# A Unified Vision and Inertial Navigation System for Planetary Hoppers

by

Theodore J. Steiner, III
B.S. Mechanical Engineering
University of Wisconsin-Madison, 2010

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Master of Science in Aeronautics and Astronautics
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2012





Author .....
Department of Aeronautics and Astronautics
May 24, 2012

Certified by .....
Jeffrey A. Hoffman, Ph.D.
Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by .....
Scott A. Rasmussen
Senior Member of the Technical Staff
The Charles Stark Draper Laboratory, Inc.
Thesis Supervisor

Accepted by .....
Eytan H. Modiano
Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

# A Unified Vision and Inertial Navigation System for Planetary Hoppers

by

Theodore J. Steiner, III

Submitted to the Department of Aeronautics and Astronautics
on May 24, 2012, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

## Abstract

In recent years, considerable attention has been paid to hopping as a novel mode of planetary exploration. Hopping vehicles provide advantages over traditional surface exploration vehicles, such as wheeled rovers, by enabling in-situ measurements in otherwise inaccessible terrain. However, significant development over previously demonstrated vehicle navigation technologies is required to overcome the inherent challenges involved in navigating a hopping vehicle, especially in adverse terrain. While hoppers are in many ways similar to traditional landers and surface explorers, they incorporate additional, unique motions that must be accounted for beyond those of conventional planetary landing and surface navigation systems.

This thesis describes a unified vision and inertial navigation system for propulsive planetary hoppers and provides demonstration of this technology. An architecture for a navigation system specific to the motions and mission profiles of hoppers is presented, incorporating unified inertial and terrain-relative navigation solutions. A modular sensor testbed, including a stereo vision package and inertial measurement unit, was developed to act as a proof-of-concept for this navigation system architecture. The system is shown to be capable of real-time output of an accurate navigation state estimate for motions and trajectories similar to those of planetary hoppers.

Thesis Supervisor: Jeffrey A. Hoffman, Ph.D.
Title: Professor of Aeronautics and Astronautics

Thesis Supervisor: Scott A. Rasmussen
Title: Senior Member of the Technical Staff
The Charles Stark Draper Laboratory, Inc.

# Acknowledgments

The process of conducting this research and writing this thesis has been one of the most challenging and rewarding experiences of my life. I would first and foremost like to thank my advisor, Professor Hoffman, and my Draper mentor, Scott Rasmussen, for all of their support, guidance, and patience during this process. Additionally, this work definitely would not have been possible without the extensive help and guidance I received from my coworkers at Draper, especially Paul DeBitetto, Pete Lommel, Megan Mitchell, Courtney Mario, and Andrew Reiter.

I would also like to thank all of my colleagues who I have worked with on the TALARIS Project, as well as all of the many students who have contributed to the project in the past several years. I am proud to have been able to work with you all and to have contributed to this solid foundation of existing work and experience. I would especially like to thank Bobby Cohanim and Phillip Cunio for all of the helpful mentoring they have provided me over the past two years, and to also make a special thank you to my fellow graduate students in our little TALARIS cohort, Farah Alibay, Joe Morrow, and Chris Rossi, for always keeping work and life enjoyable. Few students are given the opportunity to work with and be mentored by so many wonderful people, and for that I feel extremely fortunate.

And last but not least, I would also like to thank my loving family and friends, who have always supported me in everything I have done and in every time of need.

# Assignment

Draper Laboratory Report Number T-1720

In consideration of the research opportunity and permission to prepare this thesis by and at the Charles Stark Draper Laboratory, Inc., the author hereby assigns copyright of the thesis to The Charles Stark Draper Laboratory, Inc., Cambridge, Massachusetts and the Massachusetts Institute of Technology. The author hereby grants to MIT and Draper Laboratory permission to reproduce and distribute publicly paper and electronic copies of this thesis document in whole or in part.

Theodore J. Steiner, III

# Contents

# Chapter 1

# Introduction

Several general trends have surfaced over the past several decades of planetary exploration. One of these is that the scientific goals of exploration programs have grown increasingly targeted. Initial planetary exploration programs, such as Pioneer, Mariner, and Voyager, consisted mainly of planetary flybys and imprecise impactors. Trajectories and landing regions of more recent missions, such as Mars Pathfinder and Phoenix, have grown more precise, encouraging scientists to ask more specific questions about our neighbors in the solar system. Research questions have evolved over time, growing both more complex and specific, from the initial question of "What does the surface of the Moon look like?" (answered by the Lunar Ranger program), to the more recent, "Is there water on the Moon?" (answered by the LCROSS mission), to the currently unknown, "Is there water ice in the permanently shadowed regions of the Shackleton crater near the south pole of the Moon?" To accommodate this trend, the landing precisions of planetary landers have increased with time, as shown in Table 1.1.

A second, and related, trend is towards increased surface mobility upon landing on other planets. The majority of surface landers have been stationary (Viking, Phoenix, Surveyor, Venera, etc.). While returning a wealth of scientific data, stationary landers are limited to only one location: where the vehicle lands. Many of the recent Mars exploration missions have addressed this limitation with mobile rovers, which are

| Mission | Launch Year | Planetary Body | Landing Ellipse[1] (km) |
|---|---|---|---|
| Surveyor | 1967 | Moon | 20 x 20 |
| Apollo | 1969 | Moon | 1 x 1 |
| Pathfinder | 1996 | Mars | 100 x 200 |
| Mars Exploration Rovers | 2003 | Mars | 87 x 11 |
| Phoenix | 2007 | Mars | 100 x 19 |
| Mars Science Lab | 2011 | Mars | 20 x 25 |

Table 1.1: Approximate landing precisions of various planetary landers.

capable of sampling multiple locations near the landing site, greatly increasing both the effective landing precision of the spacecraft and the diversity of the data returned. These rovers have traditionally been limited by their low velocities to relatively short ranges over their lifespans, as shown in Table 1.2. A significant limitation on a rover's average speed is the capability of its onboard navigation system, as well as other factors such as power, traction, and hazard avoidance.

In recent years, considerable attention has been paid to planetary hoppers for their potential to overcome the limitations on landing precision and mobility facing current planetary surface exploration technologies. Planetary hoppers have been shown to enable a wide range of planetary exploration missions of value [1]. This thesis describes the architecture, design, and preliminary testing of a hopper navigation system, intended for use with continued Earth-based hopper testing and as a basis for flight hardware for future planetary missions.

| Rover Name | Country | Launch Year | Speed[2] (m/s) | Distance[3] (km) |
|---|---|---|---|---|
| Lunokhod 2 | Soviet Union | 1967 | 0.55 (max) | 37 |
| Sojourner | United States | 2996 | 0.007 (max) | 0.1 |
| MER Spirit | United States | 2003 | 0.01 (avg) | 7.73 |
| MRE Opportunity | United States | 2007 | 0.01 (avg) | 34.36 |
| MSL Curiosity | United States | 2011 (launched) | 0.008 (avg) | 20 |
| Chandrayaan 2 | India & Russia | 2014 (proposed) | 0.1 (max) | 150 |
| ExoMars | Europe | 2018 (proposed) | 0.001 (avg) | 22 |

Table 1.2: Past and future rovers and their average speeds.

[1]The landing ellipses for landing on Mars are significantly larger than those for landing on the Moon primarily due to uncertainties and challenges associated with atmospheric effects.

## 1.1 Planetary Hoppers

Planetary hoppers are vehicles that traverse planetary surfaces using chemical exhaust propulsion alone, freeing them from many of the limitations of rovers and stationary landers. This allows hoppers to fine-tune their landing sites to very high levels of precision, while also allowing exploration of a wide range of otherwise inaccessible terrain. For this reason, analogies such as "reusable landers" and "airless helicopters" are sometimes used to describe the unique mission profiles they enable. An example of a conceptual hopper is shown in Figure 1-1.

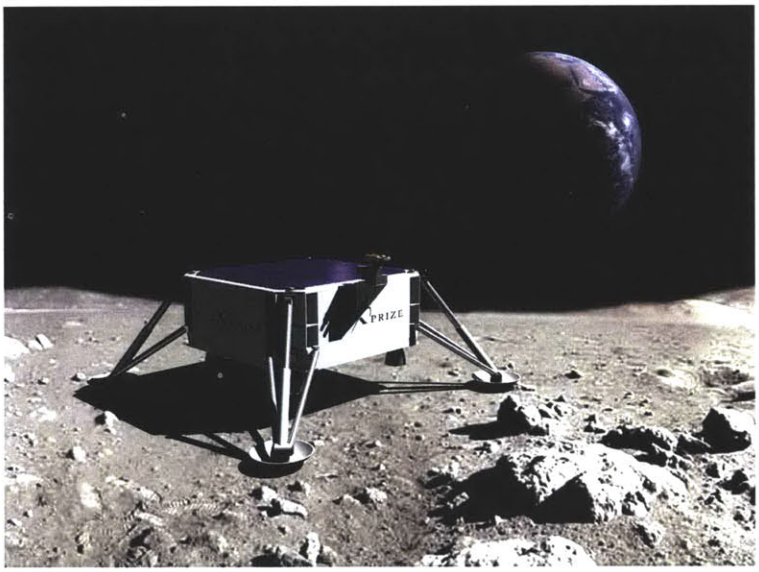



Figure 1-1: A conceptual hopper designed for use on the moon. Image credit: Draper Laboratory/Next Giant Leap

Hoppers provide a number of unique advantages over other types planetary surface exploration vehicles. Cunio, et al. [2] list five basic advantages of hoppers, which are summarized below.

[2]Rover speed and total distance is a mission parameter heavily affected by multiple mission goals and requirements, as well as the technological capability of the launch organization, but it is clear that rovers are growing more capable with each design iteration.

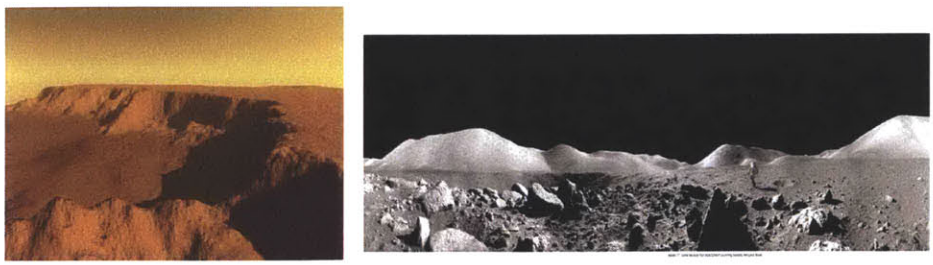
[3]Distances given for MSL, Chandrayaan, and ExoMars are success criteria for the mission, and are not necessarily comparable with those of the operational rovers. For comparison, the success criteria for the MER rovers was only 600 meters.

### Advantage 1–Rapid Regional Coverage

Hoppers can cover much greater distances than rovers or vehicles requiring a base station for power or other infrastructure. The Mars Exploration Rover Opportunity has traveled nearly 35 km during its 8 years thus far on the surface of Mars. However, a hopper could cover a comparable distance in a few minutes, if desired, and then perform additional hops to different locations. This allows rapid exploration of multiple sites of scientific interest, as well as close-range imaging throughout the trajectory.

### Advantage 2–Access to sites in rough terrain

Terrain limitations for exploration vehicles are severe. Currently, landing sites are selected largely based off the absence of hazardous obstacles. Actual rover speeds are much lower than their true maximum speeds in part due to terrain hazards, and the regions in which rovers can operate are limited by the range of slopes and surface features they are capable of handling. For example, the Mars Science Lab Curiosity rover can traverse over obstacles up to 75 cm high, but no larger. Hoppers do not have such limitations, as they traverse above the planetary surface, allowing them to traverse over cliffs or canyons, explore inside of craters and caves, and pass over especially rocky or uneven terrain, as shown in Figure 1-2. The only terrain constraint for a hopper is the requirement for the presence of a suitable landing site that can fit its small footprint somewhere near the end of its trajectory.



(a) Cliffs (b) Rocky Terrain

Figure 1-2: Hoppers are capable of exploring hazardous terrain otherwise inaccessible using wheeled rovers.

### Advantage 3–Ultraprecise landing capability

Hoppers are less reliant on entry, descent, and landing (EDL) system performance to achieve precision landings, as they can simply perform a higher precision, corrective hop upon landing, as shown in Figure 1-3. This technique, called ultraprecision, allows for landing accuracies on the order of a meter. A hopper can utilize less complex, flight-proven entry and descent techniques without the need for high-precision guidance systems during these stages, reducing overall system cost and mass. This also allows for selection of safer initial landing zones further from desired scientific destinations.

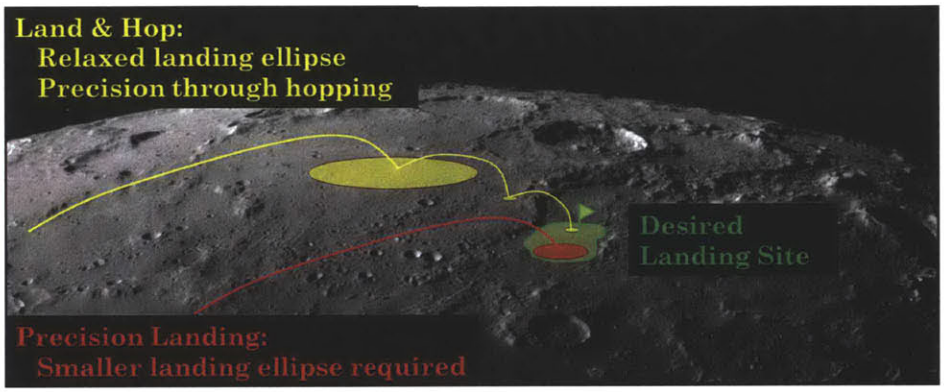



Figure 1-3: Hoppers can achieve ultraprecise placement through multiple hops.

### Advantage 4–Reduced system development cost

Hoppers are already designed for landing, so additional landing hardware development is not required. On airless bodies, the hopper can use its onboard primary thrusters for descent maneuvers and vehicle control. In the case of atmospheric bodies, the hardware capabilities already onboard a hopper could be used to augment a parachute system to provide a low- or zero-velocity touchdown utilizing Hazard Detection and Avoidance (HDA) systems, allowing a controlled landing while still utilizing heritage EDL technologies. Additionally, hoppers do not require any complex surface deployment systems, and all of their landed mass is part of the primary vehicle or payload. This is in contrast to systems such as Mars Pathfinder's airbags or the Mars Science

Lab's "Skycrane," which both require additional mass for terminal descent, landing, and deployment systems.

### Advantage 5–Shifted operational complexity and risk

Most mobile surface exploration systems require some form of deployment, which increases mission complexity and risk. However, there is generally no need to deploy the hopper after initial touchdown because a hopper is inherently a lander. As a tradeoff, a hopper has additional complexity and risk related to its guidance, navigation, and control (GN&C) systems. Whereas rovers can almost always stop their motion at any time without risk to allow additional time for decision-making, hoppers have a set start and finish location for each motion and must operate reliably in real-time, as all hesitations translate into additional fuel cost. This offers organizations and mission designers the opportunity to shift the complexity and risk of deployment toward other systems, such as GN&C. This is potentially advantageous because it removes an entire system from the vehicle, saving mass, development cost, and mission risk, at the expense of requiring another system to be designed more robustly. This also allows a shift in mission requirements, as precision navigation can occur during surface operations as opposed to during landing from orbit.

### Disadvantages of hoppers

The primary disadvantage of hoppers is that they are fuel limited, as they use chemical propulsion rather than electrical drive motors. This means that a hopper must bring all of its fuel with it from Earth, or utilize some form of in situ resource utilization (ISRU), whereas rovers can use solar panels or radioisotope thermoelectric generators (RTGs) as a long-term power source. Once a hopper expends its propellant reserves, its capability becomes roughly equivalent to a stationary lander. A second disadvantage is the high requirement for autonomy in the GN&C system. Rovers can stop and wait for human operator instructions at any time it becomes necessary. However, be-

cause of communication delays and their high rates of motion, hoppers must operate entirely autonomously. This puts stricter requirements on the navigation system for a hopping vehicle than for a rover for both performance and reliability.

## 1.2 Problem and Motivation

Extensive development of hopping vehicles has been conducted by the Terrestrial Artificial Lunar And Reduced gravIty Simulator (TALARIS) Project, which is a joint effort of the MIT Space Systems Lab and Charles Stark Draper Laboratory to develop a prototype lunar hopper and to mature planetary hopping technologies. While the TALARIS Project has made significant progress towards demonstrating the viability of hopping for planetary surface exploration, an operational hopper navigation[4] system does not yet exist. This thesis describes the architecture, design, and preliminary testing of a hopper navigation system, intended for use with continued Earth-based hopper testing and as a basis for actual flight hardware.

Hoppers require a fully autonomous, internal navigation system capable of handling rapid, near-surface motions in an unknown environment. Autonomy is required due to the long communication delays to the Moon or other planets, which eliminate the possibility of remote operation due to the rapidity of hopper motions. The system must be entirely internal and self-contained because installing an external navigation system (e.g., a GPS-like system) on another planetary body is prohibitively expensive. The system must be capable of navigating in an unknown environment to achieve terrain freedom, as a hopper might be called upon to explore areas unavailable from orbital imagery, such as permanently shadowed craters or underneath overhangs of

[4]Guidance, navigation, and control are three distinct tasks often considered as a single vehicle subsystem, called "GN&C." Navigation determines the current state of the vehicle (its location and attitude, collectively referred to as its "pose," and its velocity), guidance determines what the vehicle must do to reach its target, and control translates commands from guidance into physical actions and maintains the stability of the vehicle. Note that while the term "navigation" is colloquially typically used to refer to both navigation and guidance (such as in the case of a "GPS navigation system" for a car), in this thesis, the term refers solely to the determination of the state of the vehicle. This information then becomes the input to the guidance system.

cliffs. Additionally, the precision required of hopper trajectories surpasses the resolution currently available from orbital imagery and digital elevation maps – the primary sources of external information available.

The navigation systems developed for traditional exploration vehicles cannot meet these requirements. Navigation systems onboard prototype hoppers currently in development for testing on Earth are typically dependent on either an external system, such as the Global Positioning System (GPS), or prior knowledge of their environment. Traditional terminal-descent and landing navigation systems are not designed for extensive near-surface operation or high-rate translational motion. Helicopter navigation systems are capable of handling translational motion, but are generally dependent on GPS or other external systems, such as a remote operator.

## 1.3 Project Goals

The purpose of this work is to design, develop, and test a prototype hopper navigation system. This can be broken down into four primary goals:

1. Develop a system architecture for a navigation system capable of estimating the state of a hopping vehicle. (Addressed in Chapter 2.)
2. Develop a prototype system to evaluate the proposed hopper navigation system architecture and its capabilities. (Addressed in Chapter 3.)
3. Demonstrate that unifying inertial and relative navigation techniques provides advantages applicable to hoppers. (Addressed in Chapter 4.)
4. Evaluate the performance of the prototype system and demonstrate hopper navigation capabilities. (Addressed in Chapter 5.)

## 1.4 Literature Review

Although a hopper-specific navigation system has not been previously developed, there is a large amount of existing work on both hopping and navigation systems to build upon. This brief literature review provides some background and a summary of previous work.

### 1.4.1 Hopping

Hoppers are a relatively new form of planetary exploration vehicle, but their high promise has motivated a large amount of research in recent years, especially for use in low-gravity and low-atmosphere environments. In the early 1990s, ballistic hopping vehicles were proposed as a method of surface mobility for human explorers on Mars [3]. These vehicles were very large, required development in many areas, such as in situ resource utilization (ISRU), and would have been difficult to certify as safe for human passengers. Small hoppers for unmanned missions, however, do not share these drawbacks. Several studies of unmanned Mars exploration missions conducted by multiple organizations have included hopping vehicles [4–8]. All of these hopper concepts included some form of ISRU, typically based around extraction of CO2 from the Martian atmosphere. Most of the work done in relation to these conceptual hoppers focused on development of ISRU systems of varying complexity, some even involving prototype testing [7]. None of these early-stage development projects involved navigation of these vehicles, and none of the concepts were selected as candidates for NASA or European Space Agency (ESA) exploration missions.

In 2007, the Google Lunar X-Prize (GLXP) was announced, offering a prize of $20 million to the first non-government organization to land on the Moon and travel at least 500 meters on the surface. Shortly afterwards, team Next Giant Leap (NGL) was formed, including MIT and the Draper Laboratory amongst its team members. Architecture studies based around the framework of the GLXP resulted in the selection of a hopper as the ideal surface mobility system for the competition, as it

offers potential for significant reductions in both launch mass and overall mission risk [9]. Because hoppers can bring a limited amount of fuel onboard, a small hopper, as shown in Figure 1-4, can easily achieve the GLXP requirements without the need for complex ISRU systems.


Figure 1-4: Artist's rendition of the Next Giant Leap hopper designed to compete in the Google Lunar X-Prize. Image credit: Draper Laboratory/Next Giant Leap

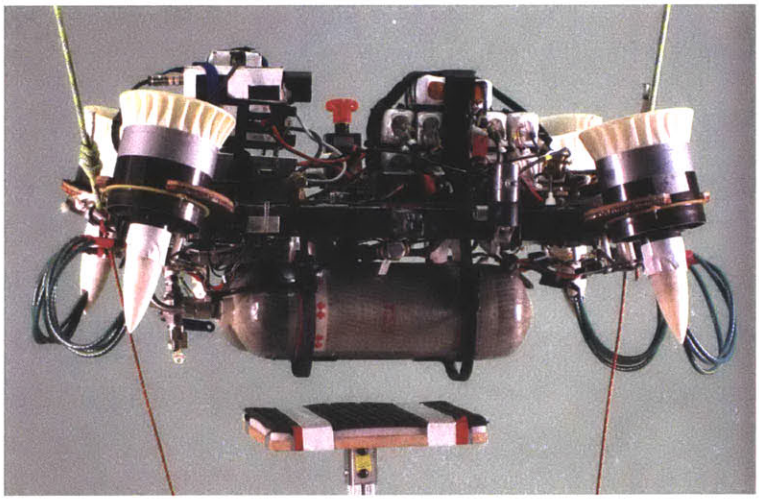


Figure 1-5: The TALARIS Prototype hopper, developed by the Draper Laboratory and MIT, is used to test hopping guidance, navigation, and control algorithms on Earth. Four electric, ducted fans are used to reduce the effect of gravity and simulate the lunar environment.

The Talaris Project, the first comprehensive research program relating to hopper technology development and detailed mission analysis, was formed as part of NGL's GLXP efforts in order to advance hoppers to flight-ready status. As part of this project, the

TALARIS prototype hopper, shown in Figure 1-5, was designed, constructed, and flown in order to gain operational experience [10–12] and validate hopping GN&C algorithms developed by Draper [13, 14]. Focuses of technology development research by the Talaris Project have included topics such as hopper trajectories [15], avionics systems [16, 17], and propulsion systems [18–20]. Additionally, considerable effort was made to define when and where hoppers make sense for planetary exploration [1,2], including point designs for hoppers intended for exploration of the Moon [9], Mars [21], and Titan [22]. Current research of the Talaris Project is focused primarily on the development of a hopper navigation system [23] and a hazard detection and avoidance (HDA) system [24].

The Draper Laboratory-built GENIE (Guidance Embedded Navigator Integration Environment) system is a reliable and robust platform capable of testing GN&C algorithms onboard terrestrial rockets, such as the Masten Xombie [25]. On February 2, 2012, a derivative of the TALARIS GN&C software was flown onboard GENIE and Xombie, and successfully demonstrated a 50 m free-flight, rocket-powered hop, shown in Figure 1-6 [26]. However, GENIE is currently dependent on GPS for its navigation. Thus, all primary systems unique to planetary hoppers have been demonstrated in some form, with the exception of the navigation and HDA systems. This thesis provides the design of such a hopper navigation system, as well as significant progress towards its demonstration.

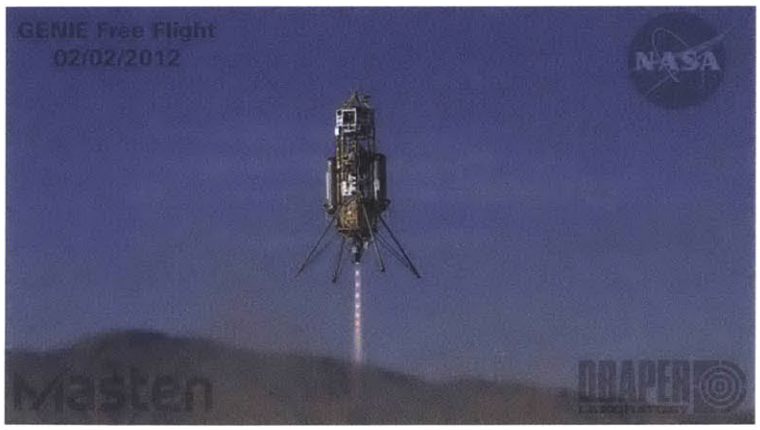



Figure 1-6: Draper's GENIE System guides the Masten Xombie terrestrial rocket during a controlled hop. Image credit: Draper Laboratory

### 1.4.2 Existing Navigation Systems

A hopper navigation system is in many ways a combination of navigation systems designed for planetary landing, unmanned aerial vehicles (UAVs), especially helicopters, and surface vehicles, such as cars or pedestrians. For this reason, a survey of capabilities and existing systems available in these adjacent fields was required.

#### Planetary Lander Navigation

Traditionally, unmanned planetary landers have used one or more inertial measurement units (IMUs) and a Doppler radar system as their primary landing navigation sensors. However, Doppler radars are large, heavy, and expensive, and are growing increasingly so as the desired landing ellipses (a measurement of landing precision) grow increasingly small. For example, the Doppler radar system for the Mars Science Lab (launched November 2011), shown in Figure 1-7, approaches the size and mass of the entire NGL hopper.

Figure 1-7: Testing an engineering model of the Mars Science Lab Doppler RADAR system, which is used for descent navigation. Image credit: NASA

To counter this trend toward prohibitively large and expensive landing navigation sensors, the vast majority of landing navigation systems currently in development

include some form of optical sensor, such as a camera, flash Light Detection and Ranging system (LIDAR), or Doppler LIDAR, typically in addition to an IMU. Optical systems are capable of determining more information at higher precision and for lower mass than Doppler radar regarding a vehicle's position and attitude (i.e., its pose), but have higher algorithmic complexity. In addition, passive optical sensors (i.e., cameras) are limited by external conditions, such as surface illumination and texture.

The NASA Autonomous Precision Landing and Hazard Avoidance Technology (ALHAT) program is currently developing a navigation system intended for manned lunar landings [25, 27, 28]. This system uses a flash LIDAR, Doppler LIDAR, passive optical cameras, IMUs, and a laser altimeter. This system will be highly reliable and redundant, with a required landing accuracy of 90 m or better, at the cost of being large, heavy, and expensive. A flight test of this system using a helicopter is shown in Figure 1-8, which shows its large size (for the prototype system). This system is not a good match for small, lightweight, unmanned vehicles, such as hoppers, which do not require the high level of redundancy and accuracy during descent from orbit of manned vehicles. The ALHAT system uses both active and passive optical sensors for terrain relative navigation (TRN) and hazard detection and avoidance (HDA). These TRN algorithms match features, such as craters, and elevation measurements to an a priori reference map of the landing region, and are partly based on navigation systems already in use on cruise missiles [27, 28].

The NASA Jet Propulsion Laboratory (JPL) is currently developing the Vision-aided Inertial Navigation (VISINAV) system for unmanned planetary descent and landing. The VISINAV system uses a single optical camera and an IMU, tightly integrated using an Extended Kalman Filter (EKF) to achieve precision landing accuracy. Two vision algorithms run onboard the system. The first extracts landmarks from images and localizes them to an a priori map of known landmarks in the landing region, providing absolute positioning information for the vehicle. The second tracks the motion of features (not necessarily known a priori) to provide additional relative

Figure 1-8: A flight test of the NASA ALHAT navigation system. Image credit: NASA

pose information, especially at low altitudes when known landmarks become sparse. An experimental test of this system, deployed from a sounding rocket, achieved an accuracy of 6.4 m in determining landing position [29].

The ESA Navigation for Planetary Approach and Landing (NPAL) program [30], and its follow-up navigation system, VisNAV [31], are also developing a descent and landing navigation system for unmanned planetary missions, primarily intended for the Moon and Mars. Unlike the JPL VISINAV, which localizes landmarks to a database, ESA's VisNAV determines the distances to extracted feature points from images and matches them to a digital elevation map (DEM) of the landing region. VisNAV also includes a purely relative vision navigation system to track the relative motion of the vehicle to unmapped features to improve upon the resolution available from the DEMs, as well as an IMU and EKF [32].

A number of other groups have begun development of vision-based navigation systems, although most have not yet approached the level of maturity of ALHAT, VISINAV, or VisNAV. The Draper Laboratory has developed a vision-based navigation system for lunar descent that uses absolute positioning of craters on an a priori crater map, as well as a feature tracking algorithm for relative positioning. The outputs of these two algorithms are combined using an EKF [33]. Additionally, researchers at the Deep Space Exploration Research Center in Harbin, China have also developed a combined

vision and inertial navigation system for Martian landing that localizes detected image features on an a priori map [34]. These systems show promising results in simulation, but results from hardware tests are not yet published.

### Unmanned Aerial Vehicle Navigation

With the growing prevalence of Unmanned Aerial Vehicles (UAVs) for both military and commercial applications, much research has focused on development of their navigation systems. Especially at low altitudes and with unmanned rotorcraft, these navigation systems fulfill similar requirements to those of hoppers.

Most UAV navigation systems include a GPS receiver, often integrated with an IMU using a Kalman filter. An example of recent work in this area involved the addition of a magnetometer to a GPS/IMU/EKF navigation system [35]. The magnetometer is able to provide observability into the yaw gyro of the IMU (i.e., normal to the surface of the Earth), which is otherwise very difficult to calibrate (because gravity does not project into yaw motions). Experimental results of this system onboard a miniature unmanned helicopter showed that the addition of the magnetometer was beneficial, especially to yaw measurement [36]. However, the vast majority of these systems include sensors unavailable to hoppers (especially in lunar environments), such as GPS receivers, magnetometers, and barometers. Additionally, many UAVs have a human operator in the loop, and therefore have not had a need for purely autonomous navigation and real-time decision-making.

Inertial navigation typically requires an additional sensor to constrain its error growth, such as a GPS receiver. However, GPS is not always available or sufficiently accurate, so vision systems have made their way onboard UAVs to further improve their accuracy and robustness. Early vision systems typically involved ground-based data processing, as hardware capable of providing the computing power required was too large for flight systems [37]. As processors grew smaller and faster with time, these systems were moved onboard UAVs, providing real-time pose updates [38].

Follow-on work by groups such as the Draper Laboratory [39] and Georgia Tech [40] led to systems capable of navigating during limited periods of GPS dropout. These systems allow small UAVs to operate in areas with sparse GPS coverage, such as urban canyons. More recent work has led to UAV navigation systems capable of operation in the absence of GPS entirely [41–43]. However, many inertial and vision-only systems still use an additional sensor, such as a barometer [44,45], to more robustly constrain altitude measurements.

### Surface Navigation

A third class of navigation systems can be formed from the many systems under development for use by pedestrians, cars, and planetary rovers. Because the pedestrians or vehicles requiring navigation are known to be on a fixed surface, the primary roles of these systems tend to be odometry measurements and hazard detection rather than attitude and altitude.

One such pedestrian system is the Draper Laboratory's Personal Navigator System (PNS), which is a wearable navigation system using an IMU, GPS receiver, barometric altimeter, and a three-axis miniature Doppler radar. Experimental results showed this system to be highly effective, but the radar was critical in order to navigate during GPS outages [46].

A number of ground vehicle navigation systems have been designed and experimentally tested, primarily in cars. The Carnegie Mellon University Robotics Institute developed a system robust to GPS outages using an IMU and a stereo camera, which is capable of accurate navigation of a passenger car over distances of several kilometers [47]. Similar systems have been developed by Oak Ridge National Lab using a monocular camera [48], and by a group from South Korea using an omnidirectional camera [49].

Development of navigation systems for planetary surface rovers is also an active area of research, as NASA, ESA, China, India, and many GLXP teams all have plans to

send mobile vehicles to the surface of the Moon or Mars within the next decade. NASA's JPL developed a combination stereo vision and IMU system for odometry measurements on their Mars rovers as a means to detect and overcome errors from wheel rate measurements due to slippage and poor traction [50]. Recent developments at JPL have led to the AutoNAV system, which enables real-time navigation and hazard avoidance onboard future Mars rovers and has been successfully tested in analogous environments [51]. A group in China developed a system to navigate an unmanned lunar rover using stereo and monocular vision, which is also capable of obstacle detection and avoidance, and has been demonstrated in simulations [52]. A number of other navigation algorithms for lunar navigation have been proposed, such as one that tracks the positions of mountain peaks on the surrounding horizon to localize a rover to within approximately 100 m on the lunar surface [53].

### 1.4.3 Navigation Algorithms

The trend in lander, UAV, and surface navigation research and development is towards replacement of large or high-cost sensors with small, low-cost vision systems. These vision systems are frequently combined with inertial sensors. This section briefly describes some of the stereo and monocular vision algorithms in the literature, as well as some examples of algorithms incorporating both vision and inertial sensor data.

#### Stereo Vision

Detected features in an image can be localized in space in only two dimensions, as a camera image is only two dimensional. Stereo vision systems use the disparity between images from two adjacent cameras to calculate distances to detected features within a given scene [54]. With this additional information, features can be localized in 3D space [55]. Once the locations of these features are known, pose algorithms can calculate the location of the camera with respect to these features. Motion of the camera can be detected by repeating this process over multiple frames, as was done

on the early Mars rovers [50].

This process can be taken a step further by utilizing a class of algorithms called Simultaneous Localization and Mapping (SLAM). SLAM adds an element of memory to the system, as all of the features localized in space are saved and built into a map of the vehicle's surroundings. The camera is then localized with respect to the map, rather than the series of images themselves, leading to a more robust solution. A number of stereo implementations of SLAM have been applied and experimentally tested on surface vehicles [56, 57]. One stereo SLAM implementation of particular interest to this thesis is the RSLAM system, developed by the University of Oxford [58]. This system achieves particularly strong results even in the presence of changing lighting conditions and rapid motions. The RSLAM system provides the basis from which the Draper-developed DSLAM, introduced in Chapter 4, system was built upon. Other recent work relating to stereo SLAM has related to increasing the computational speed of the algorithms, such as by running portions on highly parallel hardware architectures such as an image processing LSI [59], FPGA, or GPU [60].

### Monocular Vision

Stereo vision systems require two cameras spaced a specified distance apart, which is directly proportional to the maximum depth they can resolve. Single-camera, or monocular, vision systems have been developed in an attempt to reduce size and mass, especially for small surface robots and UAVs. These systems typically work in one of three ways. First, the camera can extract information from the environment, such as vertical and horizontal lines and their perspectives, and use them to estimate the camera's location [61,62]. However, this will not work in an unstructured environment such as the Moon.

A second monocular technique, called optical flow, tracks the motion of features within the frame of the camera. From this motion, the algorithm can determine information about the distance the camera has moved and its velocity. An example of an optical

flow system is the Mars Descent Imager, which was flown on the Mars Exploration Rovers during their descent in order to estimate their horizontal motion [63].

The third commonly used class of monocular vision algorithms is called structure from motion. This is similar to iterating upon an optical flow system, in which the relative motions of tracked features are used to determine their depths, from which the pose and motion of the camera can be determined. These types of systems require the camera to be in motion, are more computationally intensive, and have difficulty resolving depth without the aid of an additional sensor, such as a laser rangefinder or barometric altimeter. A number of SLAM implementations using monocular structure from motion are available in the literature [64–66].

### Unified Vision and Inertial

Performance improvements can come from the inclusion of multiple sensor systems using a statistical estimator, such as an Extended Kalman Filter (EKF). Multiple sensors are typically chosen to balance their strengths and weaknesses, leading to a more robust integrated system. A common combination is a vision system and an IMU, and many such systems are available in the literature using either stereo [23, 67–69] or monocular vision systems [70–73]. A detailed discussion of the advantages provided by such a unified system is provided in Chapter 4.

# Chapter 2

# Hopper Navigation

Hopping is an advantageous new means of planetary exploration. Dedicated navigation systems for vehicles similar to hoppers, such as landers, unmanned aerial vehicles (UAVs), and wheeled surface vehicles currently exist and have been experimentally tested. However, no such dedicated system has yet been developed for planetary hoppers. Development of an autonomous, real-time, precision navigation system specific to hoppers is needed prior to deployment of operational planetary hoppers.

This chapter begins by describing the flight profile of hopping trajectories and providing an overview of the challenges a hopper navigation system must overcome. These challenges are then rephrased as goals for a hopper navigation system, and a system architecture capable of meeting those goals is proposed. Subsequent chapters describe the design of a hardware and software testbed to experimentally test this architecture (Chapter 3), the advantages of the primary navigation algorithm selected for this architecture (Chapter 4), and an analysis of experimental results obtained using the testbed system (Chapter 5). Chapter 6 provides high-level conclusions on the suitability of this architecture for operational unmanned planetary hopping vehicles.

## 2.1 Hopper Trajectories

Three types of hopper trajectories have been proposed in the literature: the ballistic hop, hover hop, and gliding hop. The navigation system architecture described in this thesis was designed primarily for the hover hop trajectory. However, this architecture is expected to be additionally capable of navigating a hopper along other trajectories, though possibly less precisely or optimally than a system designed specifically for those trajectories.

### 2.1.1 The Ballistic Hop

The ballistic hop is usually the most fuel-efficient of the three, especially for long distances. It is also the most broadly proposed in hopping literature [3–6, 8]. As described by Middleton, the ballistic hop begins with a delta-V maneuver at a nonzero angle relative to the surface, placing it into a ballistic coast. When the vehicle approaches the landing site, it performs a braking maneuver in order to achieve a soft touchdown [15]. This trajectory's main drawbacks are that the vehicle must reorient itself at the apex of the trajectory and that the vehicle has large attitude changes during the flight. The upper portion of Figure 2-1 shows a sample 1 km ballistic hop trajectory. Due to their fuel-efficiency, ballistic hops are typically proposed for long-distance hops.

### 2.1.2 The Hover Hop

The second hopper trajectory is called the hover hop, during which the vehicle maintains constant attitude throughout the duration of the flight. The hover hop begins with a vertical ascent to a specified hover altitude, followed by a constant-attitude lateral traverse, and finishes with a vertical descent and soft landing. A detailed analysis of the hover hop trajectory is provided by Middleton [15].

Draper and MIT selected the hover hop for their TALARIS Prototype and Google

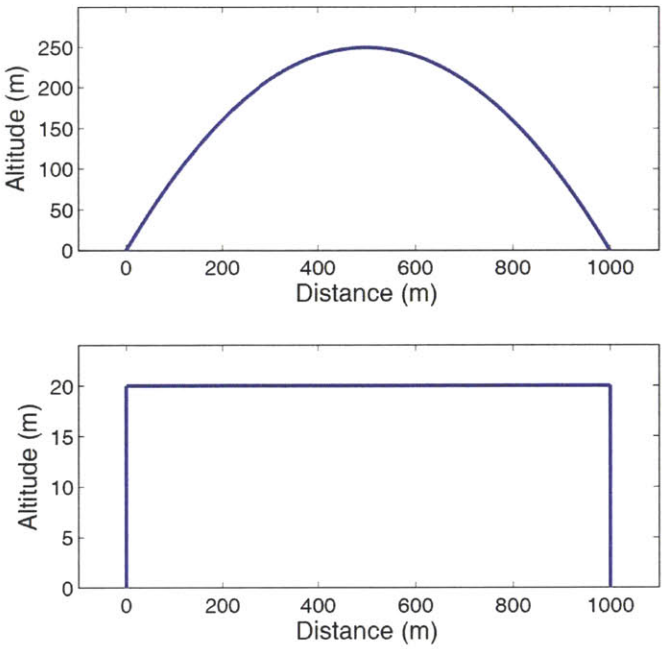



Figure 2-1: Sample 1 km ballistic hop (above) and hover hop (below) trajectories.

Lunar X-Prize (GLXP) hoppers for the many advantages it provides, including maintaining constant attitude (beneficial for safety and possible payload scientific measurements) and decoupling the vertical and lateral control systems, which simplifies their design. A sample 1 km hover hop trajectory is shown in the lower portion of Figure 2-1.

A hover hop includes seven distinct flight modes, as shown in Figure 2-2. A hopper navigation system must be capable of navigating during all of these flight modes, though each mode has unique priorities. During the first flight mode, the system must correctly measure altitude and heading angle (roll) during the ascent. During the stationary hovers, modes 2 and 6, the system will focus on attitude measurement, as the vehicle must precisely control its attitude prior to firing its lateral thrusters in order to reduce the need for corrective action later in the trajectory. During the lateral modes, the system must continue to provide accurate attitude measurements, but also must measure the distance the vehicle has traveled (i.e., its odometry). Vehicle motion occurs only along one axis of the vehicle at any given time, with the exception

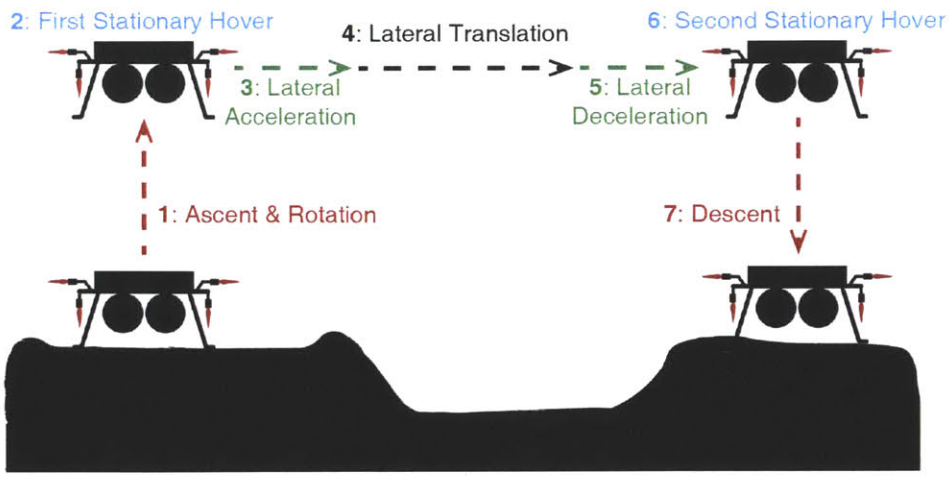



Figure 2-2: Hover hop flight modes.

of erroneous motions, which are small. This allows for an effective decoupling of the navigation system, in which the vertical and lateral motions can be considered and designed for independently.

### 2.1.3 The Gliding Hop

The gliding hop is a modified hover hop for use on planets or moons with dense atmospheres, such as Venus, Titan, or the Earth, which requires the addition of rigid wings to the vehicle [22]. This trajectory starts with a vertical ascent to a specified altitude, similar to the hover hop. The vehicle then conducts a downward pitching maneuver to gain velocity and then enters a downward sloping trajectory at its maximum lift to drag ratio. When the vehicle has lost most of its altitude, it pitches upward, shedding velocity until it stalls. At that point, the vertical actuators re-engage and the vehicle performs a vertical descent maneuver, similar to that which concludes the hover hop. A sample 850 m hover hop trajectory designed for Titan is shown in Figure 2-3.

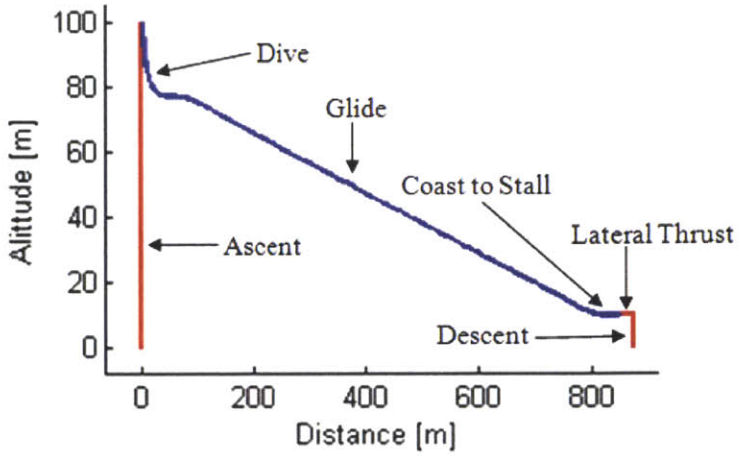



Figure 2-3: A sample 850 m gliding hop trajectory. Powered segments of the trajectory are shown in red and and unpowered are shown in blue.

## 2.2 Challenges of Hopper Navigation

Hoppers are often referred to as "reusable landers" or "airless helicopters." However, these analogies do not translate especially well to hopper navigation systems, due to several additional challenges beyond those typical for landers or helicopters. Challenges beyond those typical for landing are largely a result of the hopper executing its mission in close proximity to the planetary surface and in poorly mapped environments. Challenges beyond those typical for UAVs arise because fewer sensors are available for use, all computation must be performed onboard the vehicle in the absence of a human operator, the operational environments are less understood than those on Earth, and vehicle fuel-constraints result in an unrecoverable cost for any hesitations or corrective actions required during flight. The four key challenges of hopper navigation are: mass and power limitations, hazard detection and avoidance, terrain variability, and rapid motion handling. A successful hopper navigation system must be capable of overcoming all of these challenges.

### Mass and Power Limitations

Landers must be able to determine their location over a large variation in altitudes throughout their descent, but only have to land once. Because hoppers make repeated landings, as well as lift-offs and traverses, the mass of every onboard system carries a much higher penalty as far as fuel requirements. For this reason, it is important that the mass of the navigation system (and by extension, its power requirements) be kept to a minimum. UAV navigation systems face similar challenges, but often use additional sensors unavailable to hoppers, such as lightweight barometers or GPS receivers. These sensors significantly relax the requirements on the primary sensors, allowing the use of smaller and lighter cameras and IMUs.

### Hazard Detection and Avoidance

Hoppers have stricter requirements than landers for hazard detection and avoidance (HDA) systems. Because hoppers generally remain near to the surface during traverses, the HDA system must scan the region in front of the vehicle as well as that below it for obstacles, unlike landers, which typically only scan in one direction. Furthermore, the HDA system has significantly less time to identify and act on hazards, such as rocky areas or cliff faces, because its descent time is on the order of seconds rather than minutes. Additionally, a soft, safe touchdown on the planetary surface is a critical requirement for hoppers, as the landing gear must not be damaged prior to mission completion. Most planetary landers utilize one-time-use crushable structures for kinetic energy damping on landing. Most operational UAVs are designed to land on landing pads or runways and operate in open airspace, and thus do not typically share these strict requirements for HDA.

### Terrain Variability

Hoppers must be able to operate in environments with greater terrain uncertainty than landers. Whereas a lander typically has a designated landing region with consis-

tent, well-understood terrain, an often touted advantage of hoppers is their "terrain freedom," which allows them to explore multiple types of terrain on a single mission. This means that any surface-relative navigation system must be terrain-independent, allowing for greater mission versatility and flexibility. This is increasingly becoming the case with UAVs as well, particularly with small rotorcraft designed to operate near the surface. However, this is not as strict a requirement with UAVs, as position can be constrained by GPS when it is available, as well as other sensors.

#### Rapid Motion Handling

Hoppers must be capable of handling the rapid motions of repeated trajectory adjustments and control actions performed using small rockets. This is largely driven by the HDA requirements and the short time scales allowed for avoidance maneuvers. This is not as strict a requirement for landers, as they can begin scanning the landing region for safe zones long before they enter the flight envelope of a hopper, allowing for more gradual maneuvers initiated earlier on in the trajectory. Because UAVs are not as strictly concerned with propellant burn time and can typically be refueled and reflown, they do not necessarily require the same extent of rapid motion handling capability. As UAV missions become increasingly complex, this will likely become a stricter requirement.

## 2.3 Hopper Navigation System Goals

The aforementioned hover hop trajectory and challenges of hopping lead to a number of important goals for any hopper navigation system. These goals could loosely be considered to be stakeholder expectations for the system. In this context, the system includes both pose estimation and hazard detection functions, as they will likely share sensor systems and should be considered collectively for purposes of architecture selection. The remainder of this thesis addresses only the pose estimation and velocity and angular rate measurement functionality of the system (i.e., "navigation").

The primary goal of the pose estimation system is to provide an accurate six degree-of-freedom (DOF) pose estimate at a sufficient rate to the hopper guidance and control systems during all phases of flight. During vertical ascent maneuvers, the system must be capable of precisely measuring both roll[1] (i.e., "heading") and vertical motions. Attitude must also be measured, but achieving altitude and heading is the top priority. Small lateral drifting due to minor attitude errors is not especially harmful to the mission objectives. By the time the vehicle achieves hover, the roll maneuver should be complete so as to minimize propellant burn, and precise attitude and heading measurements become the top priority as the vehicle prepares for lateral acceleration. During vertical descent, the system still focuses on precise altitude measurements, but hazard avoidance maneuvers must additionally be tracked. These maneuvers consist of roll and lateral motions, and might involve large deviations in attitude depending on the severity of the maneuvers.

During lateral maneuvers, the system primarily needs to precisely measure heading and odometry. Altitude and attitude measurements are less important during these phases of flight, except during the brief lateral acceleration burn. The more precise the heading, altitude, and attitude are prior to the acceleration burn, the more efficient the hop will be, as fewer corrective measures will be required to reach the target location. Hover hop trajectories are not designed to handle curved flight paths, making corrective measures especially costly, as the vehicle must stop completely prior to changing its heading.

The primary goal of the hazard detection system is to identify and characterize all hazards in or near the flight path of the vehicle, such as large rocks, cliffs, or canyons. A collision with any such hazard would likely have disastrous consequences, and as such the vehicle must scan for and track hazards in the direction of its flight path. During the vertical ascent, the vehicle must coarsely scan for hazards above itself,

[1]Landing spacecraft generally use the same coordinate frame as that defined for the Apollo Lunar Module (LM) [74]. In this system, the X axis is normal to the surface being landed upon, the Z axis is the primary forward direction ("out the hatch" for the LM), and Y completes a standard right-handed coordinate system. This means that "roll," which is always defined as rotation about the X axis, for a lander is physically similar to "yaw" for an aircraft.

such as a possible overhang from a cliff. During lateral translations, the vehicle must coarsely scan for hazards in the forward direction. Because such hazards would almost certainly be large, they do not need to be tracked with high precision (they only need to be detected so the vehicle knows to reduce or increase its altitude or stop and change its heading). During the vertical descent, precise hazard detection and tracking is required, as the vehicle must determine and navigate to a safe landing site. In order to conduct a safe landing, any hazards larger than the maximum survivable size must be located precisely.

## 2.4 Navigation Sensors

A navigation system architecture capable of performing the aforementioned measurements of the vehicle state in real-time is needed. This section presents a non-exhaustive list of available navigation sensors and compares them based on their capacity for achieving the above navigation and hazard detection goals.

### 2.4.1 Sensor Descriptions

The intention of this study was to select a suite of navigation sensors that are already available or are reasonably well developed. Inventing or maturing new sensors would almost certainly increase development time and cost. The sensors considered for a hopper navigation system are briefly described below.

#### Inertial Measurement Units (IMUs)

At least one inertial measurement unit is typically included in any spacecraft navigation system. They provide high-rate acceleration and rotation rate data in all six degrees of freedom (DOFs), and operate completely independently from any external systems. Their main drawback, however, is their tendency to drift with time due to

integration errors. The most accurate IMUs, with suitable drift and error characteristics, are the largest, heaviest, and most expensive, and would, in general, not be suitable for hoppers. A common solution to this problem is to constrain the error growth of the IMU using an additional sensor and a statistical estimator, such as an Extended Kalman Filter (EKF). This way the overall system still benefits from the high-rate measurement capability of the IMU, but for a much lower cost and mass.

### Vision Systems

As discussed in Chapter 1, there are a number of ways to extract navigation information from optical cameras. Some of these algorithms, such as many SLAM implementations, can calculate the full 6-DOF pose estimation. These systems offer many advantages, such as low hardware cost and mass, but are often computationally expensive and run at relatively slow rates. The error growth in such a vision system is directly related to how many tracked optical features are leaving the field of view of the camera. This means a downward-facing camera system will outperform a forward-facing camera system during a vertical maneuver, and vice versa.

### Laser Rangefinders

Laser rangefinders provide range information in a single direction. Scanning laser rangefinders rapidly move the beam along a set path at set increments. These sensors typically are very accurate for a relatively low cost and mass, and can have very long ranges, depending on the model. Typically a single beam model will have longer range and higher accuracy, and would be used to provide precise altimetry, while scanning laser rangefinders would more likely be used for coarse hazard detection at shorter ranges.

### Horizon Cameras

A horizon camera is a special class of vision system that purely detects the horizon line in its field of view. They are often used for attitude control at high altitudes, but are less effective at this at the low altitudes hoppers explore, especially in mountainous or cratered areas. At low altitudes, horizon cameras can be used instead to localize a vehicle coarsely and provide heading information with respect to mountain peaks detected on the horizon line.

### Star Cameras

Star cameras are used on spacecraft requiring precise attitude measurement, such as space telescopes. They can localize themselves to a known star map very accurately, and can be used on airless bodies, such as the moon. However, the long exposure times required to image the stars make them infeasible for use on moving hoppers, as the images would likely have too much blurring to be effective.

### Doppler Radar

Doppler radar systems are active sensors capable of measuring range and velocity using radio waves. They are highly robust with a great deal of heritage on landing systems at altitudes up to a few kilometers, but their limited accuracy makes them a poor fit for close-proximity vehicles such as hoppers. These radar systems tend to be prohibitively large and heavy for hoppers, as well.

### Doppler LIDAR

Doppler LIDARs are a relatively new development and do not yet have any flight heritage. These systems operate similarly to Doppler radars, but use light rather than radio waves, which gives them much higher precision. In part because they are so new, these systems require power conditioning equipment that is too large and heavy for

use on an unmanned planetary hopper, although they will be incorporated into the larger and more comprehensive NASA ALHAT system. In the future, a sufficiently small and light Doppler LIDAR might be developed that could be used effectively onboard a hopping vehicle.

#### Flash LIDAR

A flash LIDAR is an active sensor that provides a 2D depth map of its field of view, independent of lighting conditions. Many visual SLAM and hazard avoidance algorithms can operate upon this data with higher accuracy than that currently obtainable from optical imagery. However, flash LIDAR systems are still in an early stage of development and have mass and power requirements exceeding those of Doppler LIDARs. As such, these systems are not considered to be suitable for unmanned hoppers at this time.

### 2.4.2 Sensor Combinations

None of the sensors described above are capable of conducting all of the measurements required of a hopper navigation system on their own, and thus the system will require some combination of complimentary sensors. The primary functions of the navigation system, derived from the goals listed above are measuring odometry, altitude, heading, attitude, forward hazard detection, downward hazard detection (especially at the landing site), and proximity hazard detection. Table 2.1 shows the capability of each sensor at performing each function. The star camera, Doppler LIDAR, and flash LIDAR systems are in the table for comparison purposes, but as described above, are not suitable for an unmanned hopper navigation system in their current state of development. From this table, combinations of sensors capable of comprehensively navigating a hopper can be determined.

The left half of Table 2.1 shows the capabilities of each sensor with respect to vehicle navigation. It is clear from the table that vision navigation systems provide the

| Sensor | Navigation Function | | | | Hazard Detection Function | | |
|---|---|---|---|---|---|---|---|
| | Odom. | Alt. | Roll | Att. | Landing | Forward | Prox. |
| IMU | Poor | Poor | Fair | Good | None | None | None |
| Vision (Downward) | Fair | Good | Good | Good | Good | None | None |
| Vision (Forward) | Good | Fair | Good | Fair | None | Good | None |
| Laser Altimeter | None | Fair | None | None | None | None | None |
| Scanning Laser | None | Good | None | Poor | Fair | Fair | Fair |
| Horizon Camera | None | None | Fair | Fair | None | None | None |
| Star Camera | Fair | Fair | Fair | Good | None | None | None |
| Doppler Radar | Fair | Fair | None | Fair | Fair | None | None |
| Doppler LIDAR | Good | Good | Fair | Good | Fair | None | None |
| Flash LIDAR | Good | Good | Good | Good | Good | None | None |

Table 2.1: Capabilities of various sensors to perform the functions required of a hopper navigation system.

broadest capability across the range of required navigation functions. They also have the added benefit of being very useful for forward and landing site hazard detection. Combining the forward and downward vision systems offers solid performance across all functions. As is explored in depth in Chapter 4, the performance of the vision systems is significantly improved when additionally combined with an IMU. A Doppler radar system could potentially be used in place of the downward facing vision system, but it would provide less functionality at a higher mass and cost.

Proximity hazard detection is coarse detection of large hazards near the vehicle but outside its predicted motion. The purpose of this is to reduce unproductive hazard avoidance measures, which always first involve a roll maneuver to scan the region of predicted motion in detail prior to commencing lateral vehicle motion. If locations of likely hazards are known prior to rotation, the vehicle can rotate to a heading that is more likely to be free of obstacles without having to wait until it is detected by the primary, forward-facing hazard detection system. This means that the hazard detection does not need to be especially accurate or precise, as it merely provides an early-warning. It is clear from Table 2.1 that few options exist for proximity hazard detection. In addition to scanning laser altimeters, either an omnidirectional vision system or many monocular vision systems would be technically possible, but they would have high computational loads and significantly complicate the system, and

thus are not considered here. A scanning laser altimeter can provide this functionality for much less complexity than full 360-degree camera coverage.

## 2.5 Selected Architecture

A thorough analysis of the available sensor systems and their combinations led to the selection of a hopper navigation system utilizing both downward- and forward-facing vision systems and an IMU, as shown in Figure 2-4. A stereo vision system is used in the downward direction to provide accurate range data at a high rate, primarily during vertical maneuvers. A monocular system is used in the forward direction to provide precise heading tracking and odometry during lateral maneuvers. An IMU is included to accommodate rapid motions and to provide continuity amongst all phases of flight. Hazard detection capabilities of this system are further improved with the addition of scanning laser rangefinders for coarse, proximal hazard scanning.

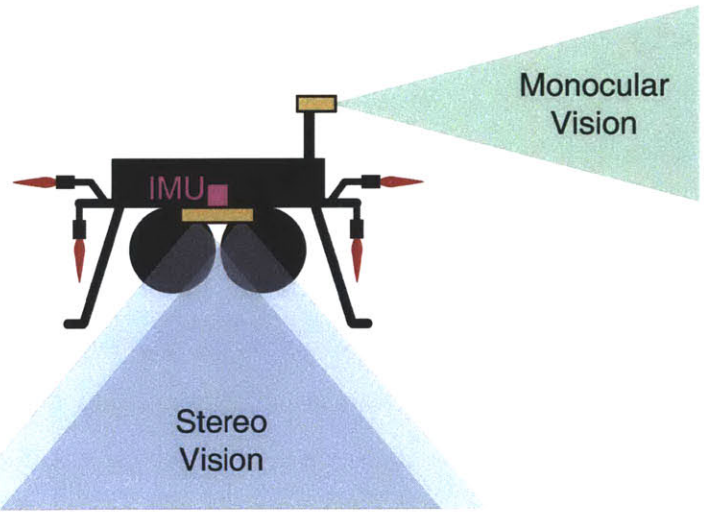



Figure 2-4: The selected hopper navigation system architecture. A downward-facing stereo vision system provides a high-rate position estimation during vertical flight phases, a monocular vision system provides odometry and heading tracking during lateral flight phases, and an IMU accomodates rapid motions and provides continuity to the system.

Error accumulated in vision systems is generally a result of detected features being lost from the field of view of the camera. This is because data is essentially "lost" from the

system as tracked features leave the field of view, and new features must be localized into the internal map to replace those recently lost. The uncertainty associated with this localization increases when the map contains fewer known locations in view. Because of this, vision systems are most accurate when they hold features in view for as long as possible. A hopper navigation system can benefit from the decoupled nature of the hover hop trajectory by using separate vision systems for vertical and lateral motions, ideally allowing features to be tracked consistently for the duration of each primary maneuver (i.e., vertical ascent, lateral traverse, and vertical descent).

For these same reasons, a backward-facing vision system should outperform a forward-facing system, because the same features will be tracked consistently for the duration of the flight, with new features being added but few ever being lost except due to being out of resolvable range (because the effective field of view is continually increasing). However, a backward-facing system provides no added benefits for hazard detection, and its tangible benefits over a forward-facing system in terms of odometry are relatively small. An ideal system, less concerned with mass, cost, or complexity, might include both forward- and backward-facing cameras.

### 2.5.1 Downward Stereo Vision System

A stereo vision system was selected for the primary sensor during vertical maneuvers. These systems provide accurate range data within their resolvable range, which is a function of the distance between the two cameras. Many currently implemented stereo systems output state estimates at higher rates than mono systems because depth calculation is a computationally simple matter of measuring the pixel distance between a single feature viewed from each camera[2].

Such a system is ideal for the vertical ascent and descent maneuvers, as tracked fea-

[2]Stereo vision systems operate on the same principle as a pair of human eyes. If an observer focuses on a single object at close range, they will see a shift in its relative position within the field of view of their eye if they alternately close their right and left eye. However, this shift becomes immeasurably small for distant objects, and the observer must rely on visual cues from their environment to estimate distances in these cases.

tures will typically be at close range and pose information is required at a high rate due to stricter hazard detection requirements during landing than during a lateral traverse. This detailed range information is particularly useful for landing site selection and hazard detection. This downward facing stereo vision system is also used during the horizontal traverse to augment altitude estimates from the laser altimeter and monocular vision system.

### 2.5.2 Forward Monocular Vision System

A monocular vision system was selected for the primary sensor during lateral maneuvers. This system is mainly used to determine the vehicle's heading and measure odometry. Monocular systems are not dependent on the disparity between images, and thus are not limited to the same extent by range. This allows a monocular system to track features that are further away, potentially even for the entire duration of the hop. As is demonstrated in the testing results presented in Chapter 4, a single downward-facing stereo camera has unacceptably high error when tracking odometry because features are continuously entering and exiting the field of view. These results made it clear that a single vision system would not be sufficient for a hopper.

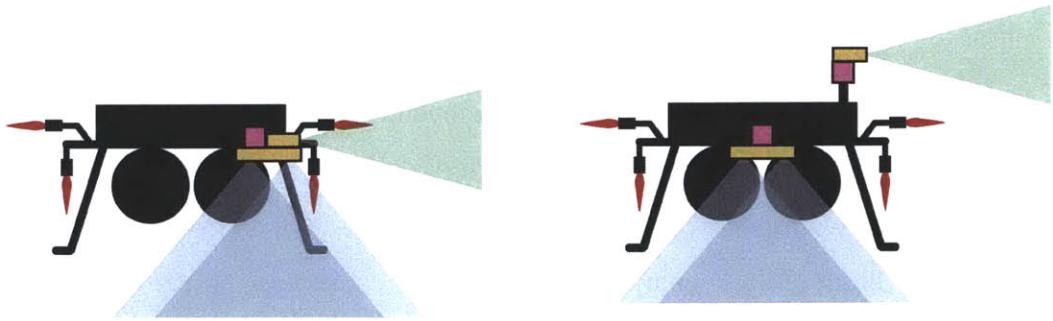
Monocular systems are more computationally intensive than stereo systems, and as such they cannot output state information as quickly. Because the system is further from obstacles and making fewer trajectory corrections during the lateral traverse, this slower output rate is sufficient. Similarly, the monocular system is sufficient for hazard detection functions during lateral maneuvers because hazards will be large and only need to be detected once, rather than localized and tracked in detail.

### 2.5.3 IMU

An IMU is desirable in addition to the dual vision systems because an IMU is always running and completely independent of external conditions. If the vision systems temporarily lose all their features, the IMU can continue to provide the guidance and

control system with navigation information, such as attitude, critical for maintaining the stability of the vehicle. The IMU also outputs at a much higher rate (typically 2-3 orders of magnitude) than the vision systems are capable of, allowing it to capture critical short period motions required for stable vehicle control. Also, the limits on IMU motion detection significantly exceed the expected dynamics of hopping, providing an additional level of robustness to the system. The added benefits of unifying an IMU with a vision system are described in greater detail in Chapter 4.

A unified IMU and vision system requires rigidity between the sensor systems. Therefore, depending on camera placement onboard the flight vehicle, two IMUs might be used instead of one if rigidity cannot be guaranteed between the two cameras. Ideally, the system will have two cameras and a single IMU rigidly mounted in a single, small package, as shown in Figure 2-5a. Alternatively, the cameras might be separated, either to simplify vehicle integration or to achieve better viewpoints for each camera. If this is the case, the system might require two IMUs, with one rigidly attached to each camera, as shown in Figure 2-5b. This should be avoided when possible because it requires additional hardware, the sensors are not rigidly mounted to one another, and it adds complexity to subsystem development and testing.



(a) Single IMU (b) Dual IMU

Figure 2-5: Alternative sensor layouts for a hopper navigation system.

### 2.5.4 Scanning Laser Rangefinders

Scanning laser rangefinders can be used to both increase the accuracy of the overall system and add proximal hazard detection capabilities to the vehicle. These systems ideally should have a range of approximately 30 m or greater, and can constrain the depth on both the downward and forward vision systems. In particular, monocular systems require an additional sensor to provide accurate scaling of distance measurements, which can easily be provided using a scanning laser rangefinder, provided that at least one of the laser beams consistently lands within the field of view of the monocular camera.

Furthermore, scanning laser rangefinders can be used for coarse navigation if the vehicle traverses above a featureless region. Because features roughly translate into hazards, a featureless environment typically is smooth and sufficient for landing [24]. If necessary, the laser rangefinders and IMU could still safely land the vehicle in such a situation.

### 2.5.5 Additional Components

Beyond the sensors themselves, a number of additional components are required for the navigation system. Because of the complexity of the vision algorithms, the navigation system should be integrated using an independent computer. This reduces the cost of integrating the system with the main flight computer. Because it requires its own computer, the architecture must also accommodate a two-way data connection to the primary flight GN&C computer, as well as a connection to the main vehicle power system. This effectively encapsulates the entire navigation system into a single sensor, as viewed by the primary flight computer, which simplifies both testing of the system and its integration onto flight vehicles. Furthermore, it allows its use with a larger variety of host vehicles and operational environments.

### 2.5.6 Required Development

Each of the classes of sensors utilized by this architecture is commercially available and already well understood. The next steps in the development of this comprehensive navigation sensor system are to develop the algorithms to convert sensor data into a navigation state estimate and to test these algorithms. The remainder of this thesis focuses on the unified stereo vision and IMU portion of this architecture. Chapter 3 describes a prototype system developed to test and evaluate the combined performance of the stereo vision and IMU sensors, and Chapter 4 describes the development and benefits of such a system in greater detail. Future work will involve similar development and testing of the monocular vision system and its integration with the unified stereo vision and IMU system.

# Chapter 3

# A Modular Testbed for Hopper Navigation

A sensor testbed, including a stereo vision camera, inertial measurement unit, and GPS receiver, was developed to allow evaluation of various combinations of navigation sensors and algorithms. This hardware and software testbed, called "Stingray," has a highly modular architecture, allowing additional sensors to be easily added to the system for real-time logging and incorporation into navigation algorithms. The algorithms themselves are fully encapsulated within the software framework, allowing multiple algorithms to run simultaneously for real-time performance comparison. Stingray, shown in Figure 3-1, is capable of operating independently of any particular carrier vehicle, allowing it to be tested onboard various flight vehicles with little or no modification required.

Stingray was developed primarily to serve as a proof-of-concept for a hopper navigation system, packaging unified inertial and terrain-relative navigation hardware and algorithms into a single "sensor," as viewed by a flight vehicle. However, its modularity and flexibility have additionally led to its use by other projects within Draper. The ease of data collection using the Stingray system allowed all results presented in this thesis to come from processing actual experimental data, with all of its imperfec-

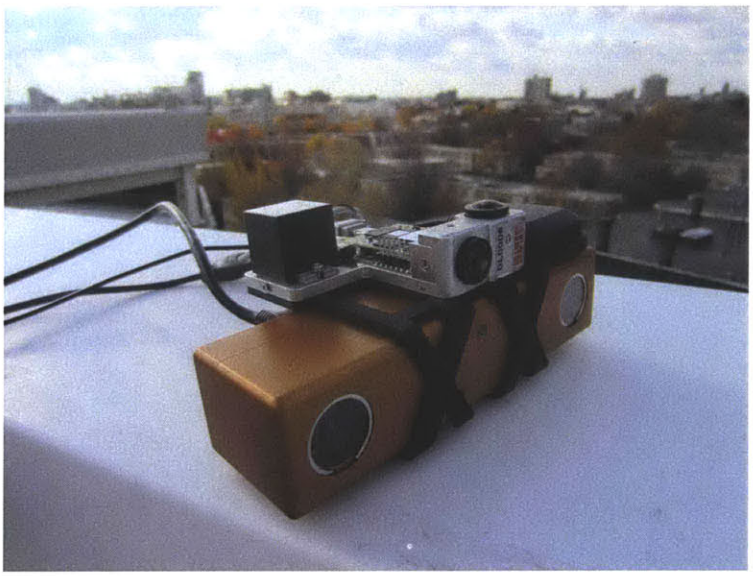


Figure 3-1: The Stingray navigation system.

tions and uncertainties, as opposed to simulations. Data collected using the Stingray navigation sensor provides both the basis for the discussion of tradeoffs between inertial and vision navigation systems applied to hopping vehicles, given in Chapter 4, and the testing results presented in Chapter 5. In the near future, a flight-ready hopper navigation system will evolve from the Stingray system, based on the same architecture.

## 3.1 Testbed Architecture

The primary goal of the Stingray testbed is to provide a hardware and software system capable of processing and logging data from a stereo camera, an IMU, and a GPS receiver. However, it was well known from the beginning that additional sensors might be added to the system, that the currently available sensors likely would not fly on an actual planetary hopping mission, and that the system requirements were uncertain. For this reason, priority was placed on the development of an architecture that heavily incorporated elements of modularity, flexibility, and upgradeability.

### 3.1.1 Prior Work and External Influences

The navigation system testbed project is heavily influenced by both prior work and a broad spectrum of goals. In part because the project evolved from previous architectures developed at the Draper Laboratory, much of the form follows that of prior work. From one perspective, this project involved linking various sensors with various algorithms that already existed or were in development. An evaluation of an existing software framework showed that it was a good fit for the purposes of this project, and with a few small modifications it was found to work very effectively.

This work is being conducted at a very early stage in the larger development process of an operational planetary hopper navigation system. For this reason, the system architecture was designed with a high priority placed on modularity and flexibility to account for both future uncertainties and the planned transition into a flight-ready system. A modular and flexible architecture allows straightforward reconfiguration to accommodate changing goals and requirements.

Additionally, multiple projects at Draper are interested in using the system for convenient test data collection or for use as an integrated navigation sensor. On some occasions this presented conflicts, which were largely resolved by designing the system to be as flexible as possible. For example, after the sensors and algorithms for the initial baseline system had been decided, two additional projects put in requests for an additional type of data in the system or algorithms running onboard. These issues were resolved by making the system even more modular than initially planned. Now various sensor and algorithm modules can be loaded or unloaded prior to system operation.

A particularly interesting project goal is to have the ability to perform demonstrations using the system. Not only must the system be capable of demonstrating hopper navigation capabilities, but another project at Draper is hoping to use the system for real-time demonstrations to first responders in the field. They have expressed a desire for a graphical interface to display real-time output in human-readable form.

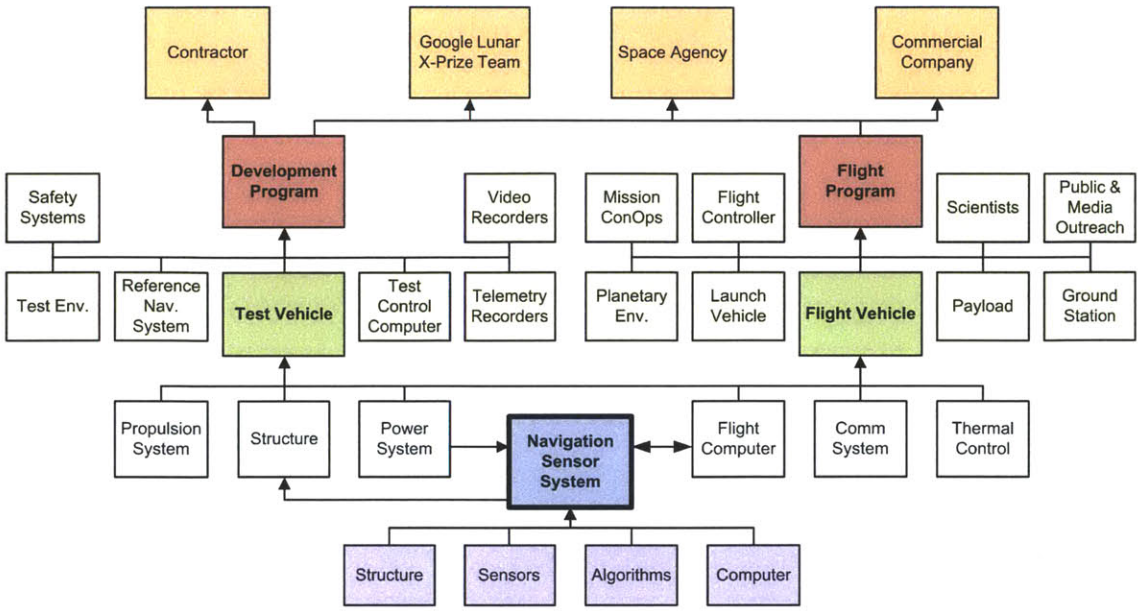



Figure 3-2: The Stingray navigation system was designed for both prototype testing as part of a development program and for incorporation into a flight program in the future. This system is encapsulated from any particular flight vehicle by interfacing only to the flight computer, structure, and power system (unless its own self-contained power system is utilized).

The modular architecture allows development of this system nearly independently of the Stingray system itself.

### 3.1.2 System Context

This prototype navigation system has been concurrently developed for two distinct operational contexts: that of a testbed for preliminary testing and development, and that of a mission-critical sensor onboard an operational flight vehicle. These two operational contexts are depicted in Figure 3-2. After the algorithms have been tested and verified using this prototype hardware as part of a development program, the same software framework can be easily ported to a new, flight-capable computer, ready to serve as part of a flight program. Only the sensor interfaces are hardware-specific, which will significantly reduce transition costs in the near future.

Flight-critical spacecraft components, such as navigation systems, have traditionally

been developed for a specific mission. However, there has been a recent push towards the development of "plug-and-play" architectures for spacecraft [75], in an effort to reduce development costs by developing fewer, more generic systems with increased reliability, reusability, and interoperability. Recent work has involved preliminary investigations of the feasibility of plug-and-play spacecraft GN&C systems [76]. Following this trend, one of the goals of this project is to determine how independently a navigation system can be developed from the larger context of a flight vehicle development program, within the scope of planetary landing and surface exploration, for plug and play-like application to future flight projects.

Traditionally, spacecraft navigation systems have been integrated within the primary flight computer, requiring direct interfacing to all navigation sensors and sharing processor time with other critical systems. This navigation system has been developed in such a way that it can be considered by a primary flight computer to be encapsulated as a single "sensor," outputting processed vehicle state estimation, as shown in the interface diagram in Figure 3-3, ready for direct use by the guidance and control systems. This reduces development costs by allowing more system-specific testing earlier in the project, reducing the number of interfaces, and simplifying vehicle integration testing. In addition, the more independently a subsystem operates from a given flight vehicle, the more applicable it is to other missions. The goal is that this system could provide near-surface navigation capability to planetary hoppers designed to explore the Moon [9], Mars [21], or Titan [22], or even surface vehicles and UAVs for use on Earth.

### 3.1.3 System Concept

A number of concepts were in contention for a system to navigate a planetary hopping vehicle, with some of those considered shown in Figure 3-4. The main choice was between a relative navigation system and an absolute navigation system, with the relative system winning out due to its significantly higher accuracy at low altitudes, where hover hop trajectories take place. The concept of this system is to use a

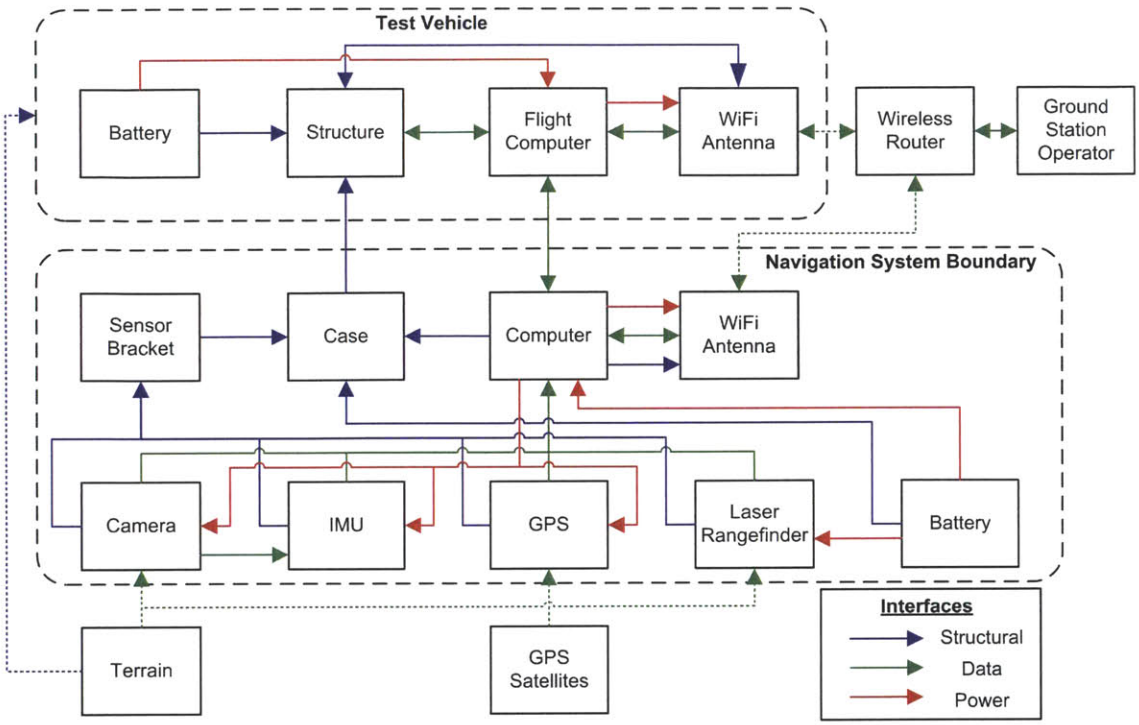



Figure 3-3: The system interfaces to a parent vehicle through a single command and data line, as well as a structural link. The majority of the interfaces are fully self-contained, reducing the integration and test burden.

vision-based sensor and an inertial sensor together to create a relative navigation system that can provide a navigation state to the primary flight computer of a hopping vehicle. This concept was chosen as a result of several tradeoffs. The basic reasoning is that neither a vision-only nor an IMU-only system can satisfy all of the requirements of a hopper navigation system, as was described in Chapter 2. However, when combined, each can overcome the weaknesses inherent in the other to provide a more comprehensive navigation solution, as described in Chapter 4.

### 3.1.4 System Function

The Stingray system provides two externally delivered functions: testing navigation algorithms and navigating an exploration vehicle. This system can be described using Object-Process Methodology (OPM), which is a modeling language used to design and depict information systems [77]. The decompositional view of system function,

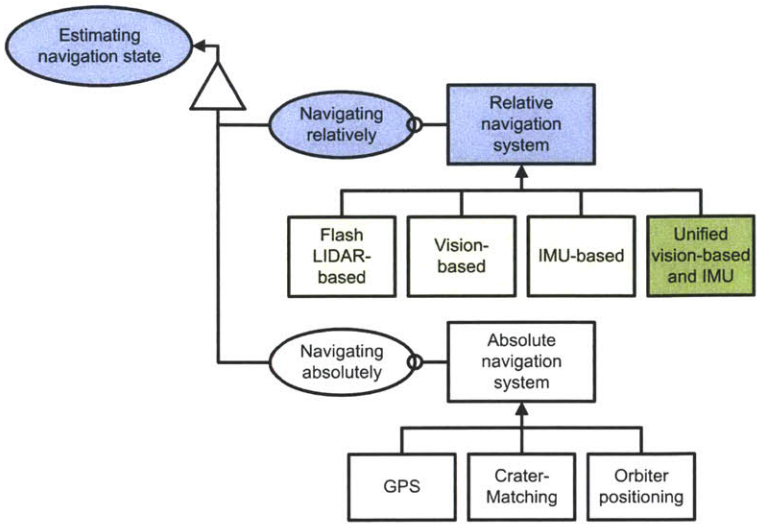



Figure 3-4: Various navigation concepts considered for hopping. A relative navigation system utilizing vision and inertial sensors was selected.

shown in Figure 3-5 using OPM notation, shows the benefits of encapsulating and isolating the navigation system from the flight vehicle as a single sensor. The majority of the complexity of the system grows from two primary internal functions: securing the individual sensors relative to one another and estimating a navigation state from the data gathered from the sensors. Both of these functions exist entirely within the system boundary, and therefore are ideal candidates for encapsulation.

As Figure 3-5 shows, the only functions of the system that require outside influences (with the exception of the individual sensors, which sense the external environment), are securing the encapsulated navigation system to a test vehicle, receiving commands, and outputting the state estimate. A typical spacecraft navigation system would have more functions performed by the host vehicle, rather than occurring onboard the sensor platform. Most significantly, a traditional system would have the state estimation software integrated with the primary flight computer, which would increase ambiguity and testing complexity.

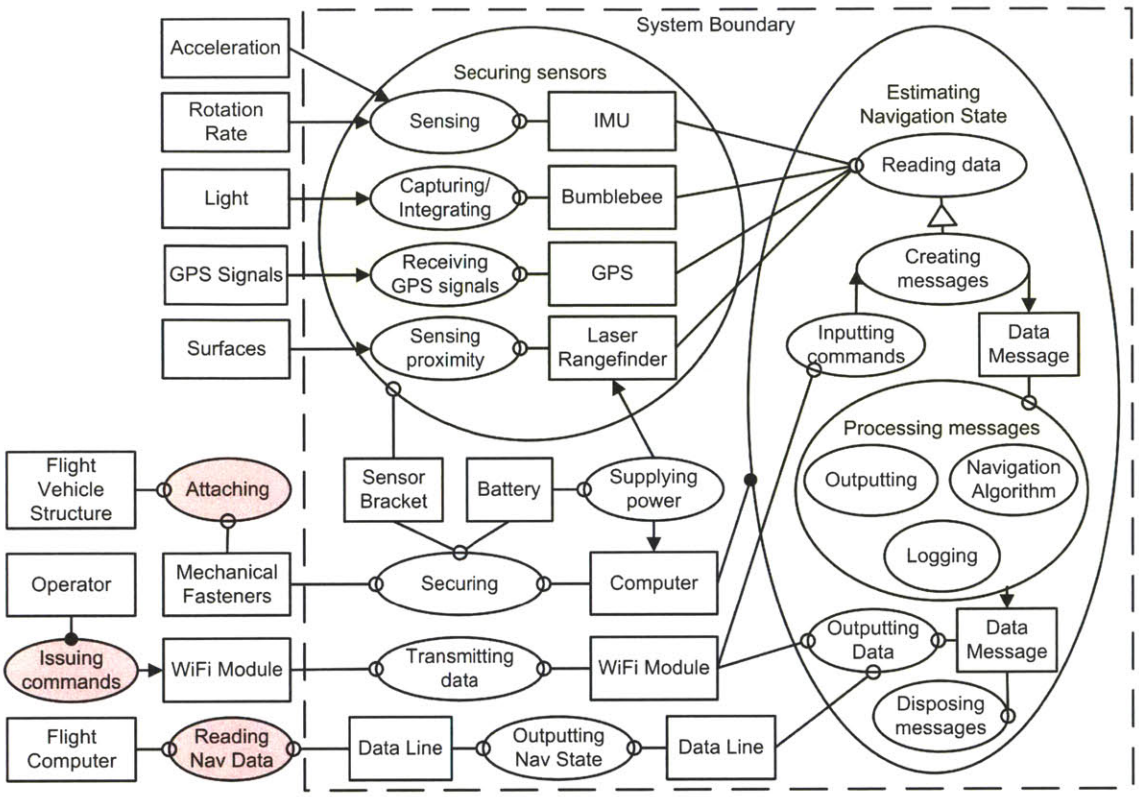



Figure 3-5: Functions provided by the navigation system. The majority of system complexity is contained within the system boundary, with the two primary functions of the system shown in green. The three functions that the parent vehicle must provide are shown in red.

### 3.1.5 Operator Attributes

This system has two distinct operational settings. In its current form, which was used to collect all data presented in this thesis, the system has a human operator. This makes it easier and less expensive to conduct repeated test runs. In the future, the system will be operated by a parent flight vehicle. The parent computer will issue commands to the navigation sensor in a very similar manner to those issued by a human operator. For these tests, the system must operate very robustly, as a human operator will not be present for troubleshooting. Before the system flies on an actual hopping mission, it must be shown to be extremely tolerant to any possible errors, as the mission will depend on consistent performance from this navigation system.

Figure 3-6 shows the standard operational sequence for the system during a test run

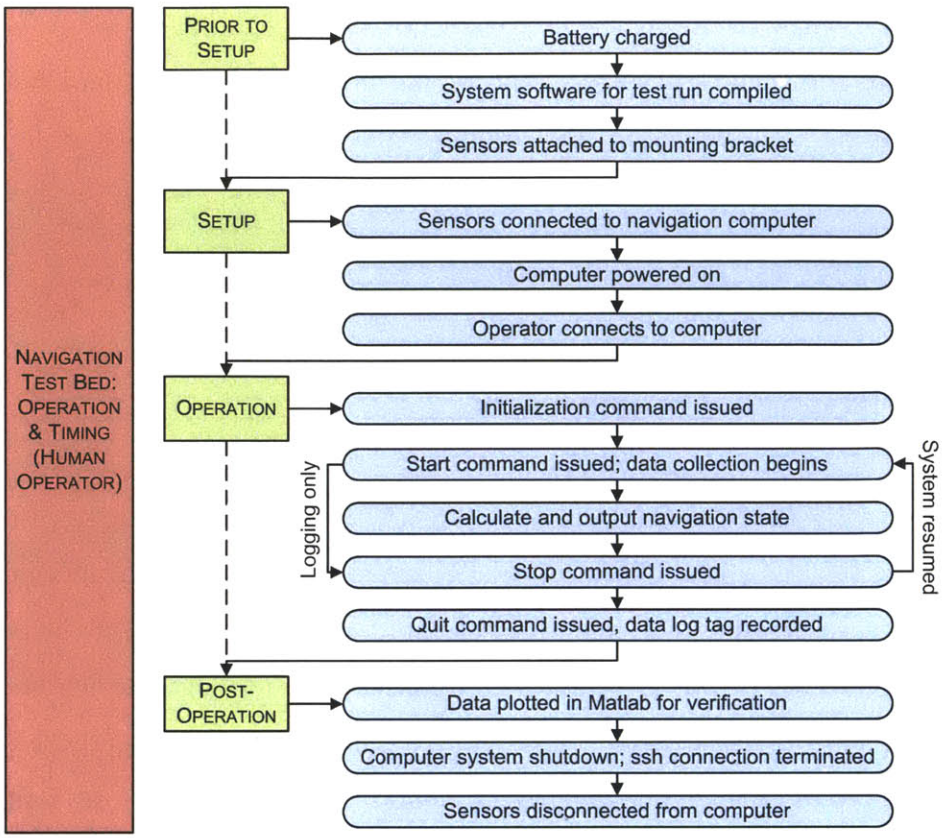



Figure 3-6: The Stingray operational sequence.

with a human operator. The operational sequence and interface are designed to be very straightforward once the operator has received a small amount of training. If this system is used with a parent flight computer, all commands currently issued by the operator via either keyboard or Secure Shell (SSH) network protocol will be issued by the flight computer over RS-232 or a similar interface. There are preliminary plans to add an external tablet interface to the system, which could provide real-time status updates, navigation state visualization, and system health monitoring to a test operator from remote.

No contingency or emergency operations have currently been implemented or accounted for. The system is not capable of issuing any internal actions (from a software perspective) that could cause damage to itself, as all of the sensors operate passively. Damage to the system could occur if a flight vehicle crashes or a human operator drops it, though this risk is largely mitigated by the low hardware cost of the system.

In the future, when the system is used as a primary navigation sensor to a critical flight system, additional measures should be put in place to conduct real-time hardware monitoring, fault detection, and recovery. Given the current state of the project as a testbed primarily used for data collection, these measures have not yet been implemented.

## 3.2 Software Framework

The system software takes the form of an asynchronous task queue with distributed message passing, implemented using C++. Such a framework can be loosely described as having tasks, messages, and queues. Tasks are the execution units of the program, which are connected by queues. Data is contained within messages, which are passed along the queues between tasks. This type of framework is very modular, in that different tasks can be added or removed to build a custom system to meet the needs of the user. A sample task-message layout is shown in Figure 3-7, where tasks are represented by blocks and queues are represented by arrows pointing in the direction of data flow.

### 3.2.1 Messages

All data within the system is encapsulated within generic "messages," which are passed between the various tasks in the system using queues. Each message contains a timestamp and a data structure. The timestamp indicates when the message was created, so the system can continue to process and log data even if the processor becomes saturated, adding an additional level of robustness.

Each type of data (e.g., IMU data, camera images, or a state estimation) has its own implementation of the generic message template. These messages stack up within the queues, waiting for tasks to operate upon them. When a message reaches a task at the end of a queue, the task first polls the message for its type. If the task operates

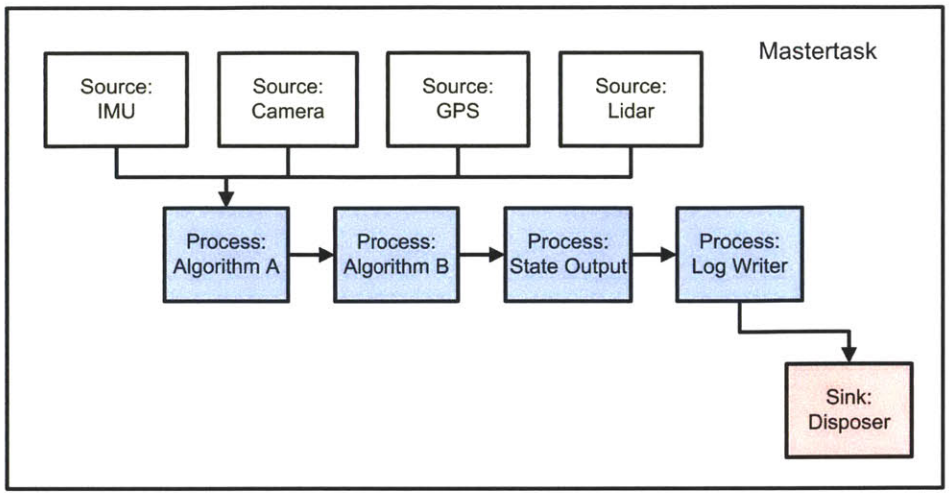



Figure 3-7: The Stingray software framework consists of tasks and message queues. Tasks create and operate upon messages, which move between tasks using queues. Source tasks (green) create messages, process tasks (blue) act upon data or commands within messages, and a single disposer (red) disposes of the messages at the end of the final queue. The "mastertask" manages all tasks within the system.

upon that data type, it processes the data and then returns the message to the next queue; otherwise the message is simply passed on to the next queue. This way all of the messages remain time-ordered. Additionally, each message type includes a description of itself for logging purposes. This allows a generic logger to easily and automatically log every message passed through the system without needing to understand each message type, which further adds to modularity.

### 3.2.2 Tasks

All of the code relating to sensor interfacing, algorithms, or data processing is encapsulated within various task objects in the system. Tasks are further categorized as source, process, or sink tasks, each of which inherits from a generic "queue task," as shown in Figure 3-8. Source tasks create data, process tasks operate upon data, and a single sink task resides at the end of the queue. A single "master task" contains all of the tasks and issues commands to each, telling them to initialize, process data, or halt, for example. The task framework handles all low-level functions of the program, allowing the tasks themselves to be written at a high level.

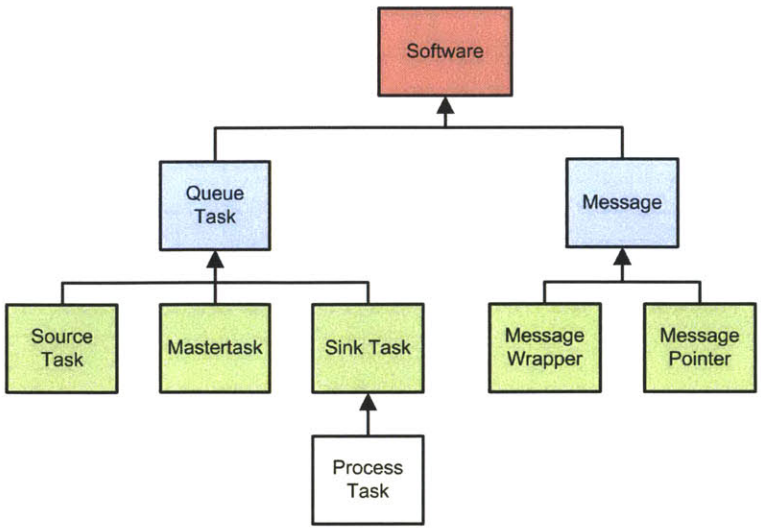



Figure 3-8: The software framework is comprised of tasks and messages, which inherit from the generic "queue task" and "message," respectively. Tasks are further specialized based on how they operate upon messages. Each message consists of a "wrapper," which encapsulates the data, and a "pointer," which is passed along the message queue.

Adding an additional sensor to the system is as simple as adding a source task, and adding an additional algorithm simply requires adding an additional process task. The asynchronous nature of the framework allows multithreading of the tasks across multiple processors, which helps keep the system flowing, especially if multiple algorithms are run simultaneously for comparison. Unnecessary tasks can be easily removed prior to a flight test to reduce the load on the processor, such as operator monitoring tasks.

## 3.3 Testbed Hardware

In its current configuration, the Stingray system comprises a single-board computer, stereo camera, IMU, and GPS module, which are described in detail below. A sensor bracket mounts the camera, IMU, and GPS receiver rigidly together, and can be mounted remotely from the computer (within range of the cabling), as shown in Figure 3-9. A scanning laser rangefinder will be added in the near future. A decom-

positional view of the current Stingray system form to three levels is provided in Figure 3-10. The modular nature of the system means that many of the first level components (shown in blue in the figure) could be added, removed, or upgraded with little or no impact on the other hardware components. Prior to flight onboard an actual hopper, it is expected that each of these components will be upgraded or replaced with higher-grade, space-rated components.

One of the key advantages of this hardware system over simply logging data natively is that each critical data packet is provided with an accurate timestamp when the data is collected, rather than simply time-tagging the packet when it is received by the host computer CPU. Time-accuracy of the data is critical to achieving maximum performance from the navigation algorithms.

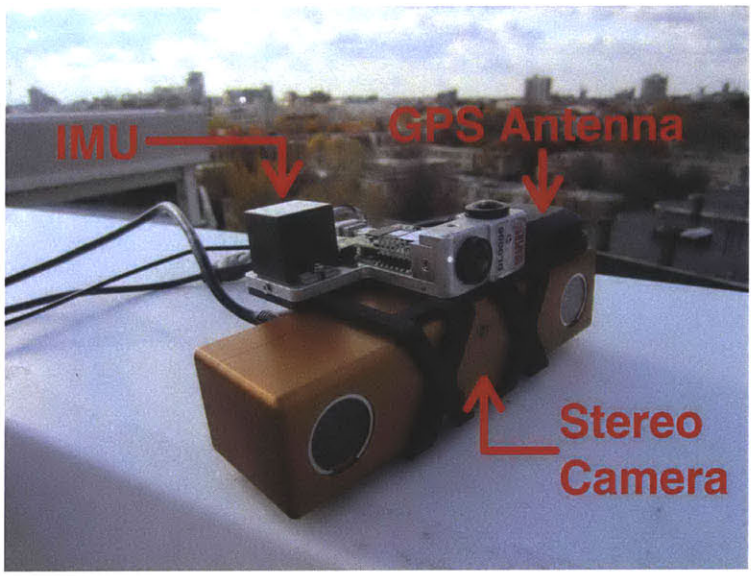



Figure 3-9: The Stingray system includes a stereo camera, IMU, and GPS antenna. The IMU is accessed using a dedicated interface board that also houses additional, unused sensors.

### 3.3.1 Computer

At the core of the Stingray system is a Jetway NF81 single-board computer. The NF81 is a mini-ITX form-factor (17 x 17 cm) motherboard with an AMD Brazos

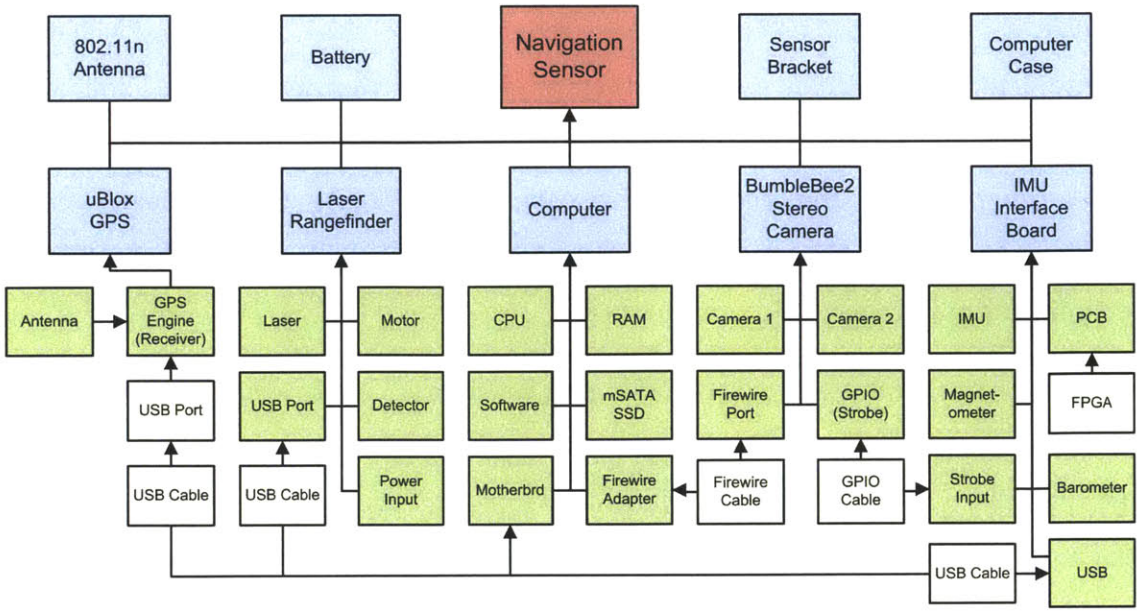



Figure 3-10: Detailed breakdown of the hardware framework.

eOntario G-Series APU (combined CPU and GPU). It has 8 GB of DDR3 RAM and an 80 GB mSATA solid state drive (SSD). The system also has a wide range of I/O, including eight USB ports, two RJ-45 (Ethernet) ports, and ten pins of GPIO. A Firewire adapter on a mini-PCIe slot adds an additional three IEEE 1394 Firewire ports. A picture of the computer is shown in Figure 3-11.

The Stingray computer runs Ubuntu Linux 11.04 Server Edition, but the software framework is cross-platform compatible, allowing the code to also run on Windows or Mac OS. The computer itself is far more powerful than a typical, space-rated flight vehicle would likely use, ideally allowing multiple navigation algorithms to run in parallel simultaneously for comparison and evaluation.

### 3.3.2 Inertial Measurement Unit

The IMU currently in use on the system is an Analog Devices ADIS16364BMLZ industrial-grade MEMS IMU, whose specifications are provided in Table 3.1. A Draper-developed interfacing board, including a digital signal processor (DSP), provides a convenient interface to the IMU, as well as a magnetometer, barometer, and addi-

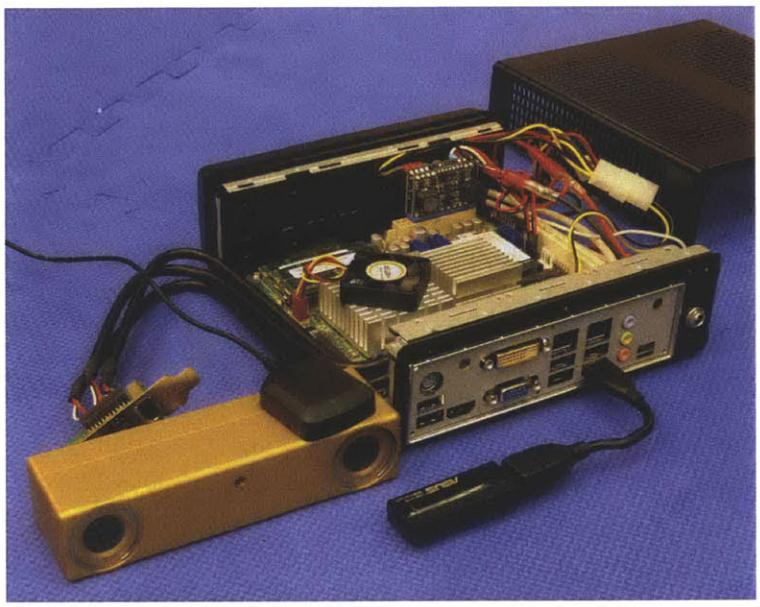


Figure 3-11: The Stingray computer, with its case open. A 802.11n wireless antenna allows remote operation of the system within range of a wireless router. The stereo camera (gold) and GPS antenna (black) are also shown.

| Property | Specification |
|---|---|
| Number of Axis | 3 |
| Max Acceleration | 5 |
| Max Rotation Rate | $\pm 300°/s$ |
| Output Rate | 1200 Hz |
| Size | 23 x 23 x 23 mm |

Table 3.1: IMU specifications.

tional cameras, though they are currently unused for this project. The DSP provides precise timestamping of the IMU data, and is capable of precisely recording an external input pulse from an additional sensor.

### 3.3.3 Stereo Camera

The stereo camera is a BumbleBee2 monochromatic stereo vision camera from Point Grey Research, which provides a well-integrated vision system for testing. Draper has multiple versions of these cameras, allowing testing of various frame rates, resolutions, and lens focal lengths, as shown in Table 3.2. The version most frequently used for

| Camera Model | Focal Length | Field of View | Max Resolution | Max Frame Rate | Max Range |
|---|---|---|---|---|---|
| BB2-08S2M-6 | 6 mm | 43° | 1032 x 776 | 20 Hz | 78 m |
| BB2-03S2M-38 | 3.8 mm | 66° | 648 x 488 | 48 Hz | X m |
| BB2-08S2M-25 | 2.5 mm | 97° | 1032 x 776 | 20 Hz | X m |
| BB2-03S2M-25 | 2.5 mm | 97° | 648 x 488 | 48 Hz | X m |

Table 3.2: Stereo camera specifications for available camera models.

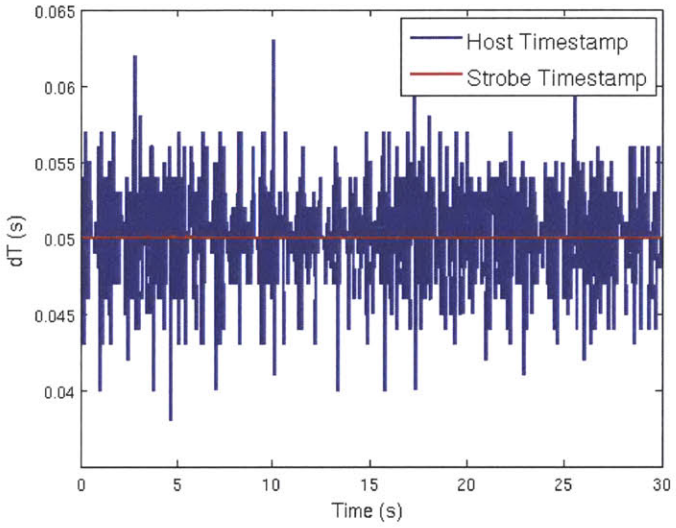



Figure 3-12: Length of each interval between receiving a new timestamp for each of the two methods, which should be a constant 0.05 seconds. Timestamping accuracy is improved by over three orders of magnitude when logged using the camera strobe output rather than natively by the host computer processor over Firewire.

testing is the BB2-08S2M-6, which has a 6 mm focal length lens (43 degree field of view) and runs at up to 20 fps. The camera resolution is typically downscaled to 512 x 384 pixels for each camera prior to image processing. The software system automatically configures any of these cameras at run-time, so they can be interchanged in the field without need for any software changes. Each BumbleBee2 camera connects to the computer via an IEEE-1394a interface, which additionally supplies 2.5 W of power.

The camera also has a 12-pin GPIO interface, which is used to output a strobe pulse to the IMU interface board at the start of integration of each camera frame, which

increases the accuracy of the timestamp for each frame. A comparison of time stamp consistency from logging using the host computer CPU versus the output strobe pulse is shown in Figure 3-12. Logging the image timestamps using this method improves timing accuracy by over three orders of magnitude, which results in improved navigation performance.

The BumbleBee2 stereo cameras rely on the disparity between two images taken simultaneously. The maximum resolvable depth is a function of the distance between the cameras (the baseline distance), image resolution, and camera focal length. Each camera has a common baseline of 12 cm, but resolution and frame rate vary from model to model. Figure 3-13 shows depth measurement uncertainty as a function of actual feature depth (corresponding to pixel disparity) for each camera focal length at a resolution of 512 horizontal pixels. The camera tested to have the best overall navigation performance (and which is used for all results presented in this thesis unless otherwise stated) is the BB2-08S2M-6, with a maximum resolvable depth of 78 m. This camera has the lowest measurement uncertainty in the 5-20 m range, which is the expected altitude range for the majority of hopper operations.

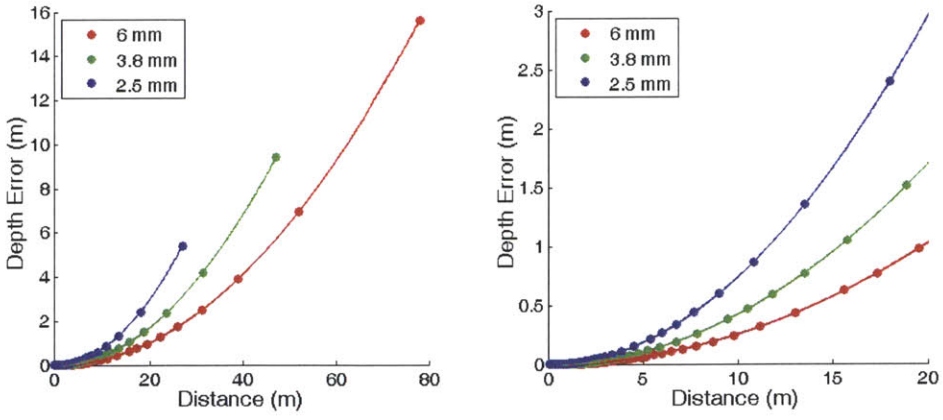


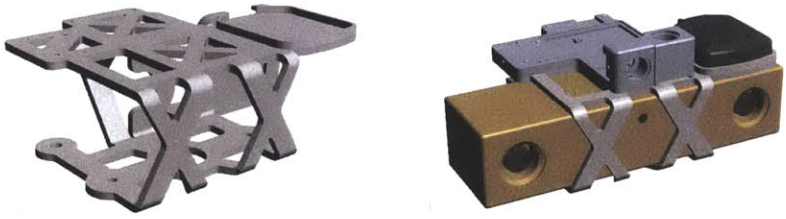

Figure 3-13: Estimated resolvable depth and measurement error for BumbleBee2 Stereo Cameras by focal length. The 6 mm lens has the highest accuracy for depths in the 5-20 m range and can measure distances up to 78 m, but has large uncertainty at long range.

### 3.3.4 GPS Receiver

GPS data is provided by a uBlox EVK-6T Precision Timing Kit and ANN-MS-0-005 active GPS antenna, which is capable of outputting GPS position and velocity estimates as well as raw pseudorange data. GPS data is logged by the main computer over a USB connection at approximately 1 Hz, and each data packet is timestamped by the host processor (as the higher level of accuracy achieved by logging with the DSP is unnecessary for GPS). Although GPS data is not available to planetary hoppers, it is used during testing to provide both an initial position for each dataset and an absolute reference to compare against. Additionally, other projects at Draper use the Stingray system to collect raw GPS data due to the system's ease of use. The GPS positioning data is typically accurate to within 2-3 meters.

### 3.3.5 Sensor Bracket

The stereo camera, IMU (with interface board), and GPS antenna are rigidly mounted to one another using a sensor bracket, shown in Figure 3-14. The bracket was designed using computer-aided design software and manufactured using a computer-controlled Stratasys Prodigy fused deposition modeler (FDM). The bracket is made of Acrylonitrile Butadiene Styrene (ABS) with a tensile strength of 22 MPa and flexural strength of 41 MPa, which provides adequate rigidity between the camera and IMU. Brackets for new sensor configurations have been designed and manufactured in less than 24-hours. These mounts can either attach to a case enclosing the computer or be mounted independently on a vehicle, connected by data and power cables.



(a) Bracket (b) Bracket with Sensors

Figure 3-14: A rapid-prototyped bracket mounts the sensors rigidly to one another.

# Chapter 4

# Unifying Inertial and Relative Navigation Solutions

As described in Chapter 3, the "Stingray" system was developed to evaluate the performance of various navigation sensors and algorithms. This system is capable of logging timestamped data from multiple sensors. Navigation algorithms can run in real-time onboard the system, or at any time afterwards using a Matlab interface developed to post-process the data. This allows a single dataset to be processed using multiple algorithms for performance comparisons. Data collected using the Stingray system is here used to provide the basis for a comparison of inertial and vision-based terrain-relative navigation systems and to demonstrate the benefits of combining these two systems into a single, unified navigation system.

Two different algorithms, developed internally by the Draper Laboratory for previous projects, were available for this comparison. The first was an inertial system intended for use with a small, low-cost IMU, which is constrained by updates from a GPS receiver and was originally developed for a personal navigation system. The second was a stereo vision algorithm capable of outputting the relative pose of a camera with respect to its initial pose. The hypothesis was that a single system, combining the advantages of both the inertial and vision systems, would outperform both existing

systems.

## 4.1 Inertial Navigation

The first existing navigation system tested, the Inertial Navigation System (INS), combines a commercially available Analog Devices IMU with a uBlox GPS receiver, in a similar fashion to those described in the literature [78, 79]. Taken alone, the IMU accrues error exponentially with time (a characteristic common to any IMU), so an absolute position update from the GPS receiver is used to frequently constrain this error growth using an extended Kalman filter (EKF).

The combination of these sensors improves pose knowledge between GPS updates, as shown in Figure 4-1. IMUs are particularly well suited to measurement of high-rate, short period motion. However, if GPS updates become unavailable, the integrity of the INS is rapidly lost, as shown in Figure 4-2. Thus, navigation systems in GPS-denied environments, such as other planetary bodies than Earth, must either use a high-grade IMU (which are typically heavy and expensive) or receive position updates from an alternative navigation sensor.

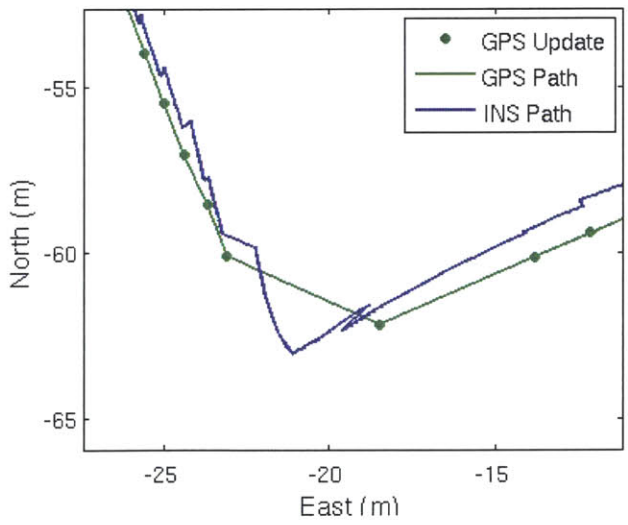



Figure 4-1: The INS path (shown in blue) provides sensor pose between GPS updates (green).

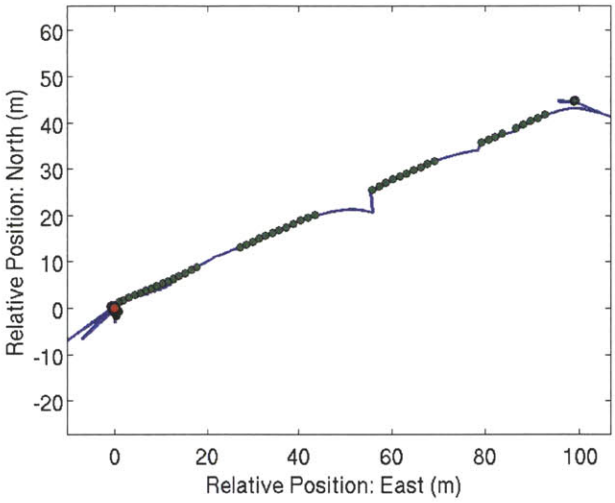



Figure 4-2: Sample INS Navigator path (blue) and associated GPS updates (green), illustrating the effects of an INS navigation system during periods of GPS unavailability.

## 4.2 Surface-Relative Navigation

The second navigation sensor tested uses the Draper Laboratory's stereo vision-based Simultaneous Localization and Mapping (SLAM) implementation, hereafter referred to as 'DSLAM,' similar to those in the literature [58]. The basic concept of SLAM is that a map of an unknown environment is built from observed 3-dimensional "landmarks," and the observer is localized within this map. DSLAM acquires images from a stereo camera and then performs the following steps to measure the position of the camera relative to its initial position: feature detection, feature matching, pose estimation, and landmark registration. These steps are briefly described below.

### Feature Detection

"Features," or 2-dimensional points of interest, are detected in the images from each camera using a FAST (Features from Accelerated Segment Test) feature detector [80, 81], and are combined into corresponding pairs between the two cameras

of known spacing. The distance between the camera and a detected feature pair (corresponding to a single point of interest in the environment), or "depth," is calculated from the distance between the detected features in the stereo image pair, or "disparity," measured in pixels. This process is repeated for each set of stereo images, or "frame."

### Feature Matching

The rotation of the current frame is first coarsely estimated using sum-of-squared difference (SSD) inter-frame tracking, which aligns the current images with the corresponding images from the previous frame using image intensity. This rotation is then used to project all of the landmarks already existing in the internal map into the 2-D plane of both the left and right stereo images. The features from the current frame are matched with these projected landmarks in the image plane using the mean sum of absolute difference (SAD) to determine best matches. A sub-pixel refining step determines which features remain matched and which are rejected.

### Pose Estimation

The camera pose is estimated using three-point-pose RANSAC (RANdom SAmple Consensus) [82], which calculates the transformation (change in pose) between three randomly chosen new feature pairs and the landmarks they were matched to in the map and repeats this process many times. This generates a large but noisy set of pose estimations, and RANSAC is used to iteratively select a subset of "inliers" from this data that represent a best estimate of the actual camera pose. The newly detected features that do not agree with the pose estimate from the inliers, called "outliers," such as features detected from objects in motion, are then thrown out to prevent causing errors in subsequent frames.

### Landmark Registration

A frame is promoted to a "key frame" if it has exceeded a given threshold of motion since the last key frame. If this threshold is exceeded, a new key frame is registered and the RANSAC-determined inliers from the previous step are registered in the world map as landmarks in 3-dimensional space. Registering landmarks only at key frames as opposed to every frame prevents the map from getting too cluttered or growing too large to operate upon efficiently. DSLAM keeps these landmarks in the map for only a set number of key frames, called the "window." Once a key frame is no longer within the window, all of its landmarks expire and are removed from the map. This system is ideal for mobile observers that do not return to their original location, such as hoppers, as the window length can be set so that landmarks are no longer tracked once they are behind the vehicle, reducing the map size and speeding up the system without affecting performance.

### Sources of Error

Pose error using DSLAM is primarily related to errors associated with estimating the locations of feature pairs, matching features to landmarks, and the loss of landmarks from the field of view. For example, uncertainty (and thus error) grows every time an established landmark is no longer tracked and a new landmark must be initialized to replace it. This is because the new landmark can, at best, only be localized to the accuracy of previous landmarks. For this reason, the system has difficulty with large rotation rates because during these motions the system must rapidly initialize new landmarks as the existing landmarks are lost from the field of view, causing error to accrue especially quickly. If all of the existing landmarks are lost, the system must effectively start over and build a new map. In contrast, the system does well with straight trajectories, where new landmarks are initialized as they come into the field of view to a much larger cloud of existing landmarks at a more gradual rate.

DSLAM also has difficulty with environments with few distinguishing features, which

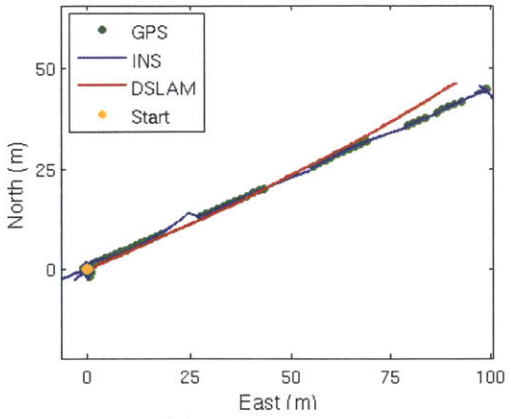
result in few landmarks and a sparse map that is less conducive to outlier rejection by RANSAC. The system is also limited by the characteristics of the stereo camera, which limit the resolvable range of distant features to the minimum pixel disparity. Beyond this distance, detected features have very high uncertainties, especially in depth, as was shown in Figure 3-13. Despite these shortcomings, DSLAM is well suited to navigating low rates of motion in environments with an abundance of high contrast terrain within the resolvable range of the stereo camera.

## 4.3 Algorithm Comparison

In order to compare the aforementioned navigation algorithms side by side, sensor data was collected for both navigation systems simultaneously using the absolute references of a GPS receiver and the lines on the MIT football field. Figures 4-3 through 4-7 show the actual path traveled in green (from GPS measurements), INS results in blue/yellow, and DSLAM results in red for several challenging situations for a stereo vision system. The concentric yellow circles in the figures represent regions of 5 and 10 percent total positioning error at the conclusion of the test. To illustrate the difficulties of inertial systems, the INS paths are calculated using random periods of intermittent GPS data. It is important to note that IMU error is easily constrained when sufficient position updates are available, but increases exponentially in the absence of such updates.

### 4.3.1 Distant Features

Stereo imaging systems rely on binocular disparity between simultaneously captured images to determine feature depth. However, as features grow increasingly distant, this binocular disparity shrinks to be less than the pixel width of the camera, eliminating the system's ability to accurately localize features in the map. The majority of the high-contrast features detected around the MIT football field are outside of the



(a) Relative Frame

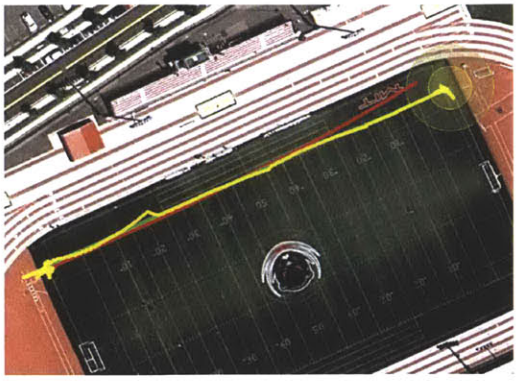

(b) Google Earth

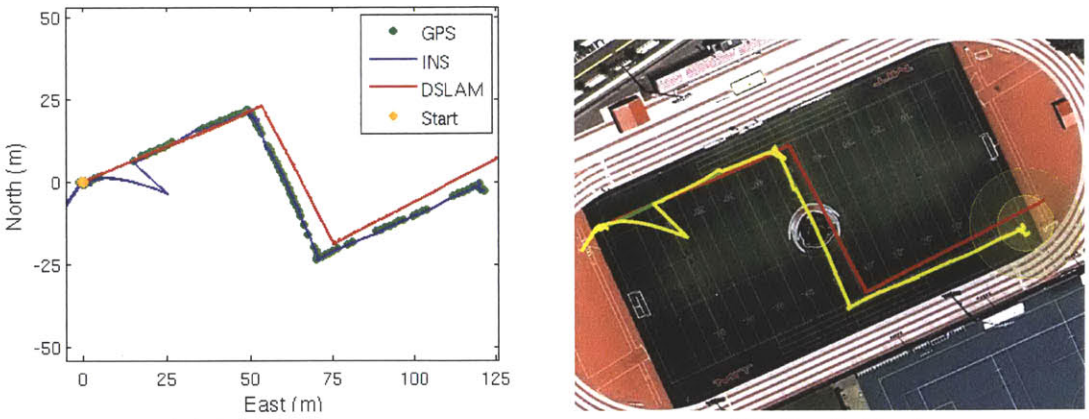
Figure 4-3: Vision (DSLAM) and inertial (INS) navigator results for a straight line trajectory with distant features and periodic loss of GPS, plotted in the relative frame (left) and in the global frame (using Google Earth, right). The concentric yellow circles in the Google Earth plot represent regions of 5 and 10% cumulative accuracy over the entire trajectory with respect to the actual finish location. The points in green at left show the specific locations of the GPS updates received by the INS system.

resolvable range, as the field itself is monochromatic and flat. As shown in Figure 4-3, the vision system does well initially at maintaining its heading, but error begins to accumulate as the features being tracked begin to be resolvable in depth (around the 50 yard line), inducing error as they are effectively lost from view and re-acquired. Additionally, the lack of depth disparity reduces accuracy in odometry measurements, typically resulting in a 5-8% cumulative positioning error.

For comparison, the INS system experiences GPS outages near the 20 and 50 yard lines, causing error to accumulate until updates are reestablished. In this case, INS total heading error grew to that of DSLAM in roughly one eighth the distance traveled.

### 4.3.2 High Rotation Rates

The vision navigation system's accumulated error increases as tracked features leave the field of view. This happens most significantly during rotations, as the system rapidly loses tracked features and must detect and localize new ones, each carrying



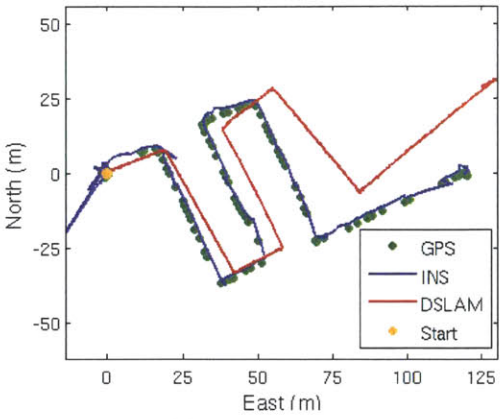

(a) Relative Frame

(b) Google Earth

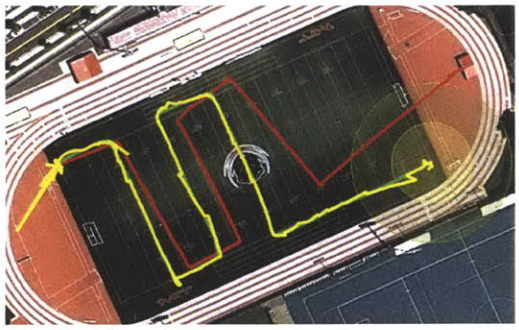
Figure 4-4: Vision (DSLAM) and inertial (INS) navigator results for a trajectory including two 90-degree turns and periodic loss of GPS. The inertial system accumulates error when GPS updates (shown in green) are unavailable, while the vision system mainly accumulates error during turns.

with it additional uncertainty. Figures 4-4 and 4-5 show paths with two and six 90-degree turns, respectively. These figures suggest that the majority of the error occurs during the turns, and error increases with additional turns. The two-turn path ends up with a position error of just over 5%, while the six-turn path misses even the 10% error target.

An abundance of high-contrast terrain borders the athletic complex on all sides. However, during both tests, the heading errors from each 90-degree turn are more than five times larger on the north sideline of the field (top of the figure) than they are on the south sideline. This is because the terrain elements to the north are roughly 10 meters further from the camera path than those to the south. This closer proximity allows the navigation system to more accurately localize the new feature pairs on the map due to less depth uncertainty, which in turn reduces drift.




(a) Relative Frame



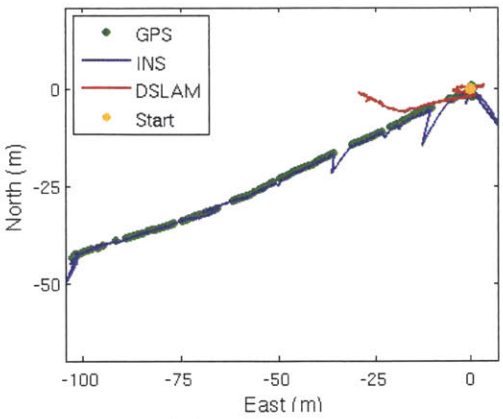
(b) Google Earth

Figure 4-5: Vision (DSLAM) and inertial (INS) navigator results for a trajectory including six 90-degree turns and periodic loss of GPS. Error tends to increment at each corner due to miscalculation of angular rotation while turning, during which the system loses all its persistent features and must rapidly acquire new ones.

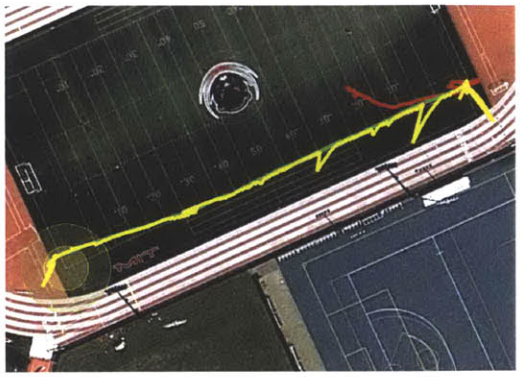
### 4.3.3 Featureless Terrain

The third major source of error in the vision system occurs when FAST is unable to extract unique features within the field of view. This can be caused by either highly patterned or low contrast terrain, both with similar effects. Figure 4-6 shows a trajectory similar to that shown in Figure 4-3, except that the camera is pointing downward at the field from an altitude of about 1 meter. In this situation, FAST is unable to distinguish a sufficient number features to calculate a pose during much of the trajectory.

Figure 4-7 shows two additional examples of the effects of feature loss. First, at the point marked with a black circle, FAST is unable to determine sufficient features as the camera pans over an empty parking lot. Second, at the point marked with a black square, the camera approaches a heavily textured brick wall and then rotates. The system is unable to identify and track unique features from so many detected points, causing roughly 50 degrees of heading error from just one turn. A planetary hopper might encounter a similar situation if it traverses repetitious or featureless terrain.

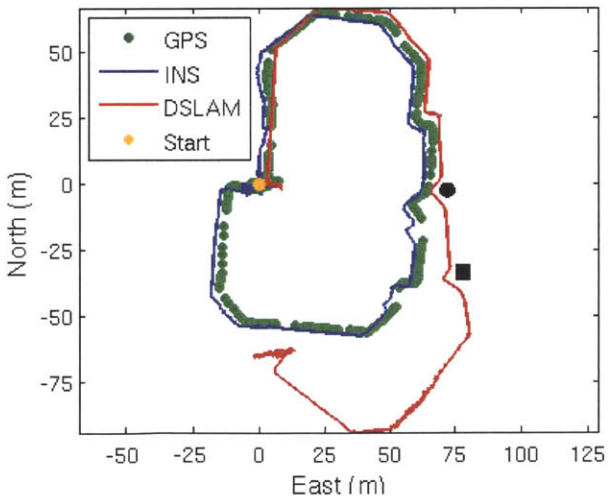



(a) Relative Frame

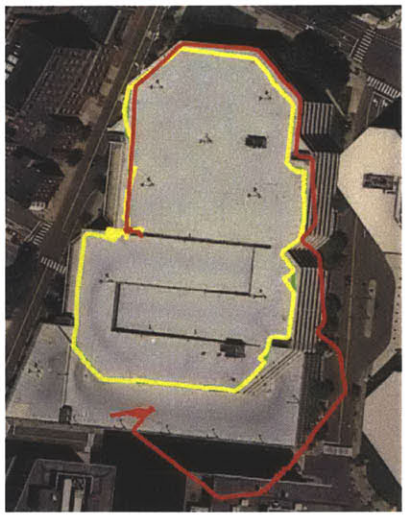


(b) Google Earth

Figure 4-6: Vision (DSLAM) and inertial (INS) navigator results for a straight-line trajectory with a downward-facing camera and periodic loss of GPS. This configuration emphasizes the weakness of the vision system when many features rapidly pass in and out of the field of view of the camera. The system is most accurate when tracking persistent features over long periods.




(a) Relative Frame



(b) Google Earth

Figure 4-7: Vision (DSLAM) and inertial (INS) navigator results for a trajectory circling the perimeter of the top of the Draper parking garage with periodic loss of GPS. The black circle and square show locations where DSLAM momentarily lost all tracked features, inducing especially large errors.

## 4.4 Advantages of the Unified System

Inertial navigation systems work well for high-rate, short-period motion, and are entirely independent of their environment. However, they have poor performance for low-rate, long-period motion, as they can drift uncontrollably within a matter of seconds. Vision navigation systems work well for low-rate, long-period motions, but poorly during rapid rotation. Performance of vision systems is also dependent on characteristics of their operational environments, such as lighting conditions, which cannot be ensured. The aforementioned inertial and stereo vision navigation systems have been combined using an EKF to form a single, unified inertial and vision-based navigation system, which exceeds the navigation performance of either system taken independently.

Figure 4-8 shows the raw data from Figure 4-3 reprocessed using this unified navigation filter. Similarly, Figure 4-9 shows the data from Figure 4-4 processed with the unified filter. Both of these examples show improvements over the individual systems, following the known path of the GPS updates very precisely. Figure 4-8 shows the filter can successfully navigate long segments with distant features, and Figure 4-9 shows the successful navigation of a sharp corner. This unified system offers many advantages over the individual systems by improving IMU bias estimation, reducing IMU drift, handling rapid rotations, and managing periods of sensor outages.

### 4.4.1 IMU Bias Estimation

The IMU often drifts far from its GPS updates, especially before the EKF has had sufficient time to estimate and account for the biases of the three gyroscopes and three accelerometers within the IMU, which typically takes 30 to 60 seconds for this system. During this time, the EKF is using all available data to calibrate itself to each sensor input. The INS system estimates these IMU biases using updates from the GPS receiver, which provides a 3 degree-of-freedom (DOF) position-only update at approximately 1 Hz. From the combination of these position updates and the

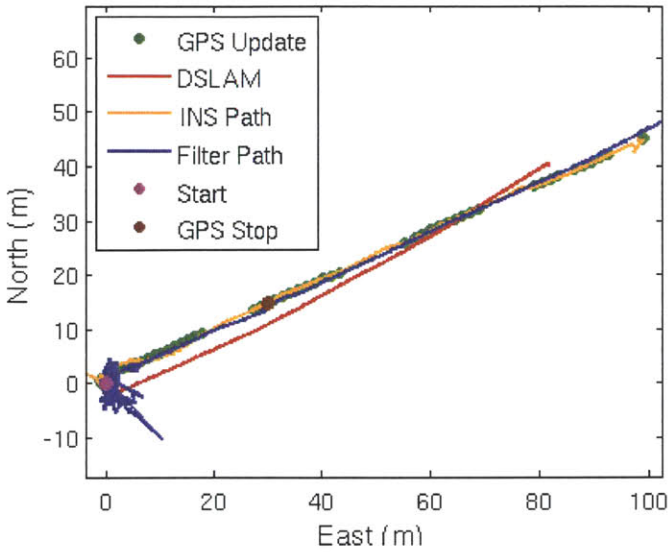



Figure 4-8: Navigation results from the stereo vision (DSLAM), inertial (INS), and unified (Filter) navigation systems for a straight-line trajectory. The unified system accepts GPS updates for self-calibration during the first segment of the trajectory, then GPS updates are turned off and used only as "truth" for comparison. The unified filter succeeds as inhibiting drift in both the IMU and vision systems.

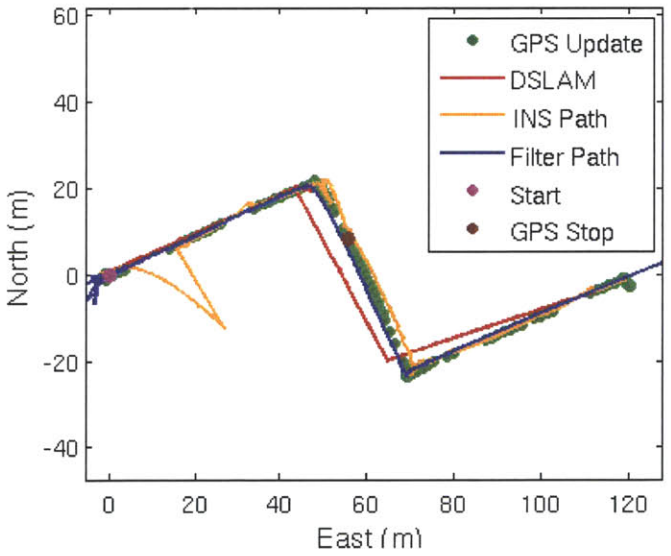



Figure 4-9: Navigation results from the stereo vision (DSLAM), inertial (INS), and unified (Filter) navigation systems. The unified system is successfully able to navigate the 90-degree corner without GPS updates, which were turned off just after the first corner. The inertial-only system drifts significantly during the loss of GPS near the beginning of the trajectory, but the vision updates keep the unified system from exhibiting this same behavior.

measurement of the gravity vector, the EKF can estimate the biases of only 5 of the 6 total DOFs of the IMU. The EKF does not have visibility into the Z-axis (yaw) gyro, as gravity does not project into this rotation axis.

DSLAM outputs a full 6-DOF navigation state at 20 Hz, and is especially precise when completely stationary, such as during the 120-second stationary calibration phase performed prior to any data collection with the system. Because DSLAM updates the filter more frequently, more precisely, and with all 6 DOFs (3 position and 3 attitude), the unified system converges upon the IMU biases more rapidly and more accurately than it does with GPS updates alone. Figure 4-10 shows the EKF gyro bias estimates over time for the same IMU measurements processed with and without vision updates. Not only does the addition of vision data allow proper estimation of the Z-axis gyro, but it also speeds up convergence for the X- and Y-axes. This allows the filter to use the IMU more effectively, improving the navigation performance of the unified system.

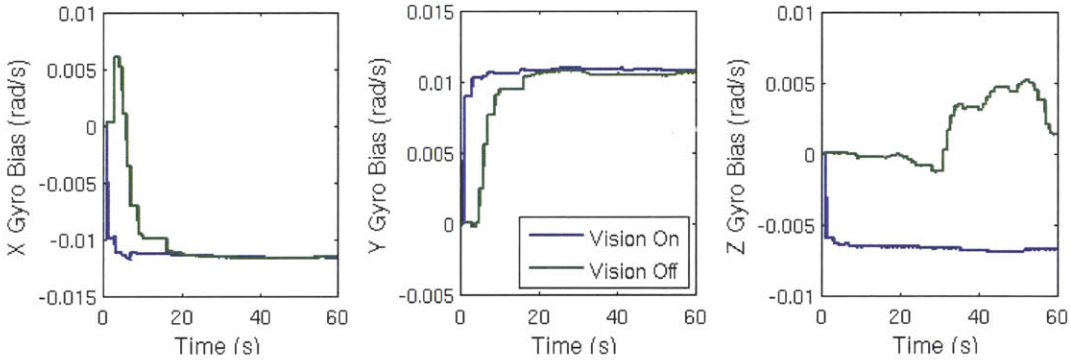



Figure 4-10: Filter-calculated estimates of IMU gyro biases with and without vision updates. By incorporating the vision updates, the filter gains visibility into the Z-axis gyro and is able to more rapidly converge upon the biases of all three gyros.

### 4.4.2 IMU Drift

Due to the numerical integration of sensor noise, IMU drift error grows exponentially with time until constrained by an external sensor update. However, despite growing increasingly erroneous, IMUs have very low drift for brief periods immediately fol-

lowing updates. An IMU receiving frequent updates is capable of providing accurate measurements over a wide range of rotational rates and accelerations. DSLAM can provide these updates more frequently (based on the frame rate of the camera) than GPS, improving both the accuracy and usability of the IMU measurements. Figure 4-11 shows the comparative performance of the IMU being updated by GPS versus by the vision system. Near the start of this path (at the left of the figure), the INS system's heading is inaccurate, requiring several updates to correct, only for it to end up being over-corrected. The unified trajectory (comprising of vision and IMU updates) is updated more frequently, resulting in a significant reduction in the effects of IMU drift.

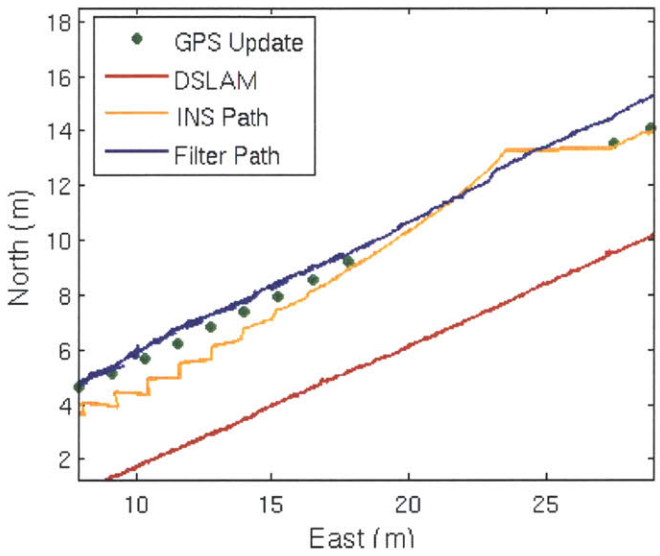



Figure 4-11: The effect of IMU drift when updated using low-rate GPS (orange) versus high-rate vision (blue) updates. The frequent updates from the vision system result in a significant improvement in positioning accuracy.

### 4.4.3 Rotation Rates

As was previously illustrated by Figures 4-4 and 4-5, rapid rotations result in errors in the vision system because the tracked features are too rapidly lost outside of the field of view of the camera. A well-constrained IMU can identify these rotation rates and

help the vision system to either maintain or regain its pose estimate after features are lost or as features are rapidly exiting the field of view, as long as the high-rate motion is brief in duration. In the unified inertial and vision navigation system, the vision system can provide accurate updates to the IMU until the high-rate motion begins, and quickly re-constrain the system again afterwards. As shown in Figure 4-12, this results in improved accuracy in traversing sharp corners.

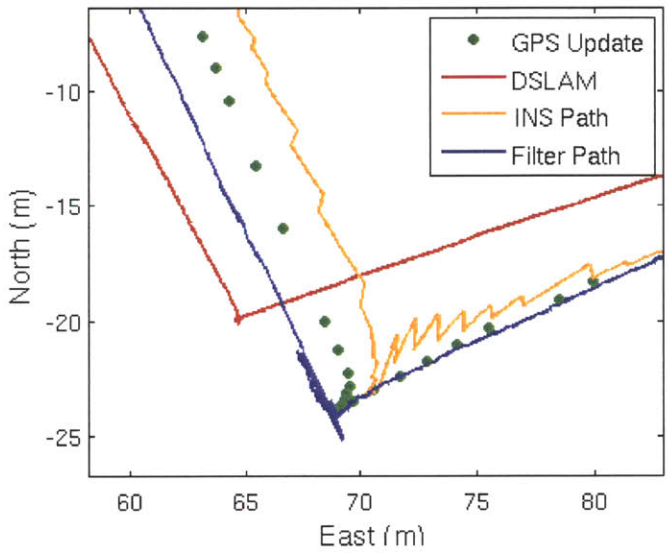
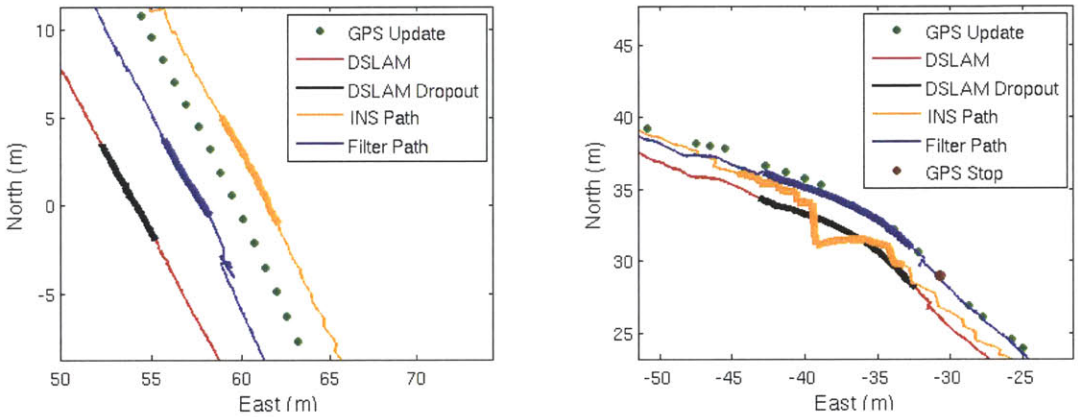



Figure 4-12: The unified navigation filter improves performance during short periods of high-rate motion, such as sharp corners. DSLAM alone measures the corner to be 98°, while the unified system (shown with GPS updates turned off) measures it correctly as 90°. GPS updates are shown for reference.

### 4.4.4 Vision System Dropouts

The worst errors in the vision system occur when all features are lost from the camera field of view, causing the vision system to stop returning motion estimates (often called a "dropout"). This happens mainly in low-contrast environments or due to lighting effects, such as lens flares. However, the IMU always outputs continuously regardless of external conditions. This means that the inertial system can propagate the filter forward during brief outages of the vision system. Figure 4-13a shows an example of a simulated 5 second vision dropout during a straight traverse, and Figure 4-13b shows

a simulated 10 second vision dropout during a gentle curve. Running independently, DSLAM halts motion entirely when it cannot detect enough features to calculate a camera pose estimate, and the INS would typically begin to rapidly drift (note that in these figures the INS is still being constrained by GPS updates, in order to better depict the behavior of the IMU during that time). When running with the combined filter, the IMU bias and drift are better accounted for, allowing the system to continue to navigate during brief periods of complete outage by both the vision and GPS systems.




(a) Straight Segment (5 sec)

(b) Curved Segment (10 sec)

Figure 4-13: Simulated vision system dropouts during a straight (left) and gently curved (right) trajectory. The combined vision and inertial system (in blue) is able to successfully navigate for brief periods in the absence of vision or GPS updates, making the system more robust to inconsistencies and changes in its operational environment. The affected segments are plotted in bold for emphasis. Note that the INS path still uses GPS updates here to constrain its error growth.

# Chapter 5

# Testing & Results

The Stingray navigation testbed, described in Chapter 3, was used to test both the DSLAM stereo vision navigation system and the unified inertial and vision navigation system described in Chapter 4, for their suitability for accomplishing the goals of hopper navigation presented in Chapter 2. This chapter presents a method for measuring the accuracy of calculated trajectories, demonstrates the capabilities of the system in various environments, and provides a discussion of the relative sensitivities of the system to key parameters.

An actual operational hopper was unavailable for testing, due to issues of availability and operational costs. For this reason, the system was designed to be portable and self-contained, making it operational in a variety of alternative contexts, such as that of a pedestrian or a car. Without question, navigating a car driving in an urban environment is different from navigating a hover above the lunar surface. However, the experimental results presented in this chapter were carefully selected for their similarities to various challenges a hopper navigation system must overcome. For example, while pedestrians move slower than hoppers, they have frequent, large disturbances and rapid rotation rates due to the dynamics of walking. A car has slower rotation rates and fewer disturbances, but velocities more comparable to those of hoppers.

Testing the various challenges of hopping in this piecewise fashion provides a good first step towards demonstration of the system in a more directly relevant hopper environment. Future work will involve integrating a modified Stingray system onboard a vehicle with similar dynamics to a hopper, such as a helicopter or terrestrial rocket, in a lunar-analog environment. Although the stereo vision system onboard an operational hopper will point downward, as described in Chapter 2, the majority of these tests were conducted with the camera oriented forward in order to observe features at more realistic depths, as mission designs for hoppers typically involve traversing at least ten meters above the planetary surface.

The inertial system is independent of external influences, and thus performs consistently amongst all of the various test environments and conditions. The performance of the vision system, which calculates a terrain-relative state estimate, is highly dependent on both testing dynamics and environments. The unified system combines the inertial and vision-based solutions, and therefore is dependent on the performance of each. For this reason, this chapter discusses only the vision and unified systems, making the assumption that the IMU-specific performance remains consistent within its specified range of dynamics and environments (with the exception of its calibration, which is provided by the unified system). These codes have not yet been optimized for computational performance, and thus processing time was not measured and is not directly addressed.

## 5.1 Testing Goals

The primary goal of the experimental system tests described here was to determine whether the system does or does not warrant continued development for the purpose of hopper navigation. To accomplish this, the following capabilities required demonstration:

1. Output of a 6 degree of freedom (DOF) navigation state.
2. Measurement of basic planetary hopper motions, including:

(a) Vehicle position and velocities.

(b) Vehicle attitude and rotation rates.

(c) Hover hop trajectory.

3. Navigation in unstructured[1] environments similar to those expected for a planetary hopper.

The second major goal of this testing campaign was to determine the major sources of error in the system. In some cases, these errors can be mitigated by changes in parameters or operational procedures. As part of achieving this goal, the position and attitude accuracy of the system are addressed. Because the only "truth[2]" data available is from GPS, only positional accuracy can be addressed directly. Attitude accuracy is only discussed at a high level. The GPS truth data provides an absolute (as opposed to relative) navigation solution, which does not suffer from the effects of error propagation or long-term drift. However, the unified vision system is significantly more precise than GPS. Future work will include incorporation of more comprehensive truth data for a more detailed assessment of attitude measurement accuracy.

## 5.2 Data Collection

All data presented in this chapter was collected using the Stingray testbed, described in detail in Chapter 3. While this system is capable of outputting a full navigation state in real-time, all of the results presented here were post-processed from the raw data logs. Post-processing the data allows more detailed analysis and plotting of the results, which are identical to those of the real-time system. Because the standard Stingray system does not have a keyboard or display, the computer was replaced with a laptop (running the same software), shown in Figure 5-1a, for added convenience

[1] In vision navigation, an environment is considered "unstructured" when it is not controlled. A structured environment might have identifiable beacons in known locations, such as in a factory, to improve navigational performance.

[2] The truth data in this case provides an absolute (as opposed to relative) navigation solution, eliminating the effects of long-term drift. However, the unified vision system tends to be more accurate than GPS over short periods.

during data collection. This allowed the test operator to monitor the performance of the sensors and restart the test in the event of a system error.

### 5.2.1 Data Collection Process

The first step in data collection is to start the computer, load the software, and position the sensors. The IMU and stereo camera are mounted securely to one another, with the sensor mount being compatible with any of the BumbleBee2 cameras listed in Table 3.2. Two identical IMUs and sensor mounts were used interchangeably for data collection. The GPS antenna was not rigidly mounted to the other sensors, as interference from the camera and IMU interface board required it to be kept at least 20 cm from the sensor mount. For tests conducted using a car, a suction cup camera mount was used to attach the sensors to the inside of the windshield, as shown in Figure 5-1b.

Prior to collection of any data set, the system is calibrated by being left entirely stationary for at least 120 seconds. The vision system itself does not require calibration, but this period is used by the EKF of the unified system to determine the biases of the IMU based on the output of the GPS receiver and vision system. The quality of the calibration is typically further enhanced by leaving GPS updates on for the first 30-60 seconds of a trajectory, especially if this segment includes rotational motion. When consistency amongst datasets collected along the same test route was desired, the trajectory began with a wide circle to provide repeatable rotational motion.

During the actual data collection, there is very little for the operator to do beyond following the planned trajectory. The sensor output rates are displayed on the screen as a high-level diagnostic, though issues with sensor logging are rare. At the conclusion of the test, the operator stops the system, records the unique dataset identifier (i.e., "log stamp"), and then either shuts down the system or restarts the process with a new calibration phase.

(a) Laptop for Data Collection

(b) Camera Mounted to Vehicle

Figure 5-1: Testing hardware for data collection.

### 5.2.2 Data Processing

Collected data is post-processed into a navigation state using a Matlab interface to the C++ application. First, the vision data is processed using the DSLAM GUI, shown in Figure 5-2, which outputs a text file of the pose calculations, as well as some diagnostics used by the filter. A Matlab script then loads this text file of vision poses and logs of the raw GPS and IMU data, processes them using the unified navigation system (implemented as a C++ library), and then provides several options for plotting and analysis of the results. Post-processing the data enables comparison of navigation performance of identical datasets processed using various parameters, as well as operation at nearly twice real-time (depending on parameter selection and system hardware).

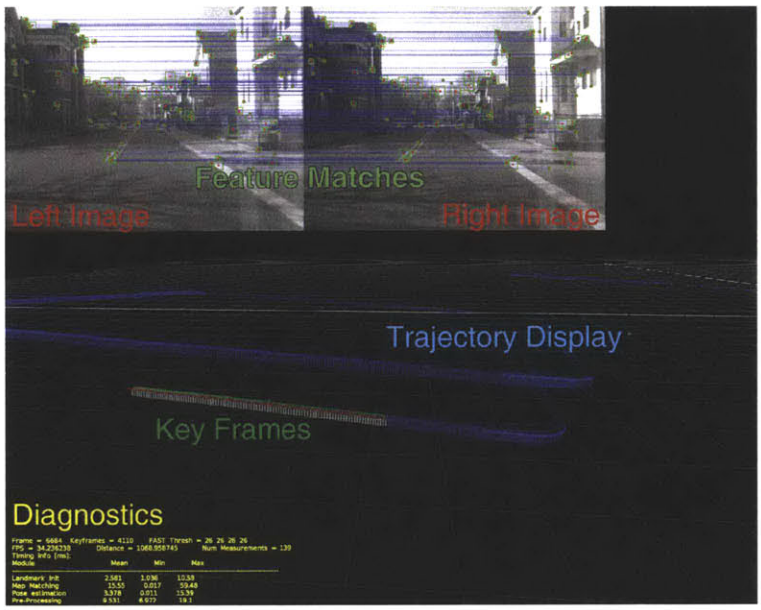



Figure 5-2: The DSLAM graphical user interface.

## 5.3 Description of Test Data

The Stingray system has logged over 500 gigabytes of sensor data since its creation. The results and discussion of system performance presented in this section are primarily based on seven datasets that were selected for their ability to address the goals stated above. These datasets are discussed based on the operational settings of:

1. walking in natural environments,
2. driving in urban environments, and
3. attached to the indoor crane at Draper Laboratory.

### 5.3.1 Walking in Natural Environments

Two datasets were collected while walking outdoors at Halibut Point State Park, near Rockport, Massachusetts. The first route, referred to as "Halibut-Trail" and shown in Figure 5-3, follows a hiking trail through the park for 790 meters. This dataset includes the largest elevation change and largest range of observable feature depths of all the datasets presented. The hiking trail includes natural terrain ranging from

an open area to a forest, with several gentle curves.

(a) Terrain Image (b) Google Earth View

Figure 5-3: Typical terrain and Google Earth path for Halibut-Trail dataset, which involves walking on a wooded hiking trail.

The second route, referred to as "Halibut-Shore" and shown in Figure 5-4, follows the rocky Atlantic shoreline for 180 meters at low tide. The only terrain visible in this dataset is large rocks, making it the most monotonous of those presented, and it is also the most lunar-like. It also includes several large disturbances and unsteady motion due to the uneven terrain being traversed by foot.

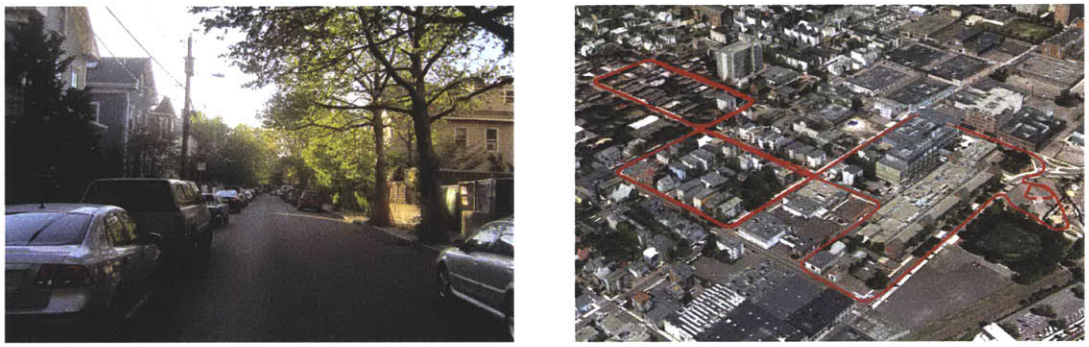
(a) Terrain Image (b) Google Earth View

Figure 5-4: Typical terrain and Google Earth path for Halibut-Shore dataset, which involves walking on rocky terrain.

### 5.3.2 Driving in Urban Environments

Two datasets were collected while driving a similar route through a residential area of Cambridge, Massachusetts. This 3.6 km route, shown in Figure 5-5, includes the

largest velocities of all the routes. These datasets are referred to as "Cambridgeport" and "Cambridgeport-Wide," and were collected using the 43-degree and 66-degree field of view cameras, respectively, and are additionally used to compare the performance of the two cameras. This route is unique because it includes three laps around a single block with only right turns followed by three laps around an adjacent block with only left turns, and it starts and ends at the same location. This makes it especially useful for evaluating cornering performance and consistency.



(a) Terrain Image (b) Google Earth View

Figure 5-5: Typical terrain and Google Earth path for Cambrideport dataset, which involves driving in an urban residential neighborhood.

### 5.3.3 Indoor Crane at Draper Laboratory

The Draper Laboratory in Cambridge, Massachusetts has an indoor crane, shown in Figure 5-6, which was used to simulate basic hopper motions. Due to the large disturbances and vibrations caused by moving the crane (larger than those expected of an actual hopper), the IMU is too noisy to be beneficial to the system. Therefore, these results are presented using only the vision system. Future tests and development will address this unacceptable level of sensor noise, as well as determine the operational vibration limits of the system.

The crane was used to collect three datasets, all of which utilize a downward-facing camera in a similar configuration to that planned for use onboard a hopper. "Crane-Roll" comprises a roll rotation (change in heading) without any altitude change.

(a) Indoor Crane

(b) Crane Test Environment

Figure 5-6: An indoor crane at Draper Laboratory was used to simulate small-scale hopper motions.

"Crane-Ascent-Descent" is an ascent, hover, and descent maneuver. "Crane-Hop" performs a small-scale hover hop, including a roughly 1 m ascent, 9 m lateral traverse, and 1 m descent. All of these datasets include large, impulsive disturbances, as the crane was not designed for slow or gradual motion. Additionally, because the High Bay is a relatively small room, the majority of the observable features were within 1-2 meters of the camera, limiting the effectiveness of testing. The Crane datasets are useful for demonstration of hopper maneuver navigation, but were not used for extensive analysis or parameter testing due to their many additional sources of error and lack of GPS truth data.

## 5.4 Measuring & Quantifying Error

The Stingray system calculates a full 6 degree-of-freedom (DOF) pose estimation plus velocities and angular rates in all axes, but has only 3-DOF positional truth measurements available from GPS. This means that, although Stingray calculates position, velocity, attitude, and angular rates for each axis (12 measurements in all), only the accuracy of positional performance (3 measurements) can be evaluated directly. Full 6-DOF truth measurements are difficult to obtain, especially in an outdoor setting over long trajectories. Evaluation of the full system performance would require either purchase or development of a similar navigation system with higher performance,

defeating the purpose of developing Stingray.

### 5.4.1 Estimating Error

Due to the nature of the EKF in the unified navigation system, the error of all estimated states is closely coupled. This means that successful estimation of position, which can be compared to truth, indicates successful estimation of velocity, attitude, and angular rate, which cannot. At this time, the only other means of evaluating the full system performance is by visual inspection of plots of the data and comparing it to the raw video feed. Due to their relative simplicity, rough measurements of roll angle and altitude can be made for the "Crane-Roll" and "Crane-Ascent-Descent" datasets. However, the navigation system is more precise than these hand calculations, and deviations from these measurements do not necessarily indicate errors.

### 5.4.2 The Error Vector Sum Metric

A metric was developed in order to objectively assess the positional accuracy of calculated trajectories. This metric, called the Error Vector Sum (EVS) metric, can be used either to compare multiple trajectories from the same dataset processed using different parameters, or to identify locations within a trajectory where the system encounters difficulties. The EVS metric summarizes system performance into a single numerical value, which can then be used to compare navigational performance of multiple trajectories or to determine the relative sensitivity of the system to various parameters. The EVS is not a perfect, all-encompassing measure of system performance, but it does provide a useful tool for comparing navigation solutions.

The EVS metric is the sum of incremental errors of a measured trajectory with respect to some reference, normalized by the total distance traveled, as measured by the reference. With relative navigation systems, a small heading error early in a trajectory can appear as a large error late in the trajectory. This means that if error is calculated simply as the distance from a particular point to truth, errors occurring

early in a trajectory would be weighted significantly higher than those occurring later in a trajectory. The EVS metric eliminates this problem by first removing the influence of previous measurements in the navigation system, and then calculating the error only for only a specific increment of the trajectory. Additionally, calculating the error independently for each small segment allows the incremental error to be tracked over time, providing insight into sources of error in the system. The lengths of these increments are determined by the frequency and availability of the reference data, in this case GPS measurements, which are typically recorded at an average value of about 0.8 Hz.

Because GPS has its own uncertainty and error, a trajectory will likely never have a "perfect" EVS metric of 0, as the error is often mistakenly inflated as a result of GPS errors. A number of heuristics have been additionally implemented in the EVS calculation algorithm to reduce the frequency of these cases. For example, the GPS receiver has a built-in motion model that causes it to grow very noisy when it is stationary (such as during calibration periods), but DSLAM is generally very accurate when stationary. To account for this, the EVS algorithm trusts DSLAM or the unified filter when it returns zero change in motion, unless DSLAM is reporting an error indicating that it is unable to calculate motion at that time (e.g., not enough features are being tracked).

### 5.4.3 EVS Metric Calculation

The Error Vector Sum (EVS) metric is a measure of how similar two 3-dimensional paths of points are. The EVS is the sum of errors calculated from short intervals, called "incremental EVS," which require only the current point and the two that preceded it for calculation. The algorithm presented here can be implemented either for post-processing of trajectories or for real-time error tracking, potentially allowing parameter adaptation during operation when reference information is present, such as in environments with intermittent GPS availability.

The EVS grows increasingly indicative of path similarity as the intervals between points grow smaller. Therefore, the intervals are always selected to match the frequency of the slower data stream, which is the GPS data in this case. First, point correspondences are established by matching each "reference" time stamp with the closest "trajectory" timestamp, as shown in Figure 5-8. The incremental EVS is calculated using the three most recent matches, from which two vectors are formed, as shown in Equation 5.1.

$$\mathbf{r}^k = \langle ref_x^k - ref_x^{k-1}, ref_y^k - ref_y^{k-1}, ref_z^k - ref_z^{k-1} \rangle \tag{5.1a}$$

$$\mathbf{r}^{k-1} = \langle ref_x^{k-1} - ref_x^{k-2}, ref_y^{k-1} - ref_y^{k-2}, ref_z^{k-1} - ref_z^{k-2} \rangle \tag{5.1b}$$

$$\mathbf{t}^k = \langle traj_x^k - traj_x^{k-1}, traj_y^k - traj_y^{k-1}, traj_z^k - traj_z^{k-1} \rangle \tag{5.1c}$$

$$\mathbf{t}^{k-1} = \langle traj_x^{k-1} - traj_x^{k-2}, taj_y^{k-1} - traj_y^{k-2}, traj_z^{k-1} - traj_z^{k-2} \rangle \tag{5.1d}$$

The incremental EVS is a measurement of the error over only the current interval, independent of any previous trajectory errors. This independence is achieved by calculating the orientation of the previous trajectory vector, $\mathbf{t}^{k-1}$, with respect to the previous reference vector, $\mathbf{r}^{k-1}$, and then rotating the current trajectory vector, $\mathbf{t}^k$, by the same amount. The incremental error is the magnitude of the vector difference between this adjusted trajectory vector and the current reference vector.

Because the individual points do not have orientations, the rotation requires only two angles: yaw and pitch, which are calculated for the previous interval of both the reference and trajectory as shown in Equations 5.2 and 5.3. The relative yaw and pitch angles are simply the differences between the individual angles, as shown in Equation 5.4.

$$\psi_r^{k-1} = \arctan \frac{\mathbf{r}_y^{k-1}}{\mathbf{r}_x^{k-1}} \tag{5.2a}$$

$$\psi_t^{k-1} = \arctan \frac{\mathbf{t}_y^{k-1}}{\mathbf{t}_x^{k-1}} \tag{5.2b}$$

$$\theta_r^{k-1} = \arcsin \frac{\mathbf{r}_z^{k-1}}{\|\mathbf{r}^{k-1}\|} \tag{5.3a}$$

$$\theta_t^{k-1} = \arcsin \frac{\mathbf{t}_z^{k-1}}{\|\mathbf{t}^{k-1}\|} \tag{5.3b}$$

$$\psi^{k-1} = \psi_t^{k-1} - \psi_r^{k-1} \tag{5.4a}$$

$$\theta^{k-1} = \theta_t^{k-1} - \theta_r^{k-1} \tag{5.4b}$$

Rotation matrices, shown in Equation 5.5, are then formed from the yaw and pitch angles, and used to transform the current trajectory vector into the adjusted trajectory vector, $\mathbf{a}^k$, as shown in Equation 5.6. The incremental error vector, $\mathbf{e}^k$, is the vector difference between the adjusted trajectory vector and the current reference vector. Figure 5-7 shows an example of the six vectors involved in the incremental EVS calculation.

$$R_\psi = \begin{bmatrix} \cos\psi^{k-1} & \sin\psi^{k-1} & 0 \\ -\sin\psi^{k-1} & \cos\psi^{k-1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.5a}$$

$$R_\theta = \begin{bmatrix} \cos\theta^{k-1} & 0 & \sin\theta^{k-1} \\ 0 & 1 & 0 \\ -\sin\theta^{k-1} & 0 & \cos\theta^{k-1} \end{bmatrix} \tag{5.5b}$$

$$\mathbf{a}^k = R_\theta R_\psi \mathbf{t}^k \tag{5.6}$$

$$\mathbf{e}^k = \mathbf{a}^k - \mathbf{r}^k \tag{5.7}$$

The scalar EVS metric is the sum of the magnitude of each incremental error vector

divided by the magnitude of each reference vector, as shown in Equation 5.8.

$$EVS = \sum_{k=3}^{n} \frac{\|\mathbf{e}^k\|}{\|\mathbf{r}^k\|} \tag{5.8}$$

In theory, the dimensionless EVS metric can be used to compare performance of trajectories calculated using different datasets. However, the integrity of the metric is dependent on the precision of the reference data, which adds "noise" to the measurement, making it difficult to compare paths calculated from dissimilar datasets. In fact, the magnitude of the GPS uncertainty often exceeds the magnitude of the incremental EVS for both DSLAM and filter-calculated trajectories, which are shown to scale in meters in Figure 5-8. This noise is specific to a given reference path, and therefore effectively cancels out when multiple trajectories are compared to the same reference path.

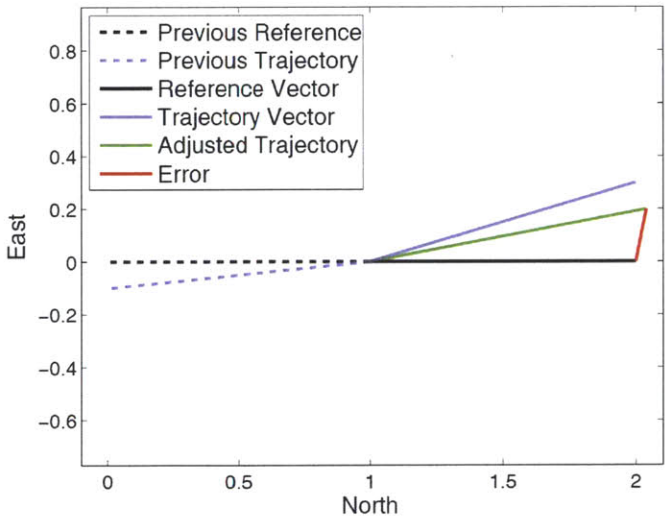



Figure 5-7: The vectors used for EVS calculation.

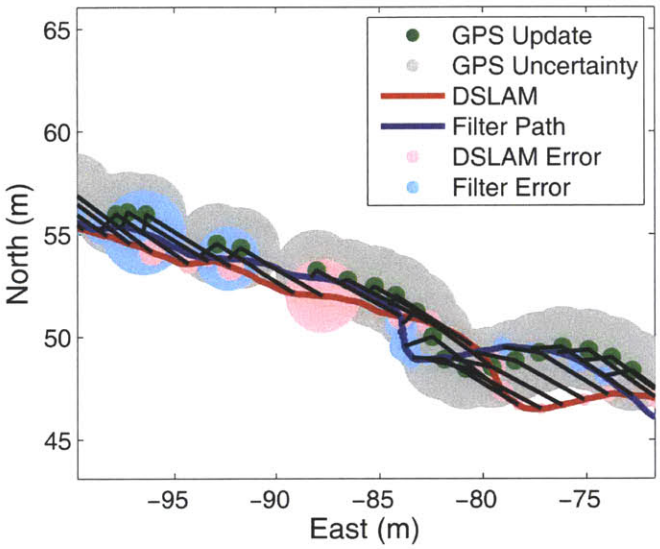



Figure 5-8: The short black lines show correspondences between the GPS updates and the DSLAM and filter trajectories. Trajectory errors (with respect to GPS) are related to changes in length and heading of these lines of correspondence, and can be measured using the incremental Error Vector Sum (EVS) values (shown in pink and light blue). If the system is perfectly accurate, these lines will all be identical. However, this is rare due to the relatively large uncertainty of the GPS positions, shown in gray, which adds considerable noise to the EVS.

## 5.5 Walking Tests

Both the Halibut-Trail and Halibut-Shore datasets described above involved walking outdoors in a natural environment. These datasets were processed using both the vision-only and unified inertial and vision extended Kalman filter (EKF). Each DSLAM trajectory, which was then directly input into the unified filter, was calculated using the parameters shown in Table 5.1. The effects of varying these parameters are discussed in greater detail in Section 5.7.

Various filter parameters, such as measurement covariances and sensor update rates, were tuned independently for each dataset in order to best show the capabilities of the system. An operational lunar hopper would have a more consistent environment, and these parameters would be carefully tuned for that specific environment. The EVS metrics calculated for the DSLAM and filter trajectories are shown in Table 5.2.

| DSLAM Parameter | Value | Description |
|---|---|---|
| Maximum Features | 150 | Maximum number of image features to attempt to match to map landmarks per frame. |
| Minimum Landmark Depth | 0.5 m | Minimum depth for a detected feature to be initialized as a new landmark in the map. |
| Maximum Landmark Depth | 1000 m | Maximum depth for a detected feature to be initialized as a new landmark in the map. |
| Window Length | 10 Key Frames | Length of time a landmark persists within the map and can be matched to features. |

Table 5.1: DSLAM parameters used to calculate trajectories for Halibut-Trail, Halibut-Shore, and Cambridgeport datasets.

| Trajectory | Halibut-Trail | Halibut-Shore | Cambridgeport |
|---|---|---|---|
| DSLAM | 0.248 | 0.241 | 0.243 |
| Unified filter | 0.296 | 0.338 | 0.163 |

Table 5.2: DSLAM and unified filter error metrics for walking and driving trajectories.

### 5.5.1 Halibut-Trail Dataset

"Halibut-Trail" is a 790 meter walk on a wide hiking trail with a moderate elevation change. Figure 5-9 shows both the vision-only and unified filter trajectories, as well as the GPS reference data. As expected, the trajectories both start out very accurately then begin to diverge as time passes due to error accumulation. GPS updates were used by the filter for calibration during the first 83 meters (10%) of the trajectory. In the figures, the magenta "GPS Stop" marker indicates where GPS is disabled.

Figure 5-10a shows the magnitudes of the incremental EVS overlaid on the trajectories, which highlights the main points of divergence. Near the last curve of both trajectories, a particularly large error is caused by camera glare, causing the only major divergence. Prior to that point, the system successfully navigates several difficult situations, such as brief vision outages and curves well over 90-degrees.

Figure 5-10b shows the elevation profile of both trajectories, which is far less accurate than the North/East position shown by the overhead view in Figure 5-9. These

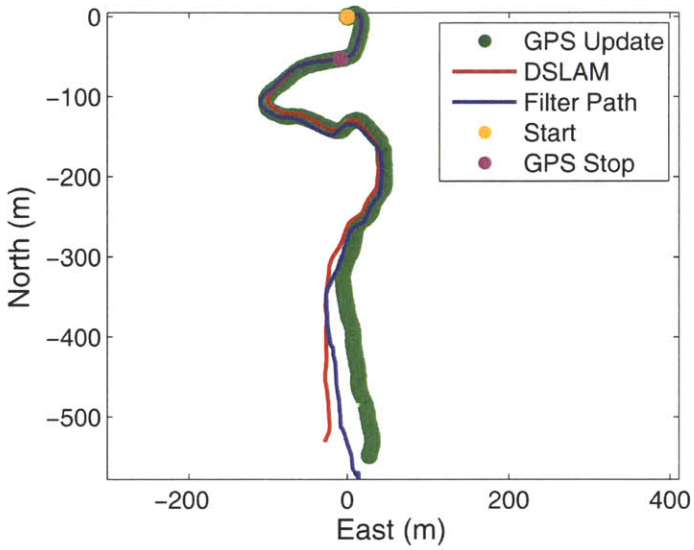
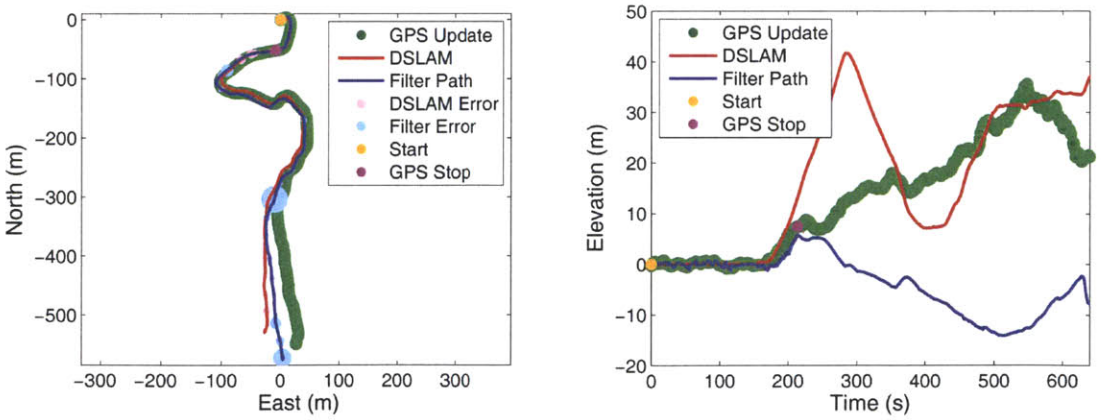



Figure 5-9: DSLAM and unified filter-calculated trajectories for the Halibut-Trail dataset, which was collected while walking on a wooded hiking trail.




(a) Error Locations

(b) Elevation Profile

Figure 5-10: DSLAM and unified filter-calculated trajectories for the Halibut-Trail dataset showing incremental EVS error contributions (left) and elevation profiles (right).

large discrepancies are partly caused by an imperfect initialization. For example, GPS has far greater vertical uncertainty and drift than lateral uncertainty, which makes the initial bias estimation less accurate for the vertical accelerometer. An additional source of error in the vertical direction is that the camera images have only 384 vertical pixels, compared to 512 horizontal pixels, allowing landmarks to be tracked longer during rotations to the left or right rather than upward or downward. The wide angle camera also allows landmarks to be distributed more widely horizontally than vertically, where more widely distributed landmarks increase the accuracy of the pose calculation.

### 5.5.2 Halibut-Shore Dataset

"Halibut-Shore" is a 177 meter walk through rocky terrain. The features in view of the camera during the vast majority of the trajectory are only rocks, making it more analogous to a lunar environment than the wooded or urban environments of the other datasets. GPS updates were used for additional calibration for the first 36 m (20%) of the filter trajectory. As shown in the trajectory plot in Figure 5-11, both DSLAM and the unified filter were very successful at navigating this terrain. Table 5.2 shows the EVS metrics of the trajectories, Figure 5-12a shows the incremental EVS overlaid on the trajectories, and Figure 5-12b shows the elevation profiles of the trajectories.

For this dataset, both the vision-only and filter trajectories were accurate to within a few meters at all times. This accuracy is partly due to the rocky features being perfectly stationary (unlike trees or grass, which often have small motions due to wind that cause measurement noise). The rocky terrain provides an abundance of high-contrast features which are guaranteed not to move, allowing accurate matching of features to existing landmarks within the map.

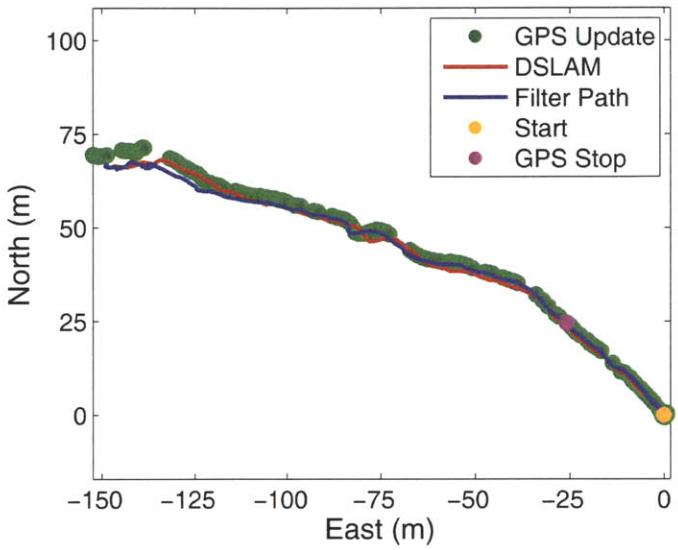
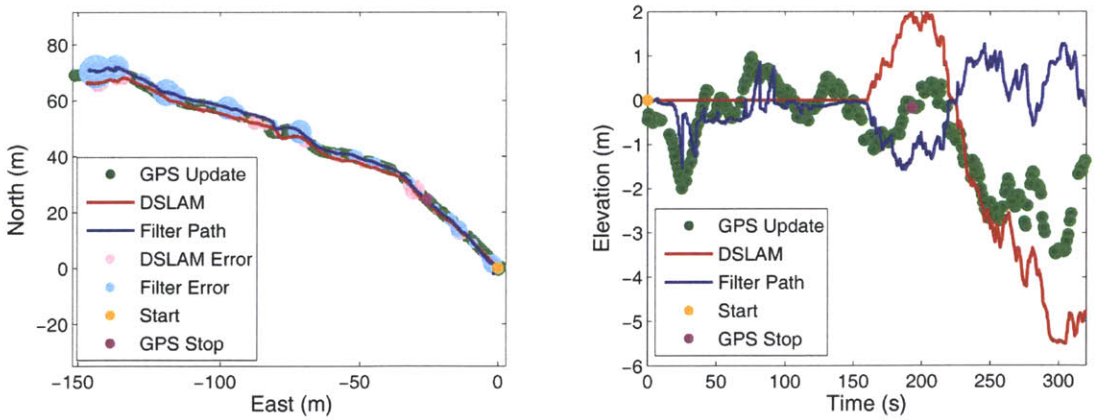



Figure 5-11: DSLAM and unified filter-calculated trajectories for the Halibut-Shore dataset, which was collected while walking on rocky terrain.




(a) Error Locations

(b) Elevation Profile

Figure 5-12: DSLAM and unified filter-calculated trajectories for the Halibut-Shore dataset showing incremental EVS error contributions (left) and elevation profiles (right).

## 5.6 Driving Tests

"Cambridgeport" is a 3.6 km drive through an urban residential neighborhood, with typical vehicle speeds ranging from 7-9 m/s (about 20 mph), and including 28 90-degree turns. The initial calibration included GPS updates for the first 275 m (7.5%) of the trajectory and a full 360-degree circle while in GPS coverage to help calibrate the inertial sensor. Figure 5-13 shows the DSLAM and filter trajectories for this dataset, and their EVS metrics are given in Table 5.2.

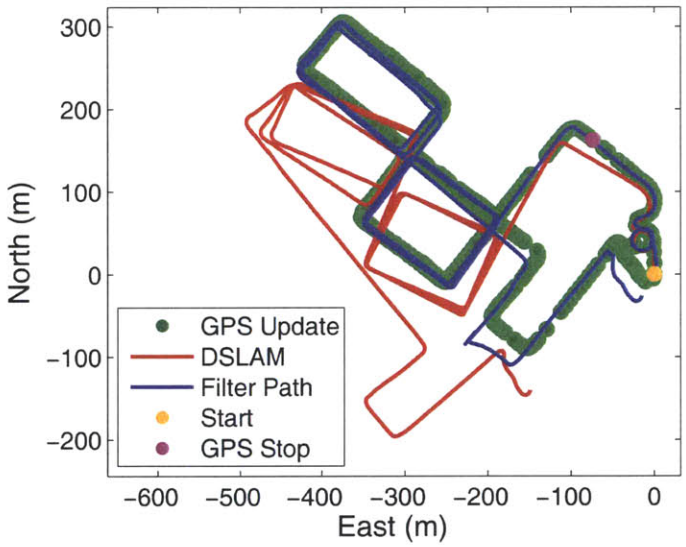



Figure 5-13: DSLAM and unified filter-calculated trajectories for the Cambridgeport dataset, which was collected while driving in an urban environment.

For the driving tests, the camera and IMU were fixed rigidly to the windshield of a passenger car. This led to very smooth motions of the camera and less "noise" in the IMU compared to the dynamics of walking and holding the camera unsteadily. This resulted in the IMU being more reliable for the Cambridgeport dataset than either of the Halibut datasets.

### 5.6.1 Navigation Performance

Due to the improved IMU accuracy when collecting data using the car, the threshold for vision frame acceptance by the unified filter could be set especially high, meaning vision updates with higher effective uncertainties were not used. As shown in Figure 5-14, the DSLAM and filter trajectories register errors in different locations, indicating that the filter is successfully rejecting the majority of the erroneous vision frames, which serves to improve the accuracy of the filter. For example, rejecting more frames results in significantly improved navigation performance at the northernmost corner, where fewer features were tracked due to solar glare on the camera lens during the turn.

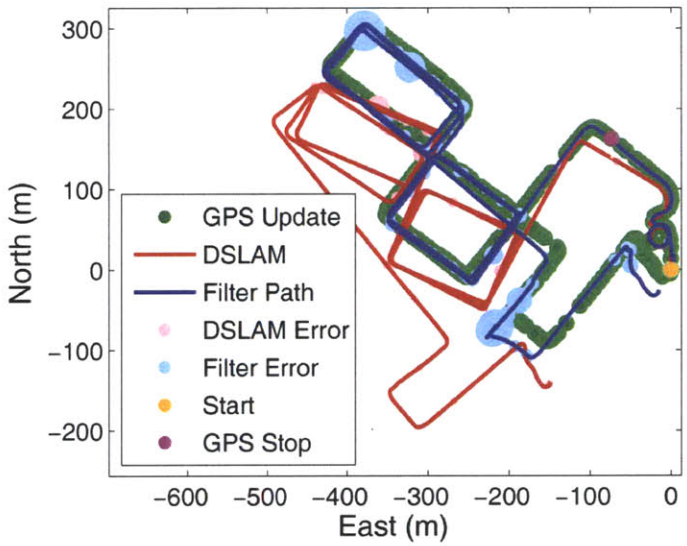



Figure 5-14: DSLAM and unified filter-calculated trajectories for the Cambridgeport dataset with incremental EVS error overlaid on the trajectories to show the locations of the largest contributions to the trajectory error metrics.

Especially with the unified navigation filter, the system exhibited consistent performance while navigating repeated turns and moderate velocities. These motions are more consistent with those planned for hoppers, with expected velocities in the range of 5-20 m/s, despite occurring in a different environment and operational context. These tests demonstrate that the navigation system is capable of accurately navigat-

ing within this range of motion, which is an important step towards preparing for a future flight demonstration of the system.

Figure 5-15 shows the total system speed, rotation rate, and incremental EVS plotted over time during the trajectory. In this figure, 28 spikes in total rotation rate correspond to 28 noticeable drops in velocity at times that align with the 28 corners navigated by the system. Although direct reference data is unavailable for velocity and rotation rate, alternative analysis methods, such as comparison of the results to those expected of the trajectory, provide strong indications that the system is correctly estimating these states.

The EVS is generally very low for this trajectory, with the exception of a few large spikes. The largest incremental EVS measurements, occurring about 500 seconds into the trajectory, are actually the result of GPS errors rather than filter errors, most likely caused by multipath. The other large EVS values, at about 750 seconds into the trajectory, occurred during an extended period of camera glare during one of the corners, during which all vision frames were rejected for an extended period.

### 5.6.2 Elevation Profiles

Figure 5-16a shows the elevation profiles for the Cambridgeport trajectories. Despite being erroneous in scale, the DSLAM elevation profile does follow the contour of the GPS profile more accurately than it did in the Halibut datasets. This is most likely the result of the steadier dynamics of the car compared to the pedestrian. However, the filter receives a particularly bad altitude calibration from GPS in this dataset. The system is stationary for the first 120 seconds, and then the calibration loop, which was conducted in a level parking lot, completes at around 200 seconds. This means the sensors are known to have remained at a very nearly constant altitude during this time, although GPS is shown to be drifting downward erroneously. This inaccurate GPS drift causes the Z-axis accelerometer to become incorrectly calibrated, leading to the system steadily drifting downward throughout the trajectory.

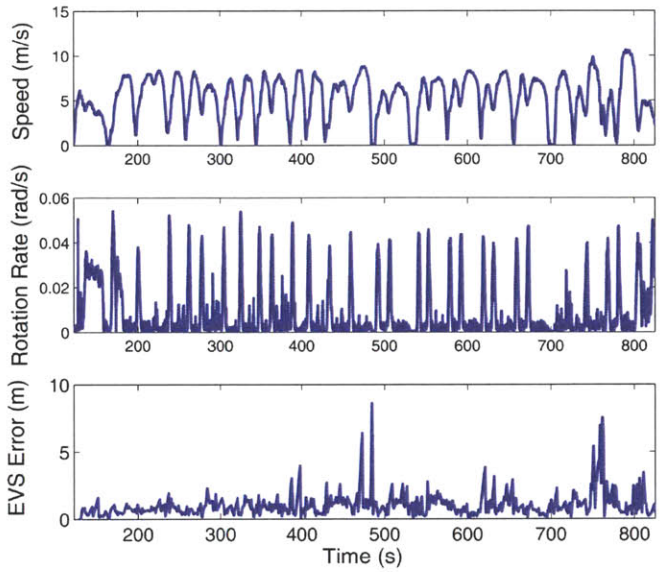
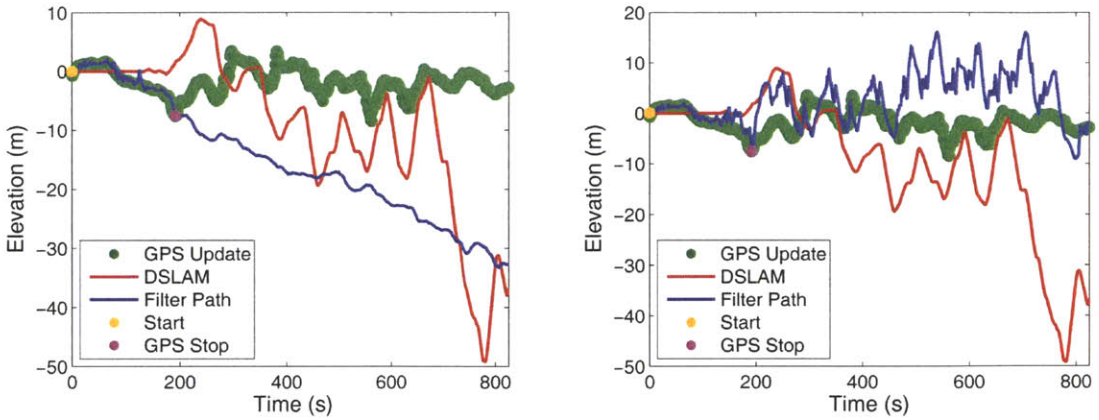



Figure 5-15: Total measured system speed, rotation rate, and incremental EVS for the Cambridgeport dataset. Comparing these values to those expected for a given trajectory serves to further validate system performance. The incremental EVS of the unified filter-calculated trajectory is never consistently zero, in part due to noise in the GPS "truth" measurements.

Figure 5-16b shows the elevation profile of the trajectory calculated using additional altitude updates from the barometric altimeter on the IMU interface board. Including even these highly uncertain altitude updates significantly improves the accuracy of the navigation filter. Although barometric altimetry is unavailable on airless bodies such as the Moon, a laser altimeter could easily provide similar updates at much higher precision, further constraining the altitude drift.




(a) No Altimetry (b) Barometric Altimeter

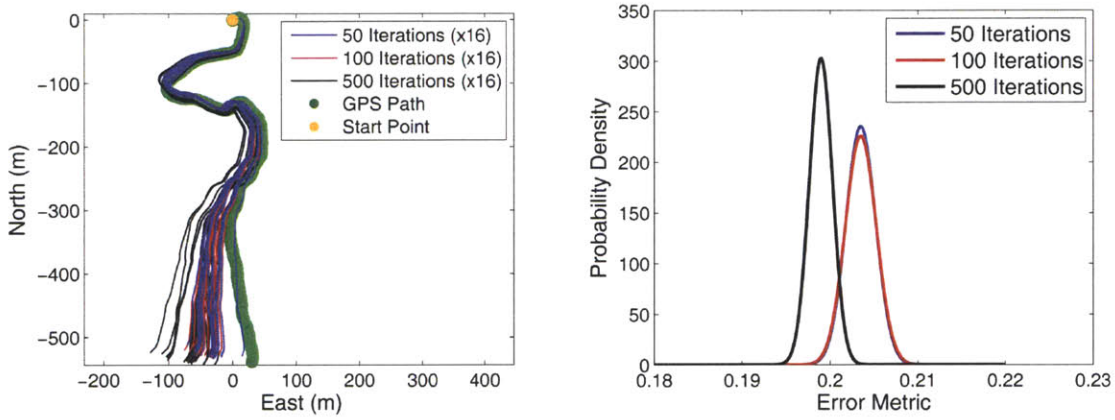
Figure 5-16: DSLAM and unified filter-calculated elevation profiles for the Cambridgeport dataset, calculated with and without the onboard barometric altimeter. Use of additional altimeter data helps to reduce elevation drift, which in this case was partly caused by a poor GPS calibration.

## 5.7 Parameter Testing

Both DSLAM and the unified system filter have many parameters that can be varied to "tune" their performance to various operational environments. Evaluating system response to a range of parameter values offers insight into sources of error and allows some extrapolation of navigation performance characteristics to hopping and the lunar environment beyond what was able to be tested directly. While there are too many of these parameters to test all of them in detail, a few DSLAM parameters were selected for further study based on their applicability to planetary hopping: the number of features tracked in the field of view, the minimum and maximum depths of landmarks added to the map, and the length of time that landmarks remain in the map. The system's response to changes in these parameters is presented below, as well as a discussion of their meaning in the context of a planetary hopper navigation system.

### 5.7.1 Effects of Non-Determinance

The effect of the number of RANSAC iterations the system performs should be mentioned prior to any discussion of system parameter sensitivity. As was discussed in Section 4.2, DSLAM uses three-point-pose RANSAC to repeatedly calculate the transformation between three randomly chosen feature pairs and the landmarks they were matched to in the map. This generates a large, noisy set of pose estimations, from which RANSAC determines a set of best-fit inliers. If the system were to test every possible combination of three features from the total set of matched features, thousands of iterations would be required. To speed up this process, the algorithm uses a random subset of the points to reduce the number of calculations performed.



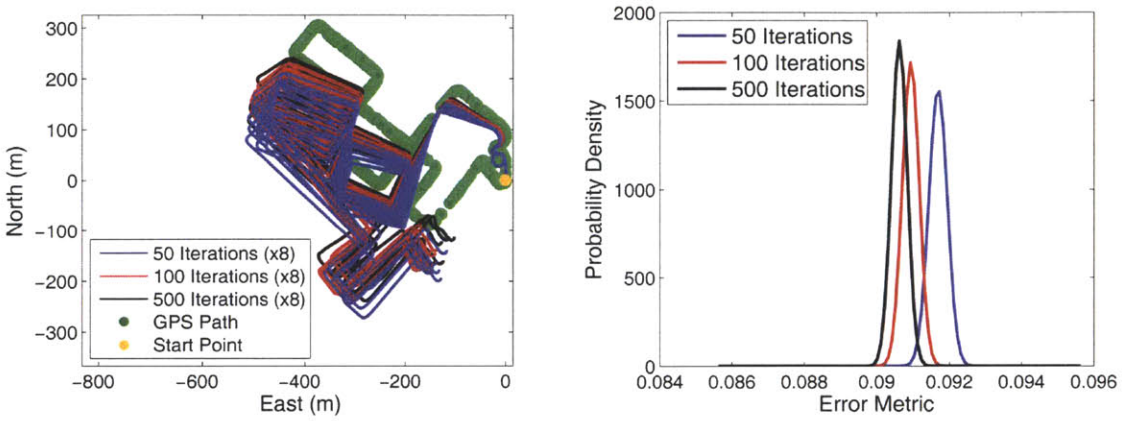

(a) DSLAM Trajectories

(b) Error Distribution

Figure 5-17: The Halibut-Trail dataset processed 16 times for each of three different numbers of RANSAC iterations to show the effect of the internal non-deterministic algorithm. The resulting EVS metrics of these trajectories were then fit to normal distributions, shown at right.

Figures 5-17a and 5-18a show the Cambridgeport and Halibut-Trail datasets processed repeatedly using 50, 100, and 500 minimum RANSAC iterations to show the effects of nondeterminism on system performance. Figures 5-17b and 5-18b show the probability distributions of the EVS metrics for these trajectories. These figures show that some deviation will always occur in the system, and that small differences in calculated trajectories might be the result of the randomness induced by RANSAC. As these plots show, the standard deviation of the EVS metric tends to slowly decrease

as more RANSAC iterations are used. A significant noise component is induced due to RANSAC even when processing identical data, and small changes in parameters should not be taken to necessarily indicate sensitivity of the system to the parameter. All of the trajectories presented in this chapter were calculated using 100 RANSAC iterations to calculate each frame, which provides a good balance between accuracy and computational load.



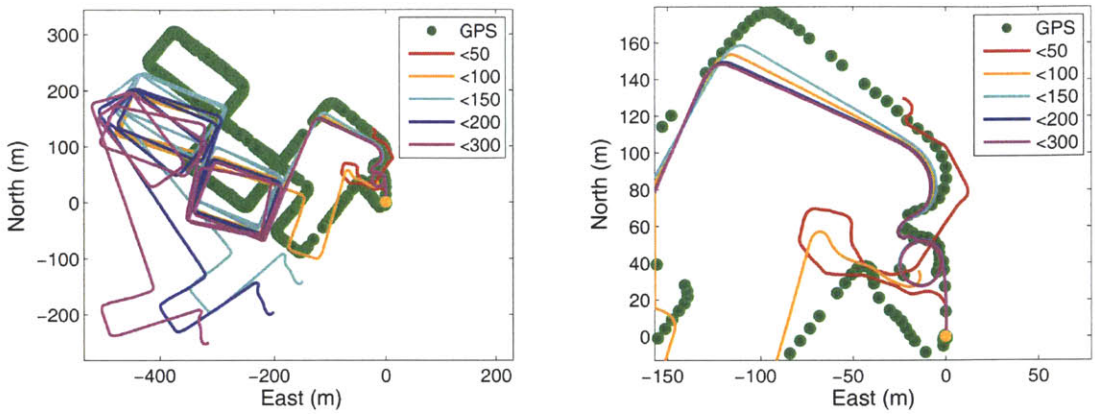

(a) DSLAM Trajectories

(b) Error Distribution

Figure 5-18: The Cambridgeport dataset processed 8 times for each of three different numbers of RANSAC iterations to show the effect of the internal non-deterministic algorithm. The resulting EVS metrics of these trajectories were then fit to normal distributions, shown at right.

### 5.7.2 Number of Features Tracked

It makes intuitive sense that tracking additional features in a scene should lead to more landmark matches and therefore greater accuracy (i.e., a smaller EVS metric). However, the quality of those features is also important. Adding an additional hundred bad features, such as moving points, weak matches, or uncertain locations, often harms accuracy by throwing off the mean, especially if these points are clustered together. DSLAM internally first weights the quality of all detected features and orders them in a list, so each marginal tracked feature will be less accurate than a feature already being tracked. The goal is to find the point at which adding additional features no longer benefits the system, as computational speed is also heavily influenced by




(a) Cambridgeport

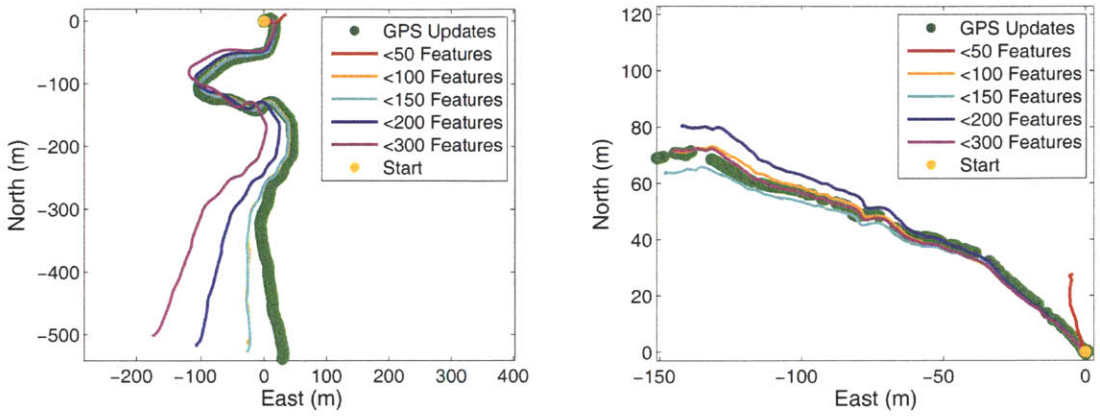
(b) Cambridgeport - Close-up

Figure 5-19: DSLAM trajectories calculated from the Cambridgeport dataset using increasing numbers of tracked features. As shown at right, the system is unable to navigate even that initial calibration circle when only allowed to track 50 features.

the number of tracked features. Figure 5-19 shows the results of processing the urban Cambridgeport dataset five times with increasing numbers of tracked features. Figure 5-20 shows the results of similarly processing the two Halibut datasets. Figure 5-21 shows the EVS metrics for the various numbers of features.

These results make it clear that 50 features are not sufficient for accurate navigation. Additionally, the figures show that the system begins to lose accuracy as the number of features grows especially large. The ideal number of features to track was found to typically be around 150, based off observing the performance of the trajectories plotted in the figures. However, as the EVS shows, the average system accuracy is generally insensitive to this parameter, and the errors tend to occur at a few difficult points, such as times of bad camera glare or rapid motions.

The number of features tracked is an important parameter for testing the applicability of such a system to hopping. The lunar environment has generally fewer features available to track than the environments tested on Earth. This can be compensated for somewhat by reducing the number of features tracked by the system. The robustness to this parameter demonstrated here is important to any vehicle operating in an uncertain environment.




(a) Halibut-Trail (b) Halibut-Shore

Figure 5-20: DSLAM trajectories calculated from the Halibut-Trail and Halibut-Shore datasets using increasing numbers of tracked features.

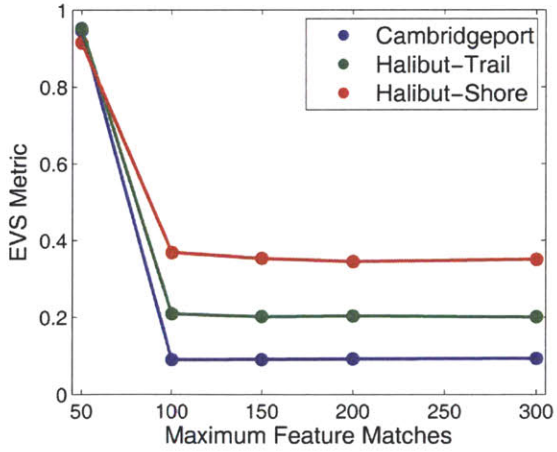
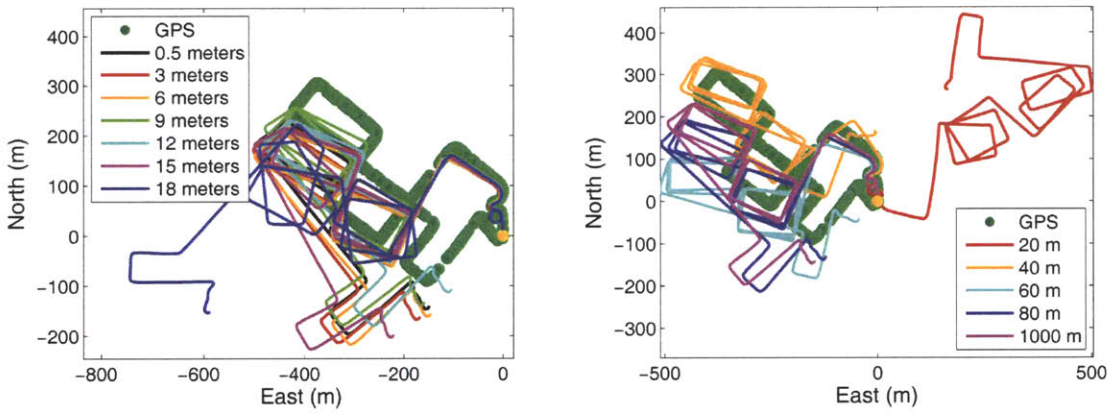



Figure 5-21: Effect on DSLAM navigation performance of limiting the maximum number of feature-landmark matches per frame in DSLAM.

### 5.7.3 Landmark Distance

The depth error associated with localizing landmarks when using a stereo camera is directly related to the actual distance to the point, meaning pose estimates are most accurate when calculated using features and landmarks close to the camera. There are two ways to test the sensitivity of the EVS metric to the distances of landmarks being used to calculate pose: either eliminate all landmarks in the map closer than some minimum depth threshold, or eliminate those further than some maximum threshold.

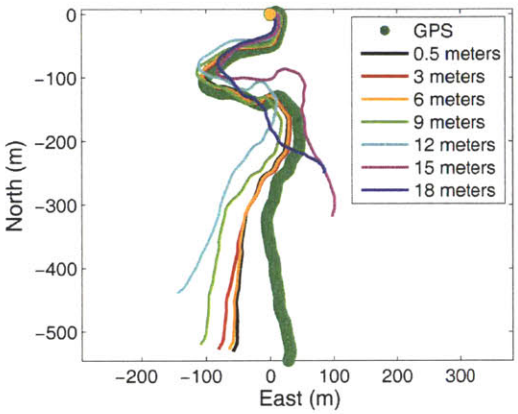



(a) Minimum Depth

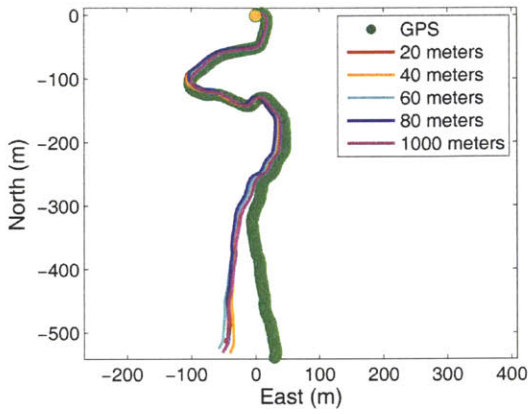
(b) Maximum Depth

Figure 5-22: DSLAM trajectories calculated from the Cambridgeport dataset using various minimum and maximum landmark depth thresholds.

Figures 5-22a, 5-23a, and 5-24a show multiple DSLAM trajectories calculated with various minimum landmark depth thresholds for the three outdoor datasets, and Figure 5-25a shows the EVS metrics for these trajectories. These results suggest that the minimum landmark depth has an effect on the system, but it does not tend to manifest itself until around 9-12 meters. This is expected, as there are very few objects within this range to the camera at any point in these datasets, as shown in Figure 5-26, which gives the distribution of landmark depths for each dataset, as calculated using the parameters given in Table 5.1. The rapid increase in EVS metric shown by Figure 5-25a indicates that features in the 10-20 meter range are particularly important when calculating the pose estimate, which is due to both the large number

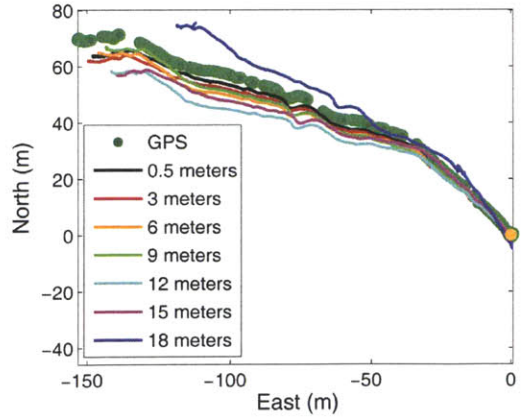



(a) Minimum Depth

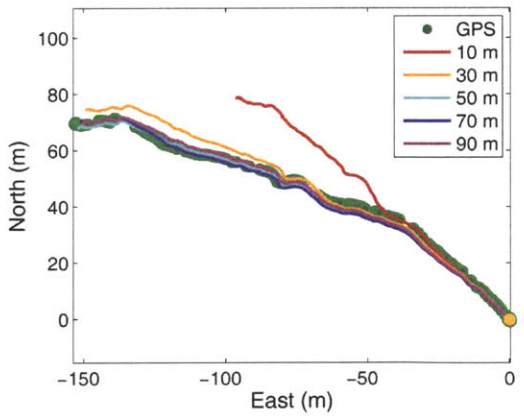



(b) Maximum Depth

Figure 5-23: DSLAM trajectories calculated from the Halibut-Trail dataset using various minimum and maximum landmark depth thresholds.

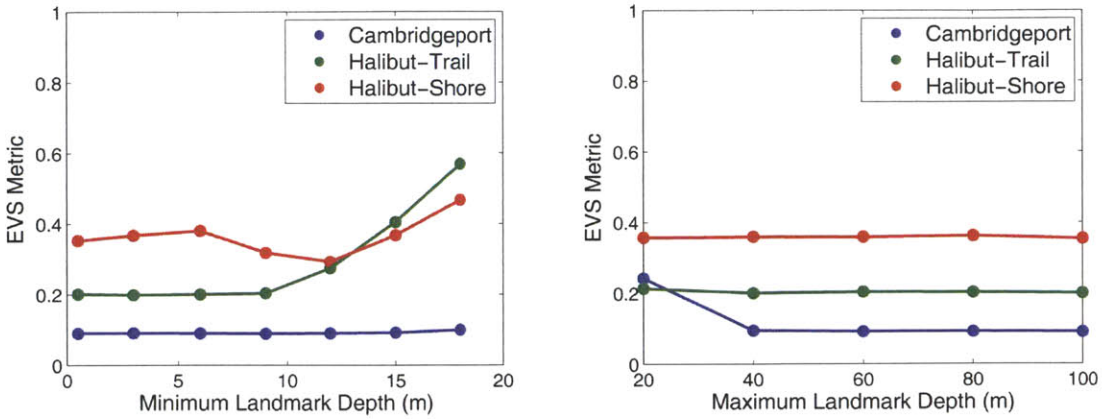



(a) Minimum Depth




(b) Maximum Depth

Figure 5-24: DSLAM trajectories calculated from the Halibut-Shore dataset using various minimum and maximum landmark depth thresholds.




(a) Minimum Depth

(b) Maximum Depth

Figure 5-25: Effect on DSLAM navigation performance of limiting the (a) minimum and (b) maximum depths of landmarks added to DSLAM's internal map.

of landmarks in this range and their relatively small depth uncertainties.

Figures 5-22b, 5-23b, and 5-24b show multiple DSLAM trajectories calculated with various maximum landmark depth thresholds, and Figure 5-25b shows the EVS metrics for these trajectories. From these figures and the EVS metric, the effect of limiting the range of the landmarks does not appear to be significant above 20-40 meters. The effect of using these distant landmarks is more one of increasing the consistency and error checking within RANSAC than one of actually shifting the trajectory. These distant landmarks should be included when available, but are not necessary for consistent performance. Additionally, as Figure 5-26 shows, there are relatively few landmarks in the map with depths greater than 40 or 60 meters, so the effect of limiting maximum depth is less significant than that of limiting minimum range because fewer landmarks are actually affected.

These results are significant when considered in the context of planetary hoppers. As was discussed in Chapter 2, a planetary hopper navigation system uses a downward-facing stereo vision system, primarily for navigation during the ascent and descent phases of the hover hop. During such maneuvers, the features in the environment being tracked (and thus also the landmarks they are being matched to) grow increasingly distant during an ascent, and increasingly proximal during descents. The stereo vision

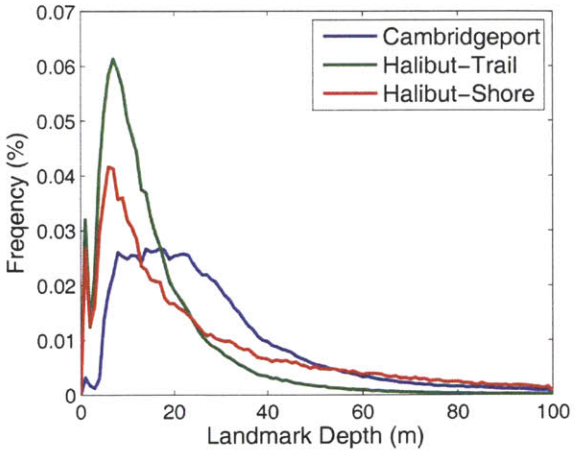


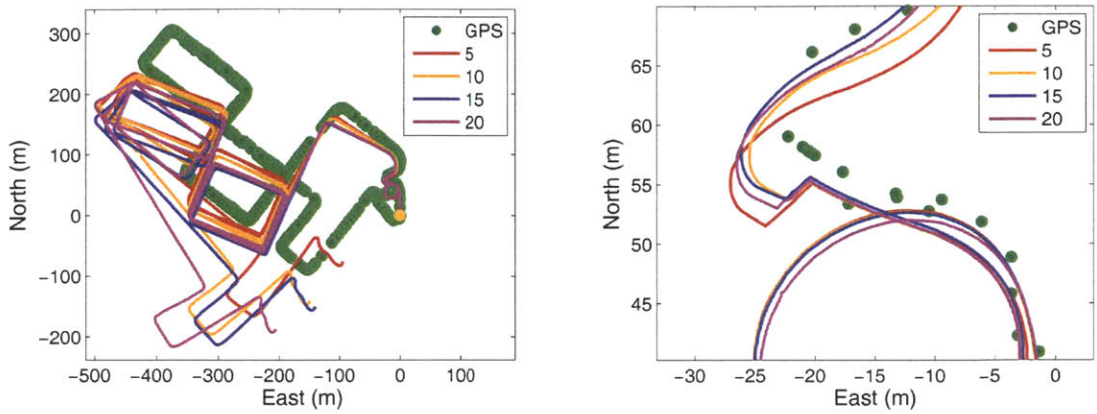

Figure 5-26: Distribution of the depths of all landmarks added to the internal map for each of the three outdoor datasets.

system will not be able to achieve the same level of accuracy at high altitudes as at low altitudes. However, navigational accuracy is most critical when the vehicle is near to the surface, as the hopper is much more robust to navigational inaccuracies when it is further from the surface, and thus further from hazards. Additionally, the most critical measurement during the ascent and descent phases of the hover hop is altitude, which can easily be constrained with the addition of a laser altimeter. Small deviations in translational motion are acceptable, as long as altitude and heading angle (measured by the forward-facing camera) can be constrained.

### 5.7.4 Window Length

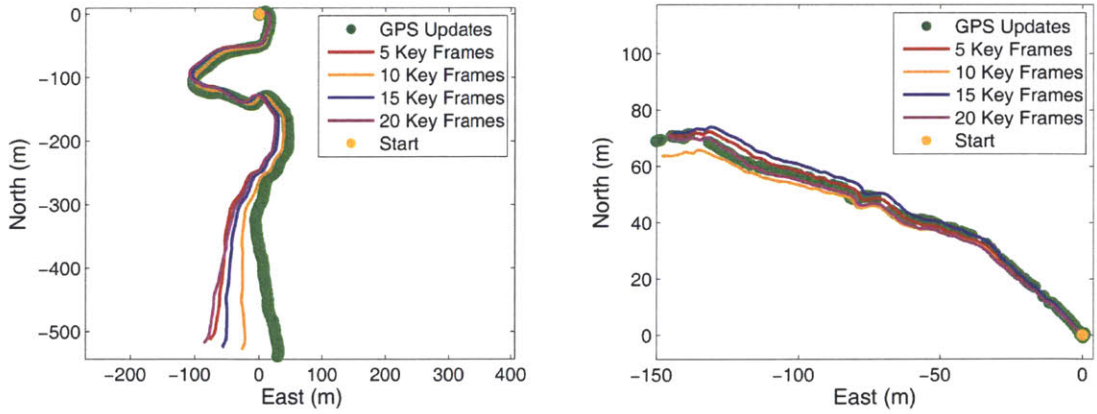
Figures 5-27 and 5-28 show DSLAM trajectories calculated from the Cambridgeport, Halibut-Trail, and Halibut-Shore datasets using window lengths of 5, 10, 15, and 20 key frames. Each of these trajectories is roughly within the expected noise due to RANSAC. Figure 5-29 shows the EVS metrics for each of these trajectories, which are approximately unchanged throughout the range of window lengths tested for each dataset.

Increasing the window length essentially equates to increasing the number of land-




(a) Cambridgeport (b) Cambridgeport - Close-up

Figure 5-27: DSLAM trajectories calculated from the Cambridgeport dataset using various window lengths. The close-up at right shows the effect of a large truck driving past the camera near the start of the dataset, blocking the majority of its field of view.




(a) Halibut-Trail (b) Halibut-Shore

Figure 5-28: DSLAM trajectories calculated from the Halibut-Trail and Halibut-Shore datasets using various window lengths.

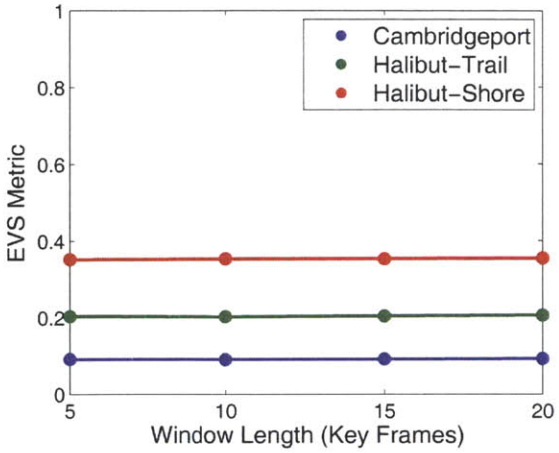



Figure 5-29: Effect of window length on DSLAM navigation performance.

marks stored in the map. A longer window is only beneficial when the oldest landmarks in the map are still being matched to features in the current frame, as the oldest landmarks should be the most accurate (ignoring all other factors affecting landmark accuracy). However, walking and driving trajectories generally do not repeat themselves, and rarely return to a location within the window length (typically 15 seconds or less). This means that for these types of trajectories, changes in window length should not have a significant effect on navigational accuracy, as long as the window is within a reasonable range (typically 5 key frames or greater).

One particular point of interest in the Cambridgeport trajectory is shown in Figure 5-27b, when a large truck drove in front of the camera. RANSAC is typically very effective at filtering out these types of disturbances, but the truck took up nearly the entire field of view of the camera, causing DSLAM to register it as camera motion (the truck was driving northeast). Here, the 5 key frame window has noticeably larger error than the 10-20 frame windows, as the longer window lengths were beneficial for relocating the camera to the original landmarks after the incorrect motion. The window with only 5 key frames had already removed the previous landmarks from its map at this point.

This insensitivity to window length might not translate as directly to hoppers as the other parameters discussed in this section. The hover hop trajectory and planetary

environments result in tracked features remaining in the camera field of view for long durations. Both the vertical maneuvers and the lateral traverse have long, straight segments along the axis of the camera. Hoppers do not turn corners, and only a small amount of drift should occur during vertical motions and hovers, meaning features will remain in the field of view of the camera for very long durations. This means that a hopper will benefit from common landmarks being held in memory for the duration of those maneuvers. However, increasing the window length significantly increases memory requirements and computational load. For this reason, future work will include the development of an intelligent algorithm to determine which landmarks are no longer in use and remove them from the map, rather than using a fixed window size.

### 5.7.5 Parameter Sensitivity

Multifactor Analysis of Variance (ANOVA) testing was used to provide an indication of which parameters have an effect on the EVS metric, and the likelihood that this effect is more significant than that of the noise caused by RANSAC nondeterminance. The three parameters tested and the range of values used are shown in Table 5.3. Maximum landmark depth was omitted from this analysis. A linear ANOVA model was selected in order to focus on the main effects of the factors, as opposed to their interactions, for two reasons. First, the interactions of combinations of two parameters are not intuitive in this system, many of them only potentially affecting system performance in rarely observed fringe cases, which were not being directly tested and are not applicable to hopping. Second, the linear model leaves four DOFs for error estimation, which was known to be of significance.

| Factor | 'Low' | 'High' | Description |
|---|---|---|---|
| Features | 100 | 150 | Maximum number of features tracked. |
| Depth | 0.5 | 10 | Minimum depth of initialized landmarks. |
| Window | 10 | 15 | Length of time to keep landmarks in the map. |

Table 5.3: Factors used for analysis of variance testing.

| Parameter | | Cambridgeport | | Halibut-Trail | | Halibut-Shore | |
|---|---|---|---|---|---|---|---|
| Factor | DOF | Mean Sq. | P-Value | Mean Sq. | P-Value | Mean Sq. | P-Value |
| Features | 1 | 0.00014 | 0.086 | 0.00483 | 0.358 | 0.00116 | 0.254 |
| Depth | 1 | 0.00917 | 0.067 | 0.04147 | 0.038 | 0.00008 | 0.749 |
| Window | 1 | 0.00001 | 0.576 | 0.00219 | 0.524 | 0.00036 | 0.500 |
| Error | 4 | 0.00003 | | 0.00449 | | 0.00065 | |

Table 5.4: DSLAM analysis of variance results for maximum number of features tracked, minimum landmark depth, and window length. P-values in red indicate a greater than 90% likelihood of significance, and those in blue indicate 75% likelihood of significance.

| Parameter | | Cambridgeport | | Halibut-Trail | | Halibut-Shore | |
|---|---|---|---|---|---|---|---|
| Factor | DOF | Mean Sq. | P-Value | Mean Sq. | P-Value | Mean Sq. | P-Value |
| Features | 1 | 0.12128 | 0.013 | 0.02906 | 0.293 | 0.00121 | 0.729 |
| Depth | 1 | 0.01831 | 0.175 | 0.07093 | 0.132 | 0.00663 | 0.433 |
| Window | 1 | 0.00072 | 0.760 | 0.03907 | 0.233 | 0.00001 | 0.978 |
| Error | 4 | 0.00676 | | 0.01986 | | 0.00876 | |

Table 5.5: Unified filter analysis of variance results for maximum number of features tracked, minimum landmark depth, and window length. P-values in red indicate a greater than 90% likelihood of significance, and those in blue indicate 75% likelihood of significance.

The results of the ANOVA analysis are presented in Table 5.4 for the DSLAM EVS and Table 5.5 for the unified system EVS for each of the three datasets introduced in Section 5.3. This analysis calculates the mean sum of squares contribution per DOF (labeled "mean sq.") for each factor, as well as for the error. In this case, "error" (not to be confused with the EVS error metric), represents randomness induced by RANSAC, as well as any interactions between the various parameters. When the mean sq. for a factor exceeds the threshold of the mean sq. for the error, it indicates that the factor causes a greater change in the variance of the measurement (i.e., the EVS metric) than that likely to be caused by random noise. The p-value gives the probability that a particular mean sq. value would be calculated as a result of randomness in the system, meaning a lower value indicates a greater likelihood of a factor having an effect on the trajectory EVS metric.

A p-value of 0.10 or less, indicating 90% or greater likelihood that the effect of the factor is not the result of random noise, was selected to identify factors of significance

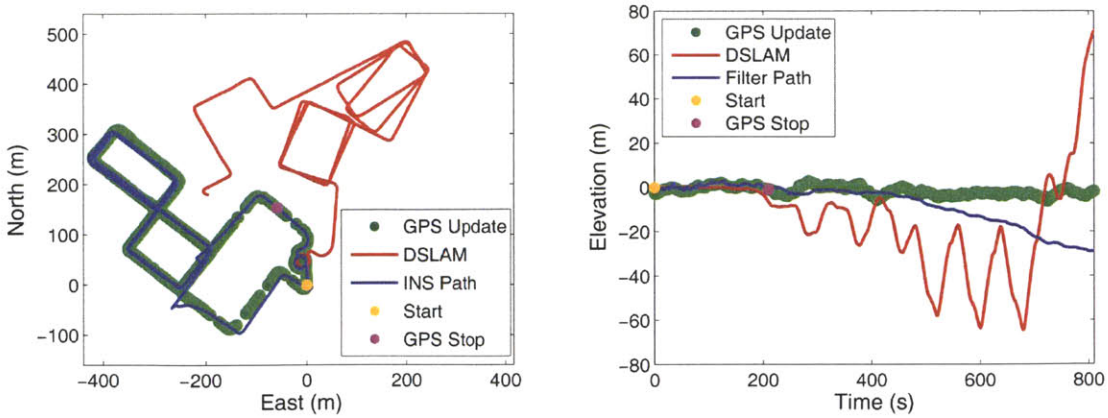
to system error. A p-value of 0.25 or less, indicating 75% or greater likelihood of a nonrandom effect, was selected to indicate likely effects. Across both DSLAM and the unified filter, changing the window length by five key frames does not appear to cause a distinguishable effect on the EVS metric. However, changing the minimum landmark depth or the number of features tracked does cause a distinguishable effect in several cases. These results support the findings of the preceeding sections discussing each parameter independently.

### 5.7.6 Camera Properties

In addition to the adjustable software parameters available in both DSLAM and the unified filter, adjustments can also be made to the camera. These include operational parameters, such as frame rate, resolution, shutter speed, and exposure time, as well as hardware parameters specific to the camera, such as field of view. Currently frame rate and resolution are manually fixed at runtime, while the camera automatically adjusts its shutter speed and exposure time during operation to its environment.

The most interesting of these parameters, however, is the camera field of view. A larger field of view means features can be located in a larger area, which should help in situations with more dynamic motion, such as cornering, because features can be tracked for longer durations. However, the larger field of view also means that a single pixel references a larger area in the map, reducing precision for feature matching. An additional drawback of a wide angle camera is that solar glare is more likely to occur on the lens, which makes the camera unusable.

Figure 5-30 shows the Cambridgeport loop repeated nearly identically using a wide field of view (66-degree) stereo camera. This trajectory's DSLAM EVS metric is 1.00, compared to 0.24 for the standard camera. However, when processed using the filter, many of the erroneous frames are rejected (10% of the total frames output by DSLAM), and the filtered trajectory EVS metric is only 0.16, which is identical to that of the standard camera (which is also 0.16). This result shows the robustness




(a) Trajectory (b) Elevation

Figure 5-30: Cambridgeport loop repeated using a wide field of view (66-degree) stereo camera. When considered independently, this camera has higher error than the narrow field of view camera shown in Figure 5-13, but exhibits similar performance when filtered using the unified system.

of the unified inertial and vision system filter. This wider field of view might be especially useful for the downward-facing vision system onboard a planetary hopper, which would allow a larger region to be scanned for both features for terrain-relative navigation and hazard detection. As long as it remains pointed downward, the system should be unaffected by solar glare on the lens, the largest additional source of error when switching to the wider camera field of view.

## 5.8 Hopping Motions

The indoor crane at Draper Laboratory was used to simulate typical hopper maneuvers, including an ascent and descent, roll rotation, and a scaled down hover hop. Unfortunately, the impulsive motions of the crane combined with the mechanical noise from its large motor meant that the IMU was too noisy to effectively use the filter. These datasets were post-processed with DSLAM using a longer window length (20 key frames) and more tracked features (200) than the outdoor walking and driving tests in order to better handle the harsh disturbances unique to the crane dynamics. DSLAM was able to recover most of the motions of the crane, but frequently

struggled with the large impulses of the crane, such as occur when starting or stopping translational motion. Because GPS was unavailable indoors and no alternative form of detailed truth measurements were available, these results are presented as is for purposes of demonstration and discussion, although no error metrics can be calculated. Future plans include testing the system on a hopper or hopper-like test vehicle, and such a vehicle will have smoother dynamics and truth data available for comparison.

The camera was attached to the crane facing downward for these datasets, rather than forward as in the previous datasets, to more realistically emulate the downward-facing stereo vision system onboard a hopper. Additionally, previous results presented in this chapter were all presented with the "world" coordinate system in north-east-down (NED) coordinates. The results of this section are presented in terms of altitude, cross-range, and down-range, due to the lack of GPS reference. The down-range axis of the world frame is defined by the direction of translational motion of the hop, when available, or as the initial heading of the system (in the case of the pure ascent and descent tests).

### 5.8.1 Roll Rotation

A roll rotation in a hopper coordinate system corresponds to a change in heading of the vehicle. This capability was tested by hanging the camera in the center of the room on the crane, rotating it about its vertical axis first in one direction and then in the other, and returning it to approximately its initial heading. Markings were added to the floor within the field of view of the camera to allow approximations of roll angle to be made by hand for comparison.

Figure 5-31 shows the DSLAM-calculated roll angle, as well as the manually measured comparison values. As the figure shows, DSLAM accurately estimates roll angle for the majority of the rotation, but struggles during one segment. When the camera reached the apex of its roll, some of the main features being tracked were lost from

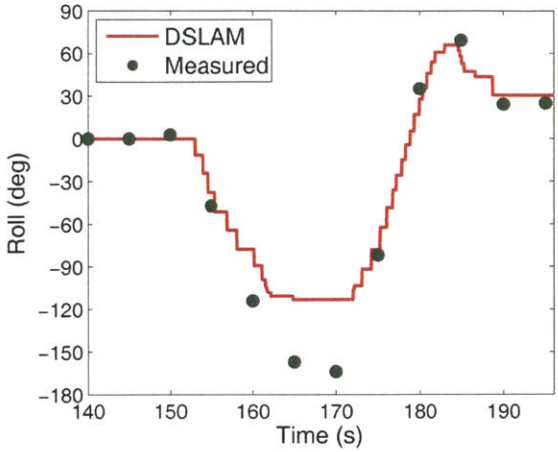



Figure 5-31: DSLAM-measured roll maneuver.

the field of view, and this may have caused the significant error in this region. When the roll direction reversed, these features returned to the field of view while the landmarks were still within the window, allowing them to be reacquired. An actual operational hopper will operate at larger altitudes, reducing the likelihood of these types of errors.

### 5.8.2 Ascent & Descent

The ascent and descent test was performed by raising the crane to its maximum altitude, holding it stationary for 10 seconds, and then returning it to its approximate initial location (similar roll angle and 5 cm lower than its initial position). DSLAM successfully measured position during this test, as shown by Figure 5-32. The slight disturbances match the motion profile of the raw video feed, indicating high likelihood that DSLAM is at least semi-accurately navigating the disturbances. However, these types of motion would typically be measured primarily using the IMU rather than the camera. The altitude vs. downrange plot, shown in Figure 5-35a, shows that the system experienced very little drift during the course of the maneuver, and Figure 5-35b shows that the maximum detected lateral motion was less than 15 cm, which is within the possible range of motion of the crane when it is started or stopped.

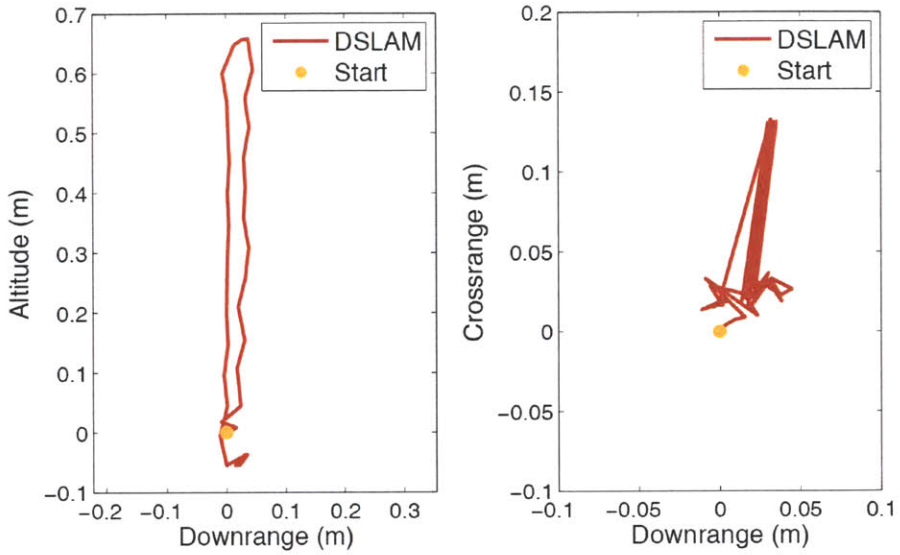
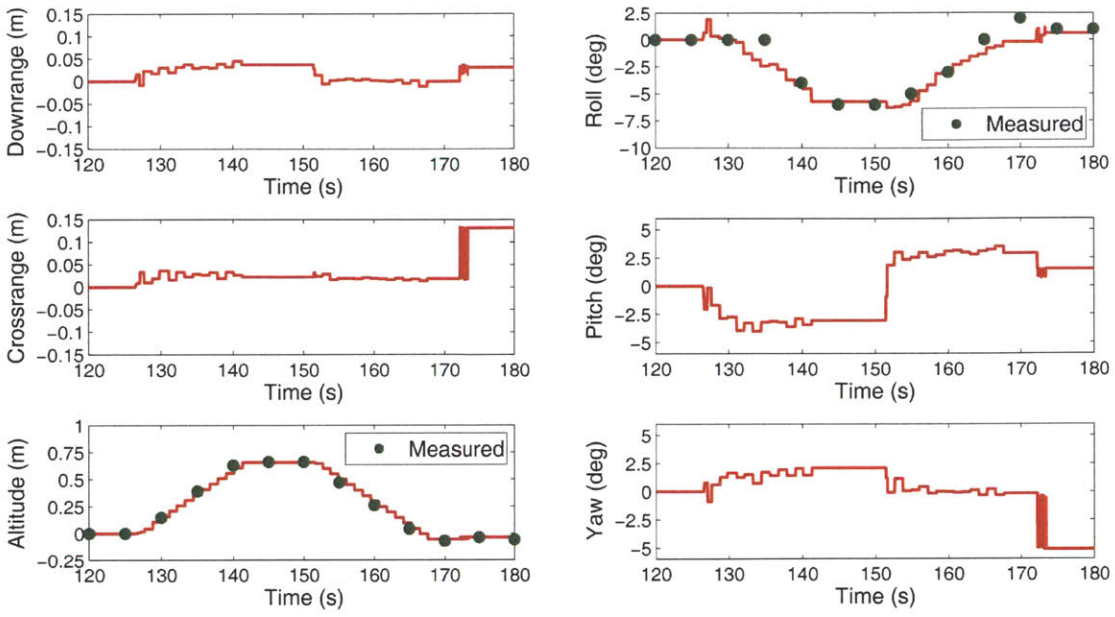



Figure 5-32: DSLAM-measured ascent and descent.

Figure 5-33 shows the position and attitude measurements from DSLAM during the ascent and descent maneuver. Coarse altitude and roll angle measurements were able to be calculated manually based off the orientation and size of markings placed on the floor before the test. These manual measurements are shown in Figure 5-33 in green, and show very good agreement with the DSLAM-calculated results.

The disturbances in this dataset are relatively small, but do appear to induce some drift. While the sensor head did experience roughly the roll profile shown in Figure 5-35b, the yaw and pitch angles should have returned to zero degrees at the apex and conclusion of the maneuver, when the crane was held stationary for 10 seconds. As soon as the descent segment begins, 150 seconds into the trajectory, both roll and yaw experience a near-discontinuity due to the impulsive jolt of the crane. However, the system is robust enough to regain its landmarks after this jolt and continue outputting a state estimate.




(a) Position

(b) Attitude

Figure 5-33: DSLAM-measured position and attitude of sensor head during a vertical ascent, hold, and descent maneuver.

### 5.8.3 Hover Hop

Figures 5-34 and 5-35 show the position results of a short-range hover hop trajectory, as measured by DSLAM from the Crane-Hop dataset. The altitude change of roughly 1 meter is consistent with expectations. However, the actual length of the trajectory was approximately 9 to 9.5 meters, rather than the 8.5 meters calculated by DSLAM.

This hover hop was a particularly difficult maneuver for the vision system to navigate, as a number of challenges were added to those already present in the Crane-Ascent-Descent dataset. One of these additional challenges is that especially large disturbances occur when lateral motion of the crane starts or stops, which cause the heavy cabling and hook to sway far enough that the camera loses nearly all of its original tracked landmarks. A second challenge occurred when the hop trajectory passed over a row of cabinets in the center of the room with close enough proximity that all features were lost for roughly a meter of travel. Once the camera exited this region, every

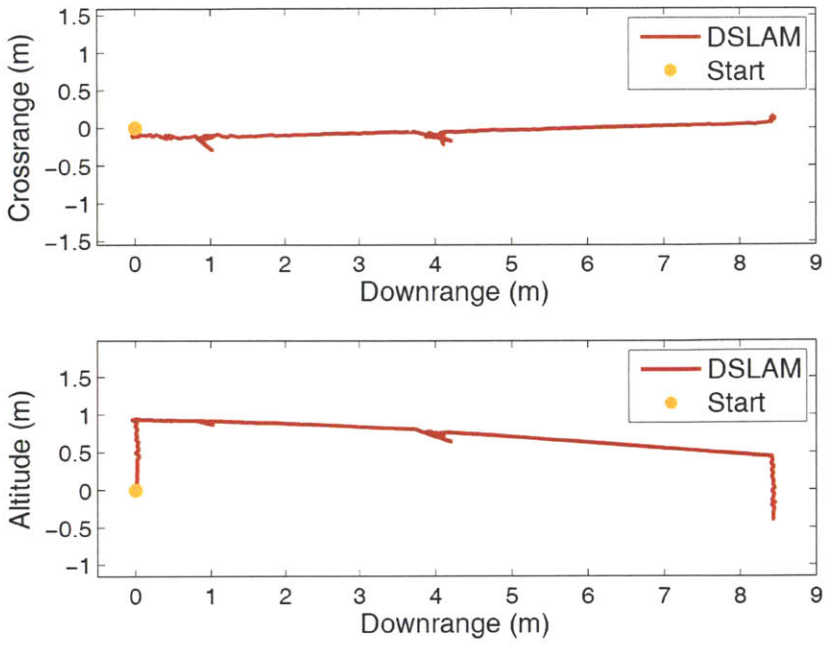
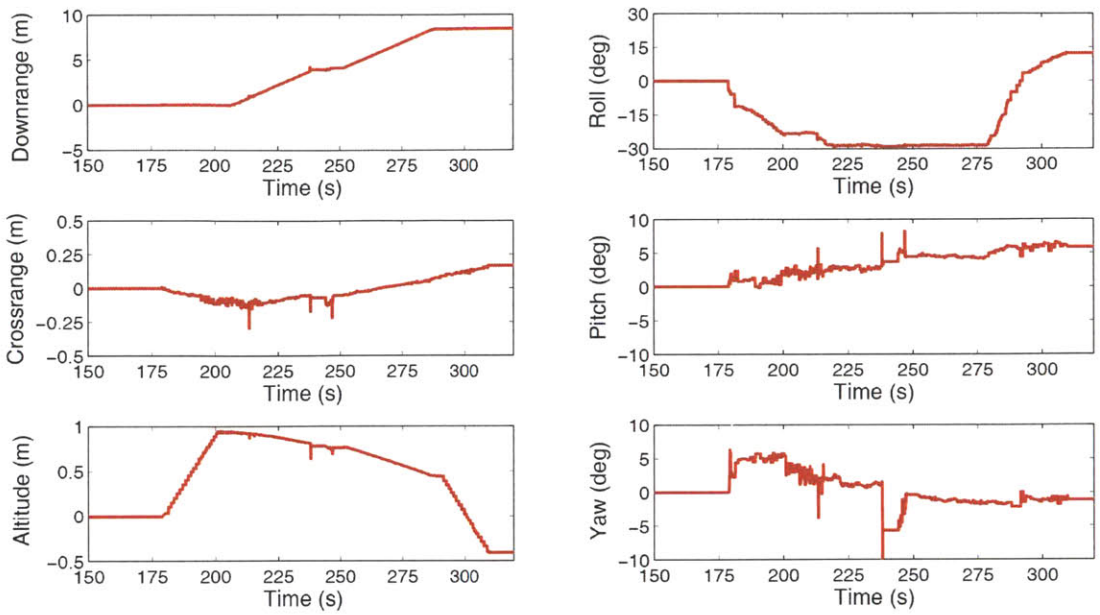



Figure 5-34: DSLAM-measured hover hop trajectory.

previously tracked landmark was out of the field of view of the camera, essentially requiring the map to be rebuilt from scratch. This occurs at the discontinuity visible in Figure 5-34 near 4 meters down-range from the starting location.

Figure 5-36 shows the measured trajectory (the location of the camera at each key frame is shown as a small blue coordinate frame, which together form the trajectory) and the locations of landmarks in the map. The landmarks that make up the cabinets at the midpoint of the trajectory can be seen near the midpoint of the trajectory, giving a sense of scale, as well as a gap in the blue coordinate frames, which indicate that no key frames were created during that brief period. The gap occurs right above the the highest landmarks (the top of the cabinets) and the trajectory is offset at the end of the gap, with a roughly 5 degree pitch error, and offset in down-range position by approximately a meter. As was discussed in Chapter 2, the system was predicted to struggle with these situations, and this is why a secondary, forward-facing vision system is additionally included in the proposed hopper navigation system architecture.




(a) Position

(b) Attitude

Figure 5-35: DSLAM-measured position and attitude of sensor head during a hover hop.

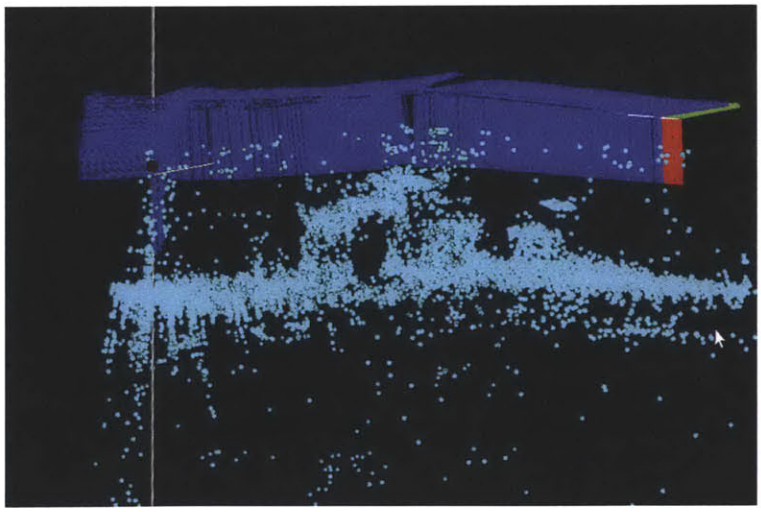


Figure 5-36: Screenshot of the DSLAM graphical interface showing locations of detected landmarks during the hover hop trajectory. The system passed over a number of large objects near the middle of the lateral traverse maneuver, causing errors in pitch angle and downrange position.

# Chapter 6

# Conclusions

This thesis describes the architecture, design, and preliminary testing of a hopper navigation system. Various capabilities of this system have been demonstrated in successful Earth-based tests, which confirmed the viability of this system for continued development for application to planetary hopping vehicles.

The goals of this project, as stated in Section 1.3, were to:

1. Develop a system architecture for a navigation system capable of estimating the state of a hopping vehicle.
2. Develop a prototype system to evaluate the proposed hopper navigation system architecture and its capabilities.
3. Demonstrate that unifying inertial and relative navigation techniques provides advantages applicable to hoppers.
4. Evaluate the performance of the prototype system and demonstrate hopper navigation capabilities.

All of these goals have been met, and the following sections present the conclusions of this work with relation to each of these goals, as well as a summary of lessons learned. Section 6.5 comments on this system's applicability to alternative systems beyond planetary hoppers, and Section 6.6 presents suggestions for further development.

## 6.1 Hopper Navigation System Architecture

A system architecture for a planetary hopper navigation system, consisting of a downward-facing stereo vision system, forward-facing monocular vision system, an IMU, and scanning laser rangefinders, was proposed in Section 2.5. This architecture overcomes the challenges of hopper navigation presented in Section 2.2, and achieves the goals established for hopper navigation in Section 2.3.

### 6.1.1 Challenges of Hopper Navigation

The first challenge of hopper navigation is the strict mass and power limitations that apply to all components onboard a hopper. This was accommodated by utilizing passive vision-based sensors in place of the more traditional, active Doppler radar sensors used for planetary landing. Using a vision-based system also allows the use of a lower-grade IMU than would traditionally be used onboard a landing spacecraft. The prototype Stingray system demonstrated that these lower mass, power, and cost sensors are capable of estimating a navigation state, although the accuracy of this system in comparison to the more traditional systems could not be directly compared. The prototype system has a total mass of about 1.5 kg and consumes approximately 40 W of power.

The second challenge is hazard detection and avoidance. These specific capabilities have not yet been demonstrated in hardware, although there are plans to do so in the future. The system architecture, however, is capable of collecting the sensor data required to run such systems. The flexible software framework of the prototype system was designed to allow seamless integration of such algorithms as soon as they become available for testing.

The third challenge is that of terrain variability in the operational environment. While the IMU is effectively immune to terrain conditions, the performance of vision systems is highly dependent on the terrain viewed by their cameras. In order to mitigate the

associated risks, the vision system was designed to use purely generic features from images, rather than requiring any particular objects to be within view. This capability was demonstrated by the results presented in Chapter 5, which showed relatively consistent performance across various classes of terrain.

The fourth challenge is that a hopper navigation system must be capable of estimating the vehicle state even during rapid motions. This challenge is especially apparent in the vision systems when the cameras blur due to rapid movements, but is mitigated by the inclusion of the IMU. The results of Chapter 5 show many rapid motions being accurately navigated by the unified navigation system, showing that this challenge has been met.

### 6.1.2 Goals of Hopper Navigation

The goals described in Section 2.3 can be broken down into three categories: vertical maneuvers, lateral maneuvers, and hazard avoidance. Navigation capability was demonstrated for these goals, with the exception of hazard avoidance, which was not evaluated by this thesis.

The primary concerns during vertical motion were determined to be measurement of altitude and heading. The test results presented in Section 5.8.2 and Figure 5-33 show that the prototype system is capable of accurately making these measurements to within 0.10 m and 2.5 degrees, respectively, using vision alone. These measurements are expected to be more accurate with the further inclusion of the unified system, which was shown to provide improved performance.

The primary concerns during lateral motion were determined to be measurement of attitude and heading, as well as vehicle odometry. While attitude measurement accuracy could not be evaluated directly, inspection of the calculated state estimations showed good agreement with the expected results. Odometry measurements taken with the prototype system and shown in Chapter 5 showed very good agreement with GPS truth data in a variety of operational settings.

### 6.1.3 Architecture Evaluation

The system's numerous advantages make it an ideal navigation sensor system for planetary exploration by hopping. First, and most importantly, the selected architecture is capable of performing all of the required top-level functions of an effective hopper navigation system, as listed in Table 2.1. The system combines multiple specialized systems into a single generic system, and at least one of these systems is capable of measuring any predicted motion of a hopper at high accuracy, while the others are always running to provide supplemental data, providing smooth transitions and redundancy. Furthermore, this architecture is designed to take advantage of attributes unique to planetary hoppers, such as the hover hop trajectory and its decoupled axes of motion.

The largest disadvantage of the selected architecture is its heavy reliance on vision systems, which have only limited flight heritage. Robust vision algorithms are still in the developmental stage and require flight demonstration. Additionally, vision systems are highly dependent on their external environments, which puts limitations on mission planning. Fortunately, vehicle hazards are generally directly correlated to detected features, so these vision systems will actually gain accuracy as their surrounding terrain grows more treacherous.

## 6.2 Stingray Testbed

As described in detail in Chapter 3, the Stingray hardware and software testbed was developed to evaluate the components of the selected architecture and to serve as a prototype hopper navigation system. The Stingray system is highly modular, allowing flexibility in future development and testing. The use of commercially available sensors and hardware reduced the cost of development, and the experimental results of Chapters 4 and 5 showed the system was operational and could meet the project goals, even with low-grade sensors.

An additional objective of developing the Stingray system was to determine whether a navigation system can be encapsulated independently of a vehicle, allowing development and testing of the navigation system to occur entirely independently of the flight vehicle. Verifying such an objective must be done in two stages: testing the system independently and testing the system once integrated with a flight vehicle. The first stage of this was demonstrated by the success of testing the system, as presented in Chapter 5. However, a flight vehicle was not yet available for testing, and thus demonstration of the more critical functionality of the integrated system must be saved for future work.

## 6.3 Unified Inertial and Relative Navigation

The Stingray navigation system testbed was used to develop and test the unified inertial and stereo vision-based navigation algorithm, which was described in detail in Chapter 4. The modularity of the Stingray system allowed multiple algorithms to be compared side by side using the same data, which showed that the unified system provides improvements to navigation accuracy over both the inertial-only and vision-only systems. Additionally, the unified system is more robust to sensor dropouts and other sources of error than either system operating independently. The unified system was shown to provide improvements to overall navigation performance by improving IMU bias estimation, mitigating the effects of IMU drift, more accurately navigating periods of rapid rotation, and being more robust to sensor dropouts. For these reasons, the unified inertial and vision-based navigation system is recommended for planetary hopping applications over either the inertial-only or vision-only systems operating independently.

## 6.4 Prototype System Performance Evaluation

The experimental testing campaign of the prototype system (conducted using the Stingray testbed) had two primary goals: to demonstrate the capability of the system and to determine its major sources of error. These capability demonstrations focused on estimation of a 6 degree-of-freedom (DOF) navigation state, measurement of velocities and rotation rates, navigation in an unstructured environment, and navigation of a hover hop trajectory. The success of these demonstrations, shown in Chapter 5, established the system to be at NASA Technology Readiness Level (TRL) 4[1], and supports additional development of the system.

### 6.4.1 System Performance

As the results presented in Chapter 5 show, the testbed system produces promising results, especially given its early stage of development. Even when using the low-grade sensors intended for preliminary development of the project, the system does a very good job of navigating the demonstration trajectories in a variety of environments. The effects of many of the observed current performance issues could be mitigated by upgrading the IMU to a model more likely to fly onboard a planetary exploration mission.

Section 5.5 presented results of tests conducted walking in outdoor environments. In these environments, the system was shown to work well as long as features were kept within range of the camera. Walking on the uneven terrain caused larger disturbances than those predicted for a hopper, but the system was able to navigate the majority of these situations with reasonable accuracy (less than 5% final positioning error with respect to distance traveled), especially when additionally using data from the IMU. This indicates that the navigation system is capable of handling moderate

[1]The Technology Readiness Level (TRL) is a metric used by NASA to evaluate the maturity of a technology. TRL 4 is defined as "Component and/or breadboard validation in laboratory environment" [83].

disturbances comparable to those caused by thruster firings of a hopper for thrusting and attitude corrections (in the absence of vehicle vibration).

Section 5.6 presented results of tests conducted driving in an urban environment, where the system successfully navigated repeated corners and moderate velocities throughout the test run. The unified system did especially well in these tests, due to the smoother dynamics of the car as opposed to the pedestrian, achieving navigation accuracies of less than 1% error in final position with respect to total distance traveled. This indicates that, if vehicle dynamics of a hopper can be kept fairly steady, navigation of long-distance hops in moderately feature-rich environments is feasible.

Section 5.8 presented results of tests conducted using an indoor crane to simulate hopper motions with a downward-facing vision system, highlighting many strengths and weaknesses of the system. The testbed system was able to reconstruct a small scale hover hop trajectory, although the large disturbances induced by the crane caused the reconstructed trajectory to have significant error (approximately 10% final positioning error with respect to total distance traveled). Furthermore, the vibrational environment of the crane induced too much noise in the low-grade IMU to effectively use the unified system. The system requires additional development to increase its robustness to vehicle vibrations. The roll and ascent and descent tests further showed that the system is capable of measuring components of vehicle pose, although truth data was unavailable to evaluate exactly how accurately.

### 6.4.2 Sources of Error

As discussed in Section 5.4, only positional accuracy of the system can be evaluated directly due to the limitations of using GPS for attitude truth data. However, an error metric was developed to evaluate system performance relative to the limited information available from GPS, which provides significant insight into the sources of error in the system by indicating locations of difficulty in the trajectory. Segments of significant error tend to be the result of vision system dropouts, high rotation rates,

and a lack of features detected near to the camera. The worst errors typically involve simultaneous combinations of these scenarios.

Sustained vision system dropouts, the largest source of error, were most often the result of either glare on the camera from the sun or a large disturbance causing the camera to point too far upwards above the horizon, losing all features. However, because a hover hop does not involve turns, a carefully designed mission plan can easily avoid ever pointing the cameras too near to the sun. Additionally, only an extreme disturbance could cause both the downward- and forward-facing camera systems to completely lose track of the planetary surface, the most likely cause of which would be a vehicle crash.

## 6.5 Applications Beyond Hopping

Predicted planetary hopper trajectories involve a broad spectrum of velocities, rotation rates, altitudes, feature depths, and unknown terrain that must all be accounted for without use of GPS, as well as many other difficult situations and scenarios. These numerous challenges of navigating planetary hoppers, combined with the vehicle-independent hardware and software system developed for preliminary testing, have led to the development of what is in many ways a "generic" navigation system. This generic system could be applied to any vehicle requiring near-surface navigation in GPS-denied environments, provided its motions do not exceed the operational thresholds of the system and the environment contains sufficient features for the vision system to detect and track.

Chapter 5 showed that the system is already capable of navigating a pedestrian or vehicle in urban and natural environments. A derivative of the Stingray system would be well suited for navigation of first responders in emergency situations, such as inside unmapped buildings or areas lacking GPS coverage. Additional system testing may include navigation within buildings and onboard low-altitude flying vehicles, such as terrestrial rockets or unmanned aerial vehicles.

## 6.6 Future Work

While the primary goals of this project have been met, an operational hopper navigation system is far from fully developed. However, the success of the various testbed demonstrations confirms that the system is ready to move forward to the next stage of system testing, which will involve re-testing the system with more accurate truth data to determine the performance limitations of the system and integrating the system with a flight vehicle for open-loop flight tests, advancing the system to TRL 5[2]. Comprehensive closed-loop testing onboard a terrestrial rocket will further advance the system to TRL 6[3], enabling selection of the system by a flight program.

Additional development of the Stingray system will focus on working towards this goal of performing a closed-loop flight test onboard a terrestrial rocket. This will require a thorough investigation of the latencies for data throughput in the system. The forward-facing monocular vision system and a scanning laser rangefinder will be added to the system in order to fully implement the architecture proposed in Section 2.5. Once this is complete, a second testing campaign will determine the hard performance, operational, and environmental limits of the system, as well as system performance in environments analogous to those of planetary hoppers.

[2] "Component and/or breadboard validation in relevant environment" [83].

[3] "System/subsystem model or prototype demonstration in a relevant environment (ground or space)" [83].

# Bibliography


[1] Ephraim Robert Lanford. Unique abilities of hopper spacecraft to enable national objectives for solar system exploration. Master's thesis, Massachusetts Institute of Technology, 2011.

[2] P. M. Cunio, F. Alibay, P. Meira, T. Sheerin, E. Lanford, E. Krupczak, and J. A. Hoffman. Options in the solar system for planetary surface exploration via hopping. In *Proceedings of the 2011 IEEE Aerospace Conference*, Big Sky, Montana, March 5 - 12, 2011.

[3] R. Zubrin. Methods for achieving long range mobility on Mars. In *Proceedings of the AIAA Joint Propulsion Conference*, 1992.

[4] S. D. Howe, R. C. O'Brien, R. M. Ambrosi, B. Gross, J. Katalenich, L. Sailer, J. Webb, M. McKay, J. C. Bridges, and N. P. Bannister. The Mars hopper: An impulse-driven, long-range, long-lived mobile platform utilizing in situ Martian resources. *Journal of Aerospace Engineering*, 225(2):144–153, 2011.

[5] Evgeny Shafirovich, Michael Salomon, and Iskender Gokalp. Mars hopper versus Mars rover. *Acta Astronautica*, 59(8-11):710–716, 2006.

[6] G. A. Landis and D. L. Linne. Mars rocket vehicle using in situ propellants. *Journal of Spacecraft and Rockets*, 38(5):730–735, 2001.

[7] R. Zubrin. Mars Gashopper. Technical Report NAS3-00074, Pioneer Astronautics, 2000.

[8] G. Savu. *'Jumper' for Mars surface exploration*, pages 163–174. The Case for Mars V. Univelt, Inc., Boulder, Colorado, 2000.

[9] Babak E. Cohanim, Nicholas A. Harrison, Todd J. Mosher, Jennifer Heron, Kathryn Davis, Jeffrey A. Hoffman, Phillip M. Cunio, Javier de Luis, and Michael Joyce. Small lunar exploration and delivery system concept. In *AIAA Space 2009 Conference*, September 14-17, 2009.

[10] Phillip M. Cunio, Alessandra Babuscia, Zachary J. Baily, Hemant Chaurasia, Rahul Goel, Alessandro A. Golkar, Daniel Selva, Eric Timmons, Babak E. Cohanim, Jeffrey A. Hoffman, and David W. Miller. Initial development of an Earth-based prototype for a lunar hopper autonomous exploration system. In *AIAA Space 2009 Conference*, 2009.

[11] Phillip M. Cunio, Sarah L. Nothnagel, Ephraim Lanford, Ryan McLinko, Christopher J. Han, Claas T. Olthoff, and Babak E. Cohanim. Further development and flight testing of a prototype lunar and planetary surface exploration hopper: Update on the TALARIS Project. In *AIAA Space 2010 Conference*, Anaheim, CA, August 30 - September 2, 2010.

[12] Christopher Rossi, Phillip M. Cunio, Farah Alibay, Joe Morrow, Sarah L. Nothnagel, Ted Steiner, Christopher J. Han, Ephraim Lanford, and Jeffrey A. Hoffman. TALARIS Project update: Overview of flight testing and development of a prototype planetary surface exploration hopper. In *Proceedings of the 62nd International Astronautical Congress*, Cape Town, South Africa, 2011.

[13] Brett J. Streetman, Michael C. Johnson, and Jules F. Kroehl. A generic framework for spacecraft GN&C emulation: Performing a lunar-like hop on the Earth. In *AIAA Guidance, Navigation, and Control Conference*, Toronto, Canada, August 2-5, 2010.

[14] Paul J. Huxel and Babak E. Cohanim. TALARIS hopper testbed navigation analysis. In *Proceedings of the 2011 IEEE Aerospace Conference*, Big Sky, Montana, March 5 - 12, 2011.

[15] Akil J. Middleton. Modeling and vehicle performance analysis of Earth and lunar hoppers. Master's thesis, Massachusetts Institute of Technology, 2010.

[16] Claas T. Olthoff, Phillip M. Cunio, Jeffrey A. Hoffman, and Babak E. Cohanim. Incorporation of flexibility into the avionics subsystem for the TALARIS small advanced prototype vehicle. In *AIAA Space 2010 Conference*, Anaheim, California, 2010.

[17] Christopher J. Han. Development of modular real-time software for the TALARIS lunar hopper testbed. Master's thesis, Massachusetts Institute of Technology, 2011.

[18] Sarah L. Nothnagel, Zachary J. Baily, Phillip M. Cunio, Jeffrey A. Hoffman, Babak E. Cohanim, and Brett J. Streetman. Development of a cold gas spacecraft emulator system for the TALARIS hopper. In *AIAA Space 2010 Conference*, Anaheim, CA, August 30 - September 2, 2010.

[19] Sarah L. Nothnagel. Development of a cold gas propulsion system for the TALARIS hopper. Master's thesis, Massachusetts Institute of Technology, 2011.

[20] Joseph M. Morrow, Sarah L. Nothnagel, Phillip M. Cunio, and Jeffrey A. Hoffman. Verification and validation of a cold gas propulsion system for the TALARIS hopper testbed. In *AIAA Space 2011 Conference*, Pasadena, CA, 2011.

[21] Phillip M. Cunio, Takuto Ishimatsu, Jim Keller, Zahra Khan, Ryan Odegard, Peter Waswa, and Geoffrey A. Landis. Near-term Mars sample return using in-situ oxygen generation. In *AIAA Space 2007 Conference*, Long Beach, California, 2007.

[22] Ted Steiner, Benjamin Saunders, Ephraim Lanford, Phillip M. Cunio, and Jeffrey A. Hoffman. Hopping vehicles for rapid regional exploration of the surface of Titan. In *Proceedings of the 62nd International Astronautical Congress*, Cape Town, South Africa, October 3-7, 2011.

[23] Ted J. Steiner, Scott A. Rasmussen, Paul A. DeBitetto, Babak E. Cohanim, and Jeffrey A. Hoffman. Unifying inertial and relative solutions for planetary surface navigation. In *Proceedings of the 2012 IEEE Aerospace Conference*, Big Sky, Montana, March 3-10, 2012.

[24] Babak E. Cohanim, Simran Dhillon, and Jeffrey A. Hoffman. Statistical hazard detection using shadows for small robotic landers/hoppers. In *Proceedings of the 2012 IEEE Aerospace Conference*, Big Sky, Montana, March 3-10, 2012.

[25] Tye Brady, Timothy Crain, and Stephen Paschall. ALHAT system validation. In *Proceedings of the 8th International ESA Conference on Guidance, Navigation & Control Systems*, Karlovy Vary, Czech Republic, June 5-10, 2011.

[26] Stephen Paschall and Tye Brady. Demonstration of a safe & precise planetary landing system on-board a terrestrial rocket. In *Proceedings of the 2012 IEEE Aerospace Conference*, Big Sky, Montana, March 3-10, 2012.

[27] D. Adams, T. B. Criss, and U. J. Shankar. Passive optical terrain relative navigation using APLNav. In *Proceedings of the 2008 IEEE Aerospace Conference*, Big Sky, Montana, March 1-8, 2008.

[28] Andrew E. Johnson and James F. Montgomery. Overview of terrain relative navigation approaches for precise lunar landing. In *Proceedings of the 2008 IEEE Aerospace Conference*, Big Sky, Montana, March 1-8, 2008.

[29] A. I. Mourikis, N. Trawny, S. I. Roumeliotis, A. E. Johnson, A. Ansar, and L. Matthies. Vision-aided inertial navigation for spacecraft entry, descent, and landing. *IEEE Transactions on Robotics*, 25(2):264–280, 2009.

[30] Christian Boleat, Benoit Frapard, Frederic Deladerriere, Marc Souyri, and Salvatore Mancuso. Architecture of a new generation vision based navigation camera for planetary lander - the NPAL developments. In *DASIA 2005 - Datasystems in Aerospace*, pages 89–91, Edinburgh, Scotland, United kingdom, May 30 - June 2, 2005. European Space Agency.

[31] G. Flandin, B. Polle, J. Lheritier, and P. Vidal. Vision based navigation for autonomous space exploration. In *Proceedings of the 2010 NASA/ESA Conference on Adaptive Hardware and Systems*, 2010.

[32] Jeff Delaune, Guy Le Besnerais, Martial Sanfourche, Thomas Voirnin, Clement Bourdarias, and Jean-Loup Farges. Optical terrain navigation for pinpoint landing: Image scale and position-guided landmark matching. In *Proceedings of the 35th Annual AAS Guidance and Control Conference*, Brekenridge, Colorado, 2012.

[33] Leena Singh and Sungyung Lim. On lunar on-orbit vision-based navigation: Terrain mapping, feature tracking driven EKF. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Honolulu, Hawaii, 2008.

[34] Li Shuang and Zhang Liu. Autonomous navigation and guidance scheme for precise and safe planetary landing. *Aircraft Engineering and Aerospace Technology*, 81(6):516, 2009.

[35] Martin Barczyk and Alan F. Lynch. Invariant extended Kalman filter design for a magnetometer-plus-GPS aided inertial navigation system. In *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 5389–5394, 2011.

[36] M. Barczyk, M. Jost, D. R. Kastelan, A. F. Lynch, and K. D. Listmann. An experimental validation of magnetometer integration into a GPS-aided helicopter UAV navigation system. In *American Control Conference (ACC)*, pages 4439–4444, 2010.

[37] E. N. Johnson, P. A. DeBitetto, C. A. Trott, and M. C. Bosse. The 1996 MIT/Boston University/Draper Laboratory autonomous helicopter system. In *Digital Avionics Systems Conference, 1996., 15th AIAA/IEEE*, pages 381–386, 1996.

[38] Peter Corke. An inertial and visual sensing system for a small autonomous helicopter. *Journal of Robotic Systems*, 21(2):43–51, 2004.

[39] R. Madison, G. Andrews, P. Debitetto, S. Rasmussen, and M. Bottkol. Vision-aided navigation for small UAVs in GPS-challenged environments. In *2007 AIAA InfoTech at Aerospace Conference*, volume 3, pages 2733–2745, Rohnert Park, CA, May 7 - 10, 2007. American Institute of Aeronautics and Astronautics.

[40] Allen D. Wu and Eric N. Johnson. Autonomous flight in GPS-denied environments using monocular vision and inertial sensors. In *AIAA Infotech at Aerospace*, Atlanta, Georgia, April 20-22, 2010.

[41] Sukchang Yun, Byoung-Jin Lee, Young Jae Lee, and Sangkyung Sung. Real-time performance test of an vision-based inertial SLAM. In *2010 International Conference on Control Automation and Systems (ICCAS)*, pages 2423–2426, 2010.

[42] M. George and S. Sukkarieh. Inertial navigation aided by monocular camera observations of unknown features. In *2007 IEEE International Conference on Robotics and Automation*, pages 3558–3564, 2007.

[43] Tianguang Zhang, Wei Li, Kolja Kuehnlenz, and Martin Buss. Multi-sensory motion estimation and control of an autonomous quadrotor. *Advanced Robotics*, 25(11-12):1493–1514, 2011.

[44] M. Achtelik, M. Achtelik, S. Weiss, and R. Siegwart. Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments.

In *IEEE International Conference on Robotics and Automation*, pages 3056–63, Piscataway, New Jersey, May 9-13, 2011.

[45] C. N. Taylor. Fusion of inertial, vision, and air pressure sensors for MAV navigation. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 475–480, 2008.

[46] D. Landis, T. Thorvaldsen, B. Fink, P. Sherman, and S. Holmes. A deep integration estimator for urban ground navigation. In *Proceedings of the 2006 IEEE/ION Position, Location, and Navigation Symposium*, pages 927–932, San Diego, California, April 25-27, 2006.

[47] J. P. Tardif, M. George, M. Laverne, A. Kelly, and A. Stentz. A new approach to vision-aided inertial navigation. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4161–4168, 2010.

[48] D. I. B. Randeniya, S. Sarkar, and M. Gunaratne. VisionIMU integration using a slow-frame-rate monocular vision system in an actual roadway setting. *IEEE Transactions on Intelligent Transportation Systems*, 11(2):256–266, 2010.

[49] S. B. Kim, J. C. Bazin, H. K. Lee, K. H. Choi, and S. Y. Park. Ground vehicle navigation in harsh urban conditions by integrating inertial navigation system, global positioning system, odometer and vision data. *Radar, Sonar & Navigation*, 5(8):814–823, 2011.

[50] D. M. Helmick, Yang Cheng, D. S. Clouse, L. H. Matthies, and S. I. Roumeliotis. Path following using visual odometry for a Mars rover in high-slip environments. In *Proceedings of the 2004 IEEE Aerospace Conference*, volume 2, pages 772–789, 2004.

[51] Thomas M. Howard, Arin Morfopoulos, Jack Morrison, Yoshiaki Kuwata, Carlos Villalpando, Larry H. Matthies, and Michael McHenry. Enabling continuous planetary rover navigation through FPGA stereo and visual odometry. In *Proceedings of the 2012 IEEE Aerospace Conference*, Big Sky, Montana, 2012.

[52] Bao Jinsong, Yu Dili, Hu Xiaofeng, Jin Ye, and Du Sanhu. The study of hybrid vision-based navigation for lunar rover in virtual environment. In *International Conference on Mechatronics and Automation (ICMA)*, pages 655–659, 2009.

[53] Fabio Cozman and Eric Krotkov. Automatic mountain detection and pose estimation for teleoperation of lunar rovers. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 2452–2457, Albuquerque, New Mexico, April 20-25, 1997.

[54] R. A. Jarvis. A perspective on range finding techniques for computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2):122–39, 1983.

[55] W. E. L. Grimson. Computational experiments with a feature based stereo algorithm [picture processing]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(1):17–34, 1985.

[56] Zhiwei Zhu, Taragay Oskiper, Supun Samarasekera, and Rakesh Kumar. Precise visual navigation using multi-stereo vision and landmark matching. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 6561, April 27, 2007.

[57] C. Drocourt, L. Delahoche, B. Marhic, and A. Clerentin. Simultaneous localization and map construction method using omnidirectional stereoscopic information. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 894–899, 2002.

[58] Christopher Mei, Gabe Sibley, Mark Cummins, Paul Newman, and Ian Reid. RSLAM: a system for large-scale mapping in constant-time using stereo. *International Journal of Computer Vision*, 94(2):198–214, 2011.

[59] N. Sawasaki, M. Nakao, Y. Yamamoto, and K. Okabayashi. Embedded vision system for mobile robot navigation. In *Proceedings 2006 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2693–2698, 2006.

[60] R. Kalarot and J. Morris. Comparison of FPGA and GPU implementations of real-time stereo vision. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 9–15, 2010.

[61] K. C. Kluge. SAMLOS: a 2D simultaneous localization and mapping algorithm based on lines of sight. In *Proceedings of the 2003 IEEE Intelligent Vehicles Symposium*, pages 438–443, 2003.

[62] Qiang Zhan, Shouren Huang, and Jia Wu. Automatic navigation for a mobile robot with monocular vision. In *Proceedings of the 2008 IEEE Conference on Robotics, Automation and Mechatronics*, pages 1005–1010, 2008.

[63] R. O. Reynolds, P. H. Smith, L. S. Bell, and H. U. Keller. The design of Mars lander cameras for Mars Pathfinder, Mars Surveyor '98 and Mars Surveyor '01. *IEEE Transactions on Instrumentation and Measurement*, 50(1):63–71, 2001.

[64] Thomas Lemaire, Cyrille Berger, Il-Kyun Jung, and Simon Lacroix. Vision-based SLAM: stereo and monocular approaches. *International Journal of Computer Vision*, 74(3):343–364, 2007.

[65] E. Royer, J. Bom, M. Dhome, B. Thuilot, M. Lhuillier, and F. Marmoiton. Outdoor autonomous navigation using monocular vision. In *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, pages 1253–1258, 2005.

[66] Zhanyu Zhang, Yalou Huang, Chao Li, and Yewei Kang. Monocular vision simultaneous localization and mapping using SURF. In *Proceedings of the 7th World*

*Congress on Intelligent Control and Automation (WCICA)*, pages 1651–1656, 2008.

[67] J. Lobo and J. Dias. Integration of inertial information with vision towards robot autonomy. In *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE)*, volume 3, pages 825–830, 1997.

[68] V. Sazdovski and P. M. G. Silson. Inertial navigation aided by vision-based simultaneous localization and mapping. *IEEE Sensors Journal*, 11(8):1646–1656, 2011.

[69] T. Oskiper, Zhiwei Zhu, S. Samarasekera, and R. Kumar. Visual odometry system using multiple stereo cameras and inertial measurement unit. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[70] Dae Hee Won, Sebum Chun, Sangkyung Sung, Young Jae Lee, Jeongho Cho, Jungmin Joo, and Jungkeun Park. INS/vSLAM system using distributed particle filter. *International Journal of Control, Automation and Systems*, 8(6):1232–1240, 2010.

[71] Gabriel Nutzi, Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. Fusion of IMU and vision for absolute scale estimation in monocular SLAM. *Journal of Intelligent & Robotic Systems*, 61:287–299, 2011.

[72] Jonathan Kelly and Gaurav S. Sukhatme. Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *The International Journal of Robotics Research*, 30(1):56–79, January, 2011.

[73] Agostino Martinelli. Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. *IEEE Transactions on Robotics*, 28(1):44–60, 2012.

[74] J. O. Cappellari Jr., J. S. Dudek, and R. L. Wagner. Project Apollo coordinate system standards. Technical Report NASA-CR-124854, NASA, 1965.

[75] Jacob H. Christensen, David B. Anderson, Mark E. Greenman, and Bryan D. Hansen. Scalable network approach for the space plug-and-play architecture. In *Proceedings of the 2012 IEEE Aerospace Conference*, Big Sky, Montana, March 3-10, 2012.

[76] Ksenia Kolcio-Prather, Stuart Parker, and Paul Graven. Investigating latency in PnP GN&C architectures. In *Proceedings of the 2012 IEEE Aerospace Conference*, Big Sky, Montana, March 3-10, 2012.

[77] Dov Dori. *Object-Process Methodology : A Holistic Systems Paradigm.* Springer, Berlin ; New York, 2002.

[78] Jan Wendel, Oliver Meister, Christian Schlaile, and Gert F. Trommer. An integrated GPS/MEMS-IMU navigation system for an autonomous helicopter. *Aerospace Science and Technology*, 10(6):527–533, 2006.

[79] Sameh Nassar, Zainab Syed, Xiaoji Niu, and Naser El-Sheimy. Improving MEMS IMU/GPS systems for accurate land-based navigation applications. In *Proceedings of the 2006 National Technical Meeting of The Institute of Navigation*, volume 1, pages 523–529, Monterey, California, January 18-20, 2006.

[80] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV)*, volume II, pages 1508–1515, Beijing, China, October 17-20, 2005.

[81] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *9th European Conference on Computer Vision (ECCV)*, volume 3951 LNCS, pages 430–443, Graz, Austria, May 7-13, 2006.

[82] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[83] Stephen J. Kapurch and Neil E. Rainwater. *NASA Systems Engineering Handbook*, pages 296–297. National Aeronautics and Space Administration, Washington, D.C., 2007.