori* estimate, or the estimate before the measurement is processed, and “ $+$ ” indicates the *a posteriori* estimate after the measurement has been processed. Since we want our filter to be unbiased, we expect our errors to be zero, or

$$E[\mathbf{e}_k^-] = E[\mathbf{e}_k^+] = 0 \quad (38)$$

To determine the update equations, we can choose a state update of the form

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\tilde{\mathbf{y}}_k - \mathbf{H}_k \mathbf{x}_k) \quad (39)$$

This form allows for some gain matrix $\mathbf{K}$ to be applied which describes the relationship between the measurement and the state estimate (i.e. lower values of $\mathbf{K}$ indicate that there is a smaller error and therefore the estimate can be better trusted).

Referring to Equation 37, we can now rewrite it as

$$\mathbf{e}_k^+ = \mathbf{x}_k - \hat{\mathbf{x}}_k^- - \mathbf{K}_k(\hat{\mathbf{y}}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \quad (40)$$

Substitution for $\hat{\mathbf{y}}_k$ as well as simplification yields

$$\begin{aligned}
\mathbf{e}_k^+ &= \mathbf{e}_k^- - \mathbf{K}_k (\mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \\
&= \mathbf{e}_k^- - \mathbf{K}_k (\mathbf{H}_k (\mathbf{x}_k - \hat{\mathbf{x}}_k^-) + \mathbf{v}_k) \\
&= \mathbf{e}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{e}_k^- - \mathbf{K}_k \mathbf{v}_k \\
&= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{e}_k^- - \mathbf{K}_k \mathbf{v}_k
\end{aligned} \tag{41}$$

5.3.2 Covariance of the State Update Equations

To obtain the state covariance update equations, we begin using the previously defined error equations to rewrite the covariance as

$$\begin{aligned}
\mathbf{P}_k^- &= E [(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T] \\
&= E [\mathbf{e}_k^- \mathbf{e}_k^{-T}]
\end{aligned} \tag{42}$$

$$\begin{aligned}
\mathbf{P}_k^+ &= E [(\mathbf{x}_k - \hat{\mathbf{x}}_k^+)(\mathbf{x}_k - \hat{\mathbf{x}}_k^+)^T] \\
&= E [\mathbf{e}_k^+ \mathbf{e}_k^{+T}]
\end{aligned} \tag{43}$$

Direct substitution of Equation 41 into Equation 43 gives

$$\mathbf{P}_k^+ = E \left[((\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{e}_k^- - \mathbf{K}_k \mathbf{v}_k) ((\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{e}_k^- - \mathbf{K}_k \mathbf{v}_k)^T \right] \tag{44}$$

Expansion of these terms results in

$$\begin{aligned}
\mathbf{P}_k^+ &= E \left[((\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{e}_k^- - \mathbf{K}_k \mathbf{v}_k) \left((\mathbf{e}_k^{-T} (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T - \mathbf{v}_k^T \mathbf{K}_k^T) \right) \right] \\
&= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) E [\mathbf{e}_k^- \mathbf{e}_k^{-T}] (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) E [\mathbf{e}_k^- \mathbf{v}_k^T] \mathbf{K}_k^T \\
&\quad - \mathbf{K}_k E [\mathbf{v}_k \mathbf{e}_k^{-T}] (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k E [\mathbf{v}_k \mathbf{v}_k^T] \mathbf{K}_k^T
\end{aligned} \tag{45}$$

Because the expected value of any terms containing $\mathbf{v}_k$ is zero, our final state covariance update equation becomes

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T \quad (46)$$

Equation 46 is often called the Joseph Form of the covariance update. It is commonly used for two reasons: 1) it is valid for any Kalman gain and 2) it enforces that $\mathbf{P}_k^+$ remains symmetric, alleviating the accumulation of numerical errors during long simulations.

5.3.3 Derivation of the Optimal Kalman Gain

In optimization, the primary goal is to maximize or minimize some objective function. When deriving the optimal Kalman gain, we specifically want to minimize the mean square error of the state after it has been updated. We can do this by writing our objective function as a function of $\mathbf{K}_k$, or

$$\min J(\mathbf{K}_k) = E [\mathbf{e}_k^{+T} \mathbf{e}_k^+] \quad (47)$$

Because the $\mathbf{e}_k^{+T} \mathbf{e}_k^+$ term is scalar, we can say

$$E [\text{Tr} [\mathbf{e}_k^{+T} \mathbf{e}_k^+]] \quad (48)$$

Noting identities for the trace operator,⁴⁸ we can say

$$\begin{aligned} \min J(\mathbf{K}_k) &= E [\text{Tr} [\mathbf{e}_k^+ \mathbf{e}_k^{+T}]] \\ &= \text{Tr} [E [\mathbf{e}_k^+ \mathbf{e}_k^{+T}]] \\ &= \text{Tr} [\mathbf{P}_k^+] \end{aligned} \quad (49)$$

Applying the first differential condition allows us to calculate the optimal gain $\mathbf{K}_k$, found as a solution to

$$\frac{\partial J(\mathbf{K}_k)}{\partial \mathbf{K}_k} = \frac{\text{Tr}[\mathbf{P}_k^+]}{\partial \mathbf{K}_k} = 0 \quad (50)$$

We also know that because the trace operator is a linear operator, the trace of the covariance can be written as

$$\text{Tr} [\mathbf{P}_k^+] = \text{Tr} \left[(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T \right] \quad (51)$$

Expansion yields

$$\text{Tr} [\mathbf{P}_k^+] = \text{Tr} \left[\mathbf{P}_k^- - \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{K}_k^T - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^- + \mathbf{K}_k (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k) \mathbf{K}_k^T \right] \quad (52)$$

Applying the first differential condition to this equation requires taking the partial derivative of Equation 52 with respect to the Kalman gain. Recall from matrix vector calculus⁴⁸ that

$$\frac{\partial \text{Tr}[\mathbf{X} \mathbf{B} \mathbf{X}^T]}{\partial \mathbf{X}} = \mathbf{X} \mathbf{B}^T + \mathbf{X} \mathbf{B} \quad (53)$$

$$\frac{\partial \text{Tr}[\mathbf{B} \mathbf{X}^T]}{\partial \mathbf{X}} = \mathbf{B} \quad (54)$$

$$\frac{\partial \text{Tr}[\mathbf{B} \mathbf{X}]}{\partial \mathbf{X}} = \mathbf{B}^T \quad (55)$$

Therefore, we get

$$\frac{\partial \text{Tr}[\mathbf{P}_k^+]}{\partial \mathbf{K}_k} = -\mathbf{P}_k^- \mathbf{H}_k^T - \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{K}_{opt} (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k) + \mathbf{K}_{opt} (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k) = 0 \quad (56)$$

This equation simplifies to

$$2\mathbf{P}_k^- \mathbf{H}_k^T = 2\mathbf{K}_{opt} (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k) \quad (57)$$

which gives our optimal Kalman gain as

$$\mathbf{K}_{opt} = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (58)$$

Figure 21 summarizes the extended Kalman filter in a flow chart.

Figure 21: Flow chart summarizing the extended Kalman filter algorithm.

6 Coastline Optical Navigation Filter Implementation

To determine the feasibility of using coastlines as landmarks for autonomous navigation, a simulation was created depicting the view of a spacecraft while in LEO. Depending on the orbital parameters and camera specifications, the viewable coastlines were used to estimate the state and covariance of the spacecraft. The mathematical models used for the EKF will be discussed, along with filter results and performance.

6.1 Spacecraft Mathematical Models

To estimate the state of the spacecraft, we first must express the dynamics of the spacecraft using the two-body problem from orbital mechanics. We derive this equation in terms of conic sections to allow the orbit of the spacecraft to be described using classical orbital elements: semi-major axis a , eccentricity e , inclination i , argument of periapsis Ω , true anomaly ν , and right ascension of the ascending node ω . Considering the forces acting on a spacecraft in orbit, we must account for gravitational forces, drag, thrust, and sometimes a 3rd body, as well as any miscellaneous forces. However, we can restrict this problem by making some assumptions. First of all, we can assume the spacecraft is traveling far enough above the Earth's atmosphere that the drag force is negligible. If we assume the spacecraft does not change its orbit, we can ignore the thrust force. In the case of LEO, we assume that there are no other body forces (i.e. forces from the Sun, moon, etc.). Also, compared to Earth's gravity, we can assume other forces such as electromagnetic forces, solar radiation, etc. are negligible. Finally, we can assume the mass of the spacecraft is much smaller than the mass of Earth. Additional assumptions include the symmetry of an assumed spherical Earth and a constant mass of the spacecraft. These assumptions allow us to model the earth as a point

mass with gravity as the only force acting on the spacecraft.

$$\sum \vec{\mathbf{F}}_{gravity} = \sum m\ddot{\mathbf{r}} \quad (59)$$

Applying Newton's Law of Universal Gravitation,

$$\vec{\mathbf{F}}_{gravity} = -\frac{\mu m}{r^3} \mathbf{r} \quad (60)$$

where r represents the magnitude of the vector $\mathbf{r}$, or $r = \|\mathbf{r}\|$.

Equation 59 and Equation 60 gives

$$\vec{\mathbf{F}}_{gravity} = -\frac{\mu m}{r^3} \mathbf{r} = m\ddot{\mathbf{r}} \quad (61)$$

Dividing by mass on each side gives our final expression to be

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3} \mathbf{r} \quad (62)$$

It is not possible to solve for the position as a function of time using a closed form solution. Instead, we can find an exact solution through variable substitution.

Crossing both sides of Equation 62 with specific angular momentum, $\mathbf{h}$,

$$\ddot{\mathbf{r}} \times \mathbf{h} = -\frac{\mu}{r^3} (\mathbf{r} \times \mathbf{h}) \quad (63)$$

which due to the cross product identity $\mathbf{a} \times \mathbf{b} = -(\mathbf{b} \times \mathbf{a})$ simplifies to

$$\ddot{\mathbf{r}} \times \mathbf{h} = \frac{\mu}{r^3} (\mathbf{h} \times \mathbf{r}) \quad (64)$$

Looking at the left side of the equation, we know that we can rewrite it using the chain rule as

$$\ddot{\mathbf{r}} \times \mathbf{h} = \frac{d}{dt} (\dot{\mathbf{r}} \times \mathbf{h}) \quad (65)$$

since $\mathbf{h}$ is constant over time.

Looking at the right side of Equation 63, we use the vector identity $(\mathbf{a} \times \mathbf{b}) \times \mathbf{c} = \mathbf{b}(\mathbf{a} \cdot \mathbf{c}) - \mathbf{a}(\mathbf{b} \cdot \mathbf{c})$ to get

$$\mathbf{h} \times \mathbf{r} = (\mathbf{r} \times \dot{\mathbf{r}}) \times \mathbf{r} = \dot{\mathbf{r}}(\mathbf{r} \cdot \mathbf{r}) - \mathbf{r}(\dot{\mathbf{r}} \cdot \mathbf{r}) \quad (66)$$

which simplifies to

$$\mathbf{h} \times \mathbf{r} = \dot{\mathbf{r}}r^2 - \mathbf{r}(\dot{\mathbf{r}} \cdot \mathbf{r}) \quad (67)$$

Knowing that $\dot{\mathbf{r}} \cdot \mathbf{r} = \dot{\mathbf{r}}r$, we can further simplify the equation to

$$\mathbf{h} \times \mathbf{r} = \dot{\mathbf{r}}r^2 - \mathbf{r}\dot{\mathbf{r}}r \quad (68)$$

If we multiply both sides by $\frac{\mu}{r^3}$, we get

$$\frac{\mu}{r^3}(\mathbf{h} \times \mathbf{r}) = \frac{\mu}{r^3}(\dot{\mathbf{r}}r^2 - \mathbf{r}\dot{\mathbf{r}}r) \quad (69)$$

Additionally, we also know that the derivative of a quotient gives

$$\mu \frac{d}{dt} \left(\frac{\mathbf{r}}{r} \right) = \mu(\dot{\mathbf{r}}r^{-1} - \mathbf{r}\dot{\mathbf{r}}r^{-2}) \quad (70)$$

Factoring $\frac{1}{r^3}$ from the right side gives

$$\mu \frac{d}{dt} \left(\frac{\mathbf{r}}{r} \right) = \frac{\mu}{r^3} (\dot{\mathbf{r}}r^2 - \mathbf{r}\dot{\mathbf{r}}r) \quad (71)$$

Equating Equation 69 and Equation 71,

$$\frac{\mu}{r^3}(\mathbf{h} \times \mathbf{r}) = \mu \frac{d}{dt} \left(\frac{\mathbf{r}}{r} \right) \quad (72)$$

Substituting Equation 65 into Equation 72 and Equation 64, we obtain

$$\frac{d}{dt} (\dot{\mathbf{r}} \times \mathbf{h}) = \mu \frac{d}{dt} \left(\frac{\mathbf{r}}{r} \right) \quad (73)$$

Integration yields

$$\dot{\mathbf{r}} \times \mathbf{h} = \mu \left(\frac{\mathbf{r}}{r} \right) + \mathbf{b} \quad (74)$$

where $\mathbf{b}$ is a constant vector of integration. Taking the dot product of both sides of the equation with $\mathbf{r}$ gives

$$\mathbf{r} \cdot (\dot{\mathbf{r}} \times \mathbf{h}) = \mathbf{r} \cdot \left(\mu \frac{\mathbf{r}}{r} + \mathbf{b} \right) \quad (75)$$

Using the vector identity $\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = (\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c}$, we have

$$\begin{aligned} \mathbf{r} \cdot (\dot{\mathbf{r}} \times \mathbf{h}) &= (\mathbf{r} \times \dot{\mathbf{r}}) \cdot \mathbf{h} \\ &= (\mathbf{r} \times \dot{\mathbf{r}}) \cdot \mathbf{h} \\ &= \mathbf{h} \cdot \mathbf{h} \\ &= h^2 \end{aligned} \quad (76)$$

Substituting this result into Equation 75, it becomes

$$h^2 = \mathbf{r} \cdot \left(\mu \frac{\mathbf{r}}{r} + \mathbf{b} \right) \quad (77)$$

Expanding the right side of this equation, we obtain

$$\begin{aligned} \mathbf{r} \cdot \left(\mu \frac{\mathbf{r}}{r} + \mathbf{b} \right) &= \frac{\mu}{r} (\mathbf{r} \cdot \mathbf{r}) + (\mathbf{r} \cdot \mathbf{b}) \\ &= \frac{\mu}{r} (r^2) + \mathbf{r} \cdot \mathbf{b} \\ &= \mu r + r b \cos \nu \end{aligned} \quad (78)$$

where ν is the angle between $\mathbf{r}$ and $\mathbf{b}$, thus

$$h^2 = \mu r + r b \cos \nu \quad (79)$$

Rearranging to solve for r ,

$$r = \frac{h^2/\mu}{1 + (b/\mu) \cos \nu} \quad (80)$$

From geometry, we know that conic sections are expressed in polar form as

$$r = \frac{k_1}{1 + (k_2 \cos \nu)} \quad (81)$$

where r is the magnitude of the position vector, k_1 is the semi-latus rectum, further described by p , k_2 is the eccentricity e of the orbit, and ν is the true anomaly given in degrees or radians. Using the classic expression

$$p = a(1 - e^2) \quad (82)$$

where a is the semimajor axis. Finally we get what is defined as the two-body equation of motion relating spacecraft dynamics and orbital geometry (see Figure 22)

$$r = \frac{a(1 - e^2)}{1 + (e \cos \nu)} \quad (83)$$

This equation is derived in the PQW reference frame, depicted in Figure 23a.

Figure 22: Visual representation of orbital elements

The vector form of r as well as the velocity can be found by

$$\mathbf{r} = \begin{bmatrix} r \cos \nu & p \sin \nu & 0 \end{bmatrix} \quad (84)$$

$$\dot{\mathbf{r}} = \begin{bmatrix} -\sqrt{\frac{\mu}{p}} \sin \nu & \sqrt{\frac{\mu}{p}}(e + \cos \nu) & 0 \end{bmatrix} \quad (85)$$

It can also be expressed in the more commonly used IJK frame, depicted in Figure 23b using the rotation matrix $\mathbf{A}$,

$$\mathbf{A} = \begin{bmatrix} \cos \Omega \cos \omega - \sin \Omega \sin \omega \cos i & -\cos \Omega \sin \omega - \sin \Omega \cos \omega \cos i & \sin \Omega \sin i \\ \sin \Omega \cos \omega + \cos \Omega \sin \omega \cos i & -\sin \Omega \sin \omega + \cos \Omega \cos \omega \cos i & -\cos \Omega \sin i \\ \sin \omega \sin i & \cos \omega \sin i & \cos i \end{bmatrix} \quad (86)$$

where

$$\mathbf{r} = \mathbf{A} \mathbf{r}_{PQW}^T \quad (87)$$

$$\dot{\mathbf{r}} = \mathbf{A} \dot{\mathbf{r}}_{PQW}^T \quad (88)$$

Figure 23: Coordinate frames used for spacecraft dynamics modeling

6.2 State Propagation using an Extended Kalman Filter

Rewriting the two-body equation in state-space form leads to the following expression for $\mathbf{f}(\mathbf{x}, t)$

$$\mathbf{x} = \begin{bmatrix} \mathbf{r} \\ \dot{\mathbf{r}} \end{bmatrix} \quad (89)$$

$$\mathbf{f}(\mathbf{x}, t) = \begin{bmatrix} \dot{\mathbf{r}} \\ -\frac{\mu}{r^3} \mathbf{r} \end{bmatrix} \quad (90)$$

Figure 24: Line-of-sight vectors for measurement model

Because $\dot{\mathbf{r}}$ is nonlinear in $\mathbf{r}$, $\dot{\mathbf{x}}$ must be linearized to first order about the estimate to find $\mathbf{F}$,

$$\mathbf{F} = \frac{\partial \mathbf{f}(\mathbf{x}, t)}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}} = \begin{bmatrix} \frac{\partial \dot{\mathbf{r}}}{\partial \mathbf{r}} & \frac{\partial \dot{\mathbf{r}}}{\partial \dot{\mathbf{r}}} \\ \frac{\partial \ddot{\mathbf{r}}}{\partial \mathbf{r}} & \frac{\partial \ddot{\mathbf{r}}}{\partial \dot{\mathbf{r}}} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \\ \frac{-\mu}{\|\hat{\mathbf{r}}\|^3} \mathbf{I}_{3 \times 3} + \frac{-\mu}{\|\hat{\mathbf{r}}\|^5} \hat{\mathbf{r}} \hat{\mathbf{r}}^T & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (91)$$

Since both $\dot{\mathbf{x}}$ and $\dot{\mathbf{P}}$ are now known, each can be integrated to propagate to a new *a priori* estimate of the state and covariance, $\hat{\mathbf{x}}_k^-$ and $\hat{\mathbf{P}}_k^-$, which are then updated by the EKF.

6.3 Measurement Model for Coastline Points in Images

Line-of-sight (LOS) measurements are produced from the images obtained by the spacecraft's camera. These measurements are a nonlinear function of the spacecraft state and may be determined by the geometry between the Earth and the spacecraft. From Figure 24, it is clear that the following geometric relations are obtained by

$$\mathbf{s}_i = \mathbf{o}_i - \mathbf{r}_{sc} \quad (92)$$

$$\mathbf{e}_{iI} = \frac{\mathbf{s}_i}{\|\mathbf{s}_i\|} \quad (93)$$

where $\mathbf{r}_{sc}$ is the position of the spacecraft from the center of the earth, $\mathbf{s}_i$ is the vector from the spacecraft to the i^{th} observed point, or landmark, and $\mathbf{o}_i$ is the vector from the center of the earth to the landmark. The measurement model for a single observed coastline point can then be described as

$$\mathbf{h}(\mathbf{x}) = \mathbf{e}_{iC} = \mathbf{T}_C^I \mathbf{e}_{iI} = \mathbf{T}_C^I \frac{\mathbf{s}_i}{[\mathbf{s}_i^T \mathbf{s}_i]^{1/2}} \quad (94)$$

where $\mathbf{T}_C^I$ is the rotation matrix from the inertial frame to the camera frame. A good estimate of $\mathbf{T}_C^I$ is assumed to be available from a star tracker. From here, the measurement sensitivity matrix may be computed as

$$\mathbf{H}_i = \mathbf{T}_C^I \frac{1}{\|\mathbf{s}_i\|} \begin{bmatrix} \{\mathbf{e}_{iI} \mathbf{e}_{iI}^T - \mathbf{I}_{3 \times 3}\} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (95)$$

Additionally, the covariance for a unit vector (LOS) measurement is given by [49] as

$$\mathbf{R}_i = E[(\mathbf{e}_{iC} - \bar{\mathbf{e}}_{iC})(\mathbf{e}_{iC} - \bar{\mathbf{e}}_{iC})^T] \approx \sigma_\Theta^2 (\mathbf{I}_{3 \times 3} - \bar{\mathbf{e}}_{iC} \bar{\mathbf{e}}_{iC}^T) \quad (96)$$

It should be noted that in order to calculate the Kalman gain, as seen in Equation 58, the quantity $\mathbf{H}_i \mathbf{P}^- \mathbf{H}_i^T + \mathbf{R}_i$ must be invertible. In this particular case, the null space of the measurement covariance, $\mathbf{R}_i$, is the same as the null space of $\mathbf{H}_i \mathbf{P}^- \mathbf{H}_i^T$ assuming $\mathbf{R}_i$ is as seen in Equation 96. Therefore, a modified version of $\mathbf{K}_i$ is used,

$$\mathbf{K}_i = \mathbf{P}^- \mathbf{H}_i^T (\mathbf{H}_i \mathbf{P}^- \mathbf{H}_i^T + \mathbf{R}_i + \nu \mathbf{e}_i \mathbf{e}_i^T)^{-1} \quad (97)$$

As explained in [31], this does not change the value of $\mathbf{K}_i$ since it is equivalent to adding a $\mathbf{0}$ to the overall equation,

$$\mathbf{H}_i^T \mathbf{e}_i = \mathbf{0} \quad (98)$$

7 LEO Simulation Results

7.1 Coastline Visibility

The ground projected field-of-view (GFOV) can be calculated based on the geometry of the spacecraft position and desired field-of-view (FOV) of the camera,

$$\text{GFOV} = 2h \tan\left(\frac{\text{FOV}}{2}\right) \quad (99)$$

where h is the spacecraft altitude. A 30° FOV was chosen (based on the average FOV used for satellite imaging) resulting in a viewable footprint of approximately 536 km. A visual representation of the relative size of this footprint as seen by a camera can be seen in Figure 25a which shows a portion of the Great Lakes.

(a) Simulated satellite image incorporating a camera with a 30° FOV in a 1,000 km circular polar orbit. The white lines represent the coastline data.

(b) View of coastlines showing exact footprint with various identified coastline points (*) within the spacecraft FOV.

Figure 25: Simulated view of coastlines and the corresponding footprint based on a 1,000 km circular polar orbit

For this simulation, it is important to ensure that coastlines will be visible based on the spacecraft's orbit and camera parameters. Therefore, simulations were run to determine the percentage of time coastlines would be visible from the spacecraft's position in a 1,000 km orbit with varying inclinations. For simplicity, other orbital parameters, including the right ascension of the ascending node, the argument of periapsis, and the true anomaly, were fixed at 0° . It is important to note that altering these values will change how often coastlines are able to be

detected and results may differ slightly from those shown here. Figure 26 shows the results of the simulations given various orbital inclinations and camera FOV. It is important to note that these results are orbit-specific and will vary slightly due to differences in orbital parameters, especially attitude. Consequently, the rate at which images are captured also plays a key role in determining how often coastlines are observable. For the results in Figure 26, images are captured once every 5 minutes for a total of 24 hours.

Figure 26: Orbit-specific contour plot showing the percentage of time coastline is observable based on inclination and FOV at a 1,000 km circular orbit with images taken once every five minutes

7.2 Assessment of Navigation Filter Performance

A Monte Carlo analysis was run to assess the performance of the EKF using only coastline observations. In this case, the initial state was dispersed for each run in the Monte Carlo in a manner consistent with the initial state covariance (standard deviation of 500 m in position and 5 mm/s in velocity). This allowed the author to verify that the covariance predicted by the filter is statistically consistent within the Monte Carlo numerical results. In each analysis, the camera was assumed to have a 30° FOV with a pointing error of 0.05°.

This simulation was carried out for a total of 1,000 runs for each of five orbits

(0° , 45° , and 90° , each at 1,000 km altitude; and 45° and 90° at 500 km altitude) to determine the steady state error achievable for each scenario. The additional orbital parameters of these orbits, such as the right ascension of the ascending node, the argument of periapsis, and the true anomaly, were fixed at 0° for simplicity. The following figures show that in all scenarios, the covariance predicted by the filter (gray lines) is consistent with the true covariance (black lines) and that the errors lie well within the 3σ bounds of the covariance (i.e. 2-5 km in position and 2-5 m/s in velocity).

Table 5 provides a summary of the results from the five orbits mentioned above. Results show that spacecraft in high-inclination orbits consistently produce the smallest steady state errors. This is likely due to the near global coverage over time and superior measurement geometry.

Table 5: Summary of steady state position and velocity errors (1σ) for specified orbits

Inclination	Altitude [km]	Along-track Position Error [km]	Cross-track Position Error [km]	Radial Position Error [km]	Along-track Velocity Error [m/s]	Cross-track Velocity Error [m/s]	Radial Velocity Error [m/s]
0°	1,000	0.33	0.47	1.67	1.67	0.47	0.33
45°	1,000	0.33	0.47	1.67	1.67	0.47	0.33
90°	1,000	0.20	0.43	1.0	1.0	0.43	0.20
45°	500	0.33	0.47	1.5	1.6	0.5	0.37
90°	500	0.17	0.43	0.6	0.67	0.47	0.20

(a) Position error [km] at $i=0^\circ$

(b) Velocity error [m/s] at $i=0^\circ$

Figure 27: Monte Carlo position and velocity errors for a 1,000 km equatorial (0° inclination) orbit

(a) Position error [km] at $i=45^\circ$

(b) Velocity error [m/s] at $i=45^\circ$

Figure 28: Monte Carlo position and velocity errors for a 1,000 km, 45° inclination orbit

Figure 29: Monte Carlo position and velocity errors for a 1,000 km polar (90° inclination) orbit

Figure 30: Monte Carlo position and velocity errors for a 500 km, 45° inclination orbit

(a) Position error [km] at $i=90^\circ$

(b) Velocity error [m/s] at $i=90^\circ$

Figure 31: Monte Carlo position and velocity errors for a 500 km polar (90° inclination) orbit

7.3 Discussion of Filter Results

An extended Kalman filter was used to demonstrate how observations of coastlines on the earth's surface may be used to autonomously estimate the position and velocity of a spacecraft. If a coastline is visible, an image processing algorithm may be able to detect such surface features and then match these with a database of known coastline locations (although the details of such image processing were not considered in this study). These coastline coordinates are input into the filter to estimate the spacecraft's position and velocity. The results in Table 5 show that, based on these preliminary simulations, spacecraft in high-inclination orbits are likely to have smaller steady state errors in both position and velocity measurements. This is likely due to more favorable observation geometry and measurement frequency, although a more detailed study is required. Future work related to this topic includes creating an appropriate image processing algorithm able to extract the required information from sample satellite images. It is the research team's hope that this type of approach can soon be demonstrated on-board a CubeSat, such as the WATSON (WVU Advanced Technology Satellite for Optical Navigation) mission concept being developed at West Virginia University.

8 Outreach Opportunities

The research presented in this thesis was sponsored in part by the NASA West Virginia Space Grant Consortium through the NASA WV EPSCoR (Experimental Program to Stimulate Competitive Research) program. The goals of this program are to:⁵⁰

1. Contribute to and advance NASA's vision and strategic goals as outlined in various NASA documents, specifically in terms of STEM research and workforce development
2. Contribute to the state of West Virginia's efforts at research infrastructure development particularly in the high-technology sector, and improved level of STEM capacity

3. Increase the participation of underrepresented groups in our programs for students and faculty.

To help the space grant consortium meet these goals, our research team felt it was important to contribute to community outreach efforts. In this case, the author spent a morning with Christina Maloney's 7th grade science classes at Suncrest Middle School. The author had the opportunity to talk to approximately 110 students throughout the day while introducing them to engineering and the vast amount of opportunities within the field of science, technology, engineering and mathematics (STEM). Specifically, the author was able to tell them about what different types of engineers do on a daily basis (see Figure 32) while applying Newton's Laws with specific emphasis on space applications. To demonstrate Newton's 3rd law, the classes made balloon rockets as seen in Figure 33. The students were very perceptive, willing to participate, and seemed to really enjoy the activities.

Figure 32: Discussing what different types of engineers do on a daily basis with Mrs. Maloney's 7th grade science classes at Suncrest Middle School

Figure 33: Racing balloon rockets to demonstrate Newton's 3rd law of motion in Mrs. Maloney's 7th grade science classes at Suncrest Middle School

9 Conclusion

In conclusion, this thesis has discussed in detail the trend towards small satellites and CubeSats while providing specific details about approximately 115 CubeSat missions launched from the United States. The information gathered in the CubeSat survey paves the way for the design of future CubeSat missions at West Virginia University by providing details of CubeSat missions in regards to flight processors and operating systems. Some of the information gathered in this survey has already been considered for the design of a CubeSat mission currently being developed in the WVU Applied Space Exploration Lab. This mission, entitled the WVU Advanced Technology Satellite for Optical Navigation (WATSON) will be used to implement optical navigation algorithms resulting from various research projects within the lab. Specifically, WATSON will be used to capture images of Earth for landmark navigation using coastlines. These images will be processed using image processing algorithms before being matched with a coastline database to identify the latitude and longitude coordinates of the observed coastlines. The

resulting points will be implemented into an extended Kalman filter in order to autonomously estimate the spacecraft's state and covariance. A simulation of this process was created in MATLAB to obtain results. It was determined that the filter could estimate the position and velocity of the spacecraft to within approximately 1 km and 1 m/s, respectively, for an image of approximately 536 km, based on a 1,000 km orbit and a 30° FOV, making this concept feasible for use in low Earth orbiting spacecraft.

10 Appendix

10.1 Outreach Presentation

3/24/2015

West Virginia University
BENJAMIN M. STATLER COLLEGE OF
ENGINEERING AND MINERAL RESOURCES
www.statler.wvu.edu

Introduction to Engineering and Newton's Laws of Motion

March 23rd, 2015
Suncrest Middle School

Miranda Straub
Applied Space Exploration Laboratory (ASEL)
Department of Mechanical and Aerospace Engineering
Benjamin M. Statler College of Engineering & Mineral Resources
West Virginia University (WVU)
<http://asel.mae.wvu.edu>

1

West Virginia University
BENJAMIN M. STATLER COLLEGE OF
ENGINEERING AND MINERAL RESOURCES
www.statler.wvu.edu

What is an engineer?

An **engineer** is a person who **designs**, **builds**, and **tests** products, machines, systems, structures, etc.

2

West Virginia University
BENJAMIN M. STATLER COLLEGE OF
ENGINEERING AND MINERAL RESOURCES
www.statler.wvu.edu

Is this a type of engineering?

3

West Virginia University
BENJAMIN M. STATLER COLLEGE OF
ENGINEERING AND MINERAL RESOURCES
www.statler.wvu.edu

What do I do as an engineer?

- Spacecraft navigation:
 - Where is the spacecraft in space?

4

West Virginia University
BENJAMIN M. STATLER COLLEGE OF
ENGINEERING AND MINERAL RESOURCES
www.statler.wvu.edu

How do YOU use engineering?

Newton's Laws of Motion

1. An object at rest will stay at rest (and an object in motion will stay in motion) unless acted upon by another force
2. Acceleration of an object produced by a force is directly related to the mass of the object ($F = ma$)
3. For every action there is an equal and opposite reaction

5

West Virginia University
BENJAMIN M. STATLER COLLEGE OF
ENGINEERING AND MINERAL RESOURCES
www.statler.wvu.edu

How does a spacecraft get to orbit?

- Propulsion, a perfect example of equal and opposite reactions (Newton's 3rd Law)
- Propulsion will take the spacecraft to a point far above the surface of the Earth
- Propulsion also can help increase the speed of the spacecraft

6

West Virginia University
BENJAMIN M. STATLER COLLEGE OF
ENGINEERING AND MINERAL RESOURCES
www.statler.wvu.edu

What keeps a spacecraft in orbit?

- Velocity
 - Think about swinging a bucket upside down...what happens?
- Gravity
 - Without it, the spacecraft would float away
- Why is this true?
 - **Newton's Laws!**

7

West Virginia University
BENJAMIN M. STATLER COLLEGE OF
ENGINEERING AND MINERAL RESOURCES
www.statler.wvu.edu

Questions?

8

West Virginia University
BENJAMIN M. STATLER COLLEGE OF
ENGINEERING AND MINERAL RESOURCES
www.statler.wvu.edu

Balloon Rockets!

- A simple example of propulsion
- As the air exits the balloon, the force of the air escaping backwards drives the balloon forward
- This process is very similar to how rocket fuel ignites and thrusts the rocket up into space

9

References

- [1] Garber, S., “Sputnik and the Dawn of the Space Age,” 2007.
- [2] Garber, S., “Explorer-I and Jupiter-C,” 2007.
- [3] Lightsey Research Group, “FASTRAC,” 2015.
- [4] Air Force Research Laboratory, “University Nanosat Program,” 2013.
- [5] The Aerospace Corporation, “Communications Established with Smallest Satellites Ever Place Into Orbit,” 2000.
- [6] Hinkley, D. and Janson, S., “Building Miniature Spacecraft at The Aerospace Corporation,” *Crosslink*, Vol. 10, No. 1, 2009.
- [7] Manchester, Z., Peck, M., and Filo, A., “KickSat: A Crowd-Funded Mission to Demonstrate the World’s Smallest Spacecraft,” *27th Annual AIAA/USU Conference on Small Satellites*, Logan, UT, August 2013.
- [8] Clark, C. and Logan, R., “Power Budgets for Mission Success,” *8th Annual CubeSat Developers’ Workshop*, San Luis Obispo, CA, April 2011.
- [9] California Polytechnic State University, San Luis Obispo, “CubeSat Design Specifications Rev. 13,” 2014.
- [10] NASA Ames Research Center Mission Design Division Staff, “Small Spacecraft Technology State of the Art,” Tech. rep., NASA Ames Research Center, July 2014.
- [11] Clyde Space, “Small Satellite Solar Panels,” 2015.
- [12] Pumpkin, Inc., “Begin your CubeSat Mission with the CubeSat Kit,” 2013.
- [13] Pumpkin, Inc., “Pumpkin Price List,” 2013.
- [14] Messier, D., “GLXP News: Team FREDNET Offers CubeSat Launch Opportunity,” 2013.
- [15] Chris Adolphus Esionwu, J., “CubeSat Market Analysis and Cost Breakdown,” 2014.
- [16] National Aeronautics and Space Administration, “CubeSat Launch initiative (CSLI),” 2015.
- [17] Buchen, E. and DePasquale, D., “2014 Nano/Microsatellite Market Assessment,” *SpaceWorks Enterprises, Inc.*, Atlanta, GA, March 2015.
- [18] National Aeronautics and Space Administration, “2014 Strategic Plan,” Tech. rep., NASA, 2014.
- [19] National Aeronautics and Space Administration, “Space Technology Roadmaps,” 2015.

- [20] The Space Technology Laboratory, *Space Mission Engineering: The New SMAD*, Microsom Press and Springer, third edition ed., 2011.
- [21] Jerath, N. and Ohtakay, H., “Mariner IX Optical Navigation Using Mars Lit Limb,” *Journal of Spacecraft and Rockets*, Vol. 11, No. 7, 1974, pp. 505–511.
- [22] Owen, W., “Methods of Optical Navigation,” *AAS Spaceflight Mechanics Conference*, No. AAS 11-215, New Orleans, LA, February 2011.
- [23] Christian, J. A. and Lightsey, E. G., “Onboard Image-Processing Algorithm for a Spacecraft Optical Navigation Sensor System,” *Journal of Spacecraft and Rockets*, Vol. 49, No. 2, 2012, pp. 337–352.
- [24] Liounis, A. J., Gerhard, J., and Christian, J., “Performance Assessment of Horizon-Based Optical Navigation Techniques,” *AAS GN&C Conference*, Breckenridge, CO, January 2015.
- [25] Christian, J. A., “Optical Navigation Using Planet’s Centroid and Apparent Diameter in Image,” *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 2, 2015.
- [26] Cheng, Y. and Miller, J. K., “Autonomous Landmark Based Spacecraft Navigation System,” *AAS/AIAA Astrodynamics Specialist Conference*, No. AAS 03-223, Ponce, Puerto Rico, February 2003.
- [27] Rowell, N., Parkes, S., and Dunstan, M., “Image Processing for Near Earth Object Optical Guidance Systems,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 49, No. 2, 2013, pp. 1057–1072.
- [28] Hanak, C., *Lost in Low Lunar Orbit Crater Pattern Detection and Identification*, Ph.D. thesis, University of Texas at Austin, 2009.
- [29] Rohrschneider, R. R., “Terrain Relative Navigation Using Crater Identification in Surface Topography Data,” *AIAA Guidance, Navigation, and Control Conference*, No. AIAA 2001-6490, Portland, Oregon, August 2011.
- [30] Cheng, Y., Johnson, A. E., Matthies, L. H., and Olson, C. F., “Optical Landmark Detection for Spacecraft Navigation,” *AAS/AIAA Astrodynamics Specialist Conference*, No. AAS 02-224, Ponce, Puerto Rico, February 2003.
- [31] Liounis, A., Daniel, S., and Christian, J. A., “Autonomous Navigation System Performance in the Earth-Moon System,” *AIAA Space Conference*, San Diego, CA, September 2013.
- [32] Liu, H. and Jezek, K. C., “Automated Extraction of Coastline from Satellite Imagery by Integrating Canny Edge Detection and Locally Adaptive Thresholding Methods,” *International Journal of Remote Sensing*, Vol. 25, No. 5, 2004, pp. 937–958.
- [33] National Aeronautics and Space Administration, “Earth Features Seen From Space,” 1999.

- [34] Gonzalez, R. C. and Woods, R. E., *Digital Image Processing (3rd Edition)*, Prentice Hall, Inc., Upper Saddle River, NJ, 2007.
- [35] Maini, R. and Aggarwal, H., “Study and Comparison of Various Image Edge Detection Techniques,” *International Journal of Image Processing*, Vol. 3, No. 1, 1981, pp. 1–12.
- [36] Bhardwaj, S. and Mittal, A., “A Survey on Various Edge Detector Techniques,” *Procedia Technology*, Vol. 4, 2012, pp. 220–226.
- [37] Canny, J., “A Computational Approach to Edge Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, 1986, pp. 679–698.
- [38] Heene, G. and Gautama, S., “Optimisation of a Coastline Extraction Algorithm for Object-Oriented Matching of Multisensor Satellite Imagery,” *IEEE Geoscience and Remote Sensing Symposium*, Vol. 6, 2000, pp. 2632–2634.
- [39] Patt, F. S., Woodward, R. H., and Gregg, W. W., “An Automated Method for Navigation Assessment for Earth Survey Sensors using Target Islands,” *International Journal of Remote Sensing*, Vol. 18, No. 16, 1997, pp. 3311–3336.
- [40] Eugenio, F., Marques, F., and Marcello, J., “A countour-based approach to automatic and accurate registration of multitemporal and multisensor satellite imagery,” 2002.
- [41] Christian, J. A., *Optical Navigation for a Spacecraft in a Planetary System*, Ph.D. thesis, University of Texas at Austin, 2010.
- [42] Briechle, K. and Hanebeck, U. D., “Template Matching Using Fast Normalized Cross Correlation,” *Proc. SPIE Optical Pattern Recognition XII*, Vol. 4387, No. 95, 2001.
- [43] Environmental Systems Research Institute, Inc., “ESRI Shapefile Technical Description,” Tech. rep., ESRI, July 1998.
- [44] Sellers, J. J., *Understanding Space: An Introduction to Astronautics, Third Edition*, The McGraw-Hill Companies, Inc., New York, NY, 2005.
- [45] Brown, R. G. and Hwang, P. Y. C., *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley and Sons, Inc., Hoboken, NJ, 1997.
- [46] Gelb, A., *Applied Optimal Estimation*, The M.I.T. Press, Cambridge, MA, 1974.
- [47] Maybeck, P. S., *Autonomous Robot Vehicles*, chap. The Kalman Filter: An Introduction to Concepts, Springer New York.
- [48] Petersen, K. B., Pedersen, M. S., Larsen, J., Strimmer, K., Christiansen, L., Hansen, K., He, L., Thibaut, L., Barão, M., Hattinger, S., Sima, V., and The, W., “The matrix cookbook,” Tech. rep.

- [49] Shuster, M. D. and Oh, S. D., “Three-axis Attitude Determination from Vector Observations,” *Journal of Guidance, Control, and Dynamics*, Vol. 4, No. 1, 1981, pp. 70–77.
- [50] NASA WV Space Grant Consortium, “EPSCoR,” 2015.