THEORY AND APPLICATIONS OF GRAM-SCALE SPACECRAFT

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Van Hunter Adams

May 2020

THEORY AND APPLICATIONS OF GRAM-SCALE SPACECRAFT

Van Hunter Adams, Ph.D.

Cornell University 2020

This dissertation introduces R-selected spacecraft as a field of study that draws from concepts in ecology, and introduces the Monarch spacecraft as a case study for a system designed in accordance with the principles of this field. The Monarch is a 2.5-gram spacecraft that is the first to trade quantity, rather than cost, for low mission risk. By taking advantage of recent technological advancements in unrelated disciplines and taking a statistical approach to mission assurance, R-selected spacecraft open the door to an entirely new paradigm in space access and exploration. This dissertation describes the challenges and advantages unique to gram-scale, R-selected spacecraft. It also presents a number of use cases — involving distributed in-situ sensing and planetary science — that are unique to spacecraft of the Monarch's diminutive size and large quantity. This dissertation presents a routing policy for moving information through large collections of Monarchs in low-Earth orbit, and results from simulated lunar impact survival tests. Demonstrations of distributed sensing, leaderless cooperation, routing, and actuation are presented and discussed to illustrate the viability of some entirely new mission concepts. The final chapters anticipate future capabilities for Monarchs and present a method for extracting insights from the sorts of datasets which swarms of Monarchs will produce. The appendices discuss applications for distributed in-situ sensing in digital agriculture, and present datasets gathered by the Monarchs from vineyards and dairy calves.

BIOGRAPHICAL SKETCH

Hunter's parents, Anne and Van Adams, raised him in Lebanon, Pennsylvania. He grew up with two older sisters, Brittany and Kelly, and a younger brother, Chad. His sister, Kelly, got him interested in science and technology in elementary school. He looked up to her (and still does). She liked science and technology, so he did too.

At 11, Hunter read the introduction to Bill Bryson's *A Short History of Nearly Everything*, which sparked an interest in physics. A number of other books throughout middle and high school, and the television show *Mythbusters*, kindled that interest. He matriculated at Cornell to study physics in 2011.

Hunter spent his first two years at Cornell working in the Laboratory for Elementary Particle Physics and running varsity cross country/track and field. He joined Mason Peck's spacecraft research group at the beginning of his second year and discovered a passion for engineering. That passion led him to manage a team of engineers constructing a spacecraft called Violet, work for a short time at SpaceX, and, ultimately, to start his PhD with Mason. He both finished his degree in physics and started his PhD in engineering in 2015.

Hunter's PhD was almost entirely devoted to hardware work - specifically to building chipsats and the infrastructure for communicating with them. He also worked extensively on estimation problems, navigation, and on-orbit routing mechanisms. Hunter is curious about a lot of topics. His curiosity led to collaborations with researchers in plant sciences, veterinary science, electrical engineering, ornithology, and computer science. He started a company based on his research. With that company, he will pursue a career building things that interest him and add value to the world.

To my brother.

ACKNOWLEDGEMENTS

For their academic guidance and technical help, thank you to Mason Peck, Bruce Land, Dmitry Savransky, Kirstin Peterson, Justine Vanden Heuvel, Francisco Yeal-Lepes, Mark Psiaki, and Hod Lipson.

For their help around the lab, thank you to Stewart Aslan, Gabe Rampichini, Matt DiStefano, Sam Feibel, Philip Whitmarsh, Alexander Mayo-Smith, Sebastian Colom, Johnny Crossman, and Ryan Begin.

For the conversations, and for making these years so enjoyable, thank you to all the members of the Space Systems Design Studio. Elise Eckman, Zac Manchester, Katherine Wilson, Matt Walsh, Frances Zhu, Kalani Danas Rivera, Aneesh Heintz, Doga Yucalan, Sawyer Elliot, Josh Umansky-Castro, and Aaron Zucherman.

For their help with commercialization, thank you to Tom Schryver, Ken Rother, Andrea Ippolito, Brad Treat, Steven Gal, Rodrigo Taipe, David Erickson, Dan Olmstead, Peter Martini, Tim Martinson, Sasank Vemulapati, Nicole Diamantides, Kevin O'Brien, and Taylor Clawson.

And for everything I am, thank you to my family.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION: R-SELECTION AND MISSION ASSURANCE

Spacecraft design has historically traded high-cost development and engineering for low mission risk. This successful model has changed the world. It has brought us decades of discovery and exploration, rewriting the textbooks on planetary science, heliophysics, Earth science, and astrophysics. But despite how well it has served the scientific community, this model limits the types of missions that we can perform.

There is no shortage of threats to the survival and operation of a spacecraft. Wide temperature swings, many forms of radiation damage, and impacts with micrometeoroids or larger objects are only a few that flight hardware experiences. Vast communication distances, a dearth of resources for power scavenging, and launch mass-related challenges in power storage and generation top the list of technological barriers [42]. Unlike for most everyday, terrestrial engineering problems, the cost and high stakes of spacecraft motivate a formal process, what engineering organizations know as mission assurance. Metrics for mission assurance attempt to capture the probability of overcoming these threats and achieving mission success. For conventional missions involving a single, high-cost spacecraft, mission assurance essentially reduces to the probability of spacecraft success, which rarely exceeds about 95%. And unless the spacecraft of interest is far more valuable than a single launch — examples include the International Space Station and the Hubble Space Telescope — we never fix them when they fail. In fact, until the past decade, we gave little consideration to servicing and repair in the design of spacecraft, making them virtually impossible to fix even if we wanted to do so [37].

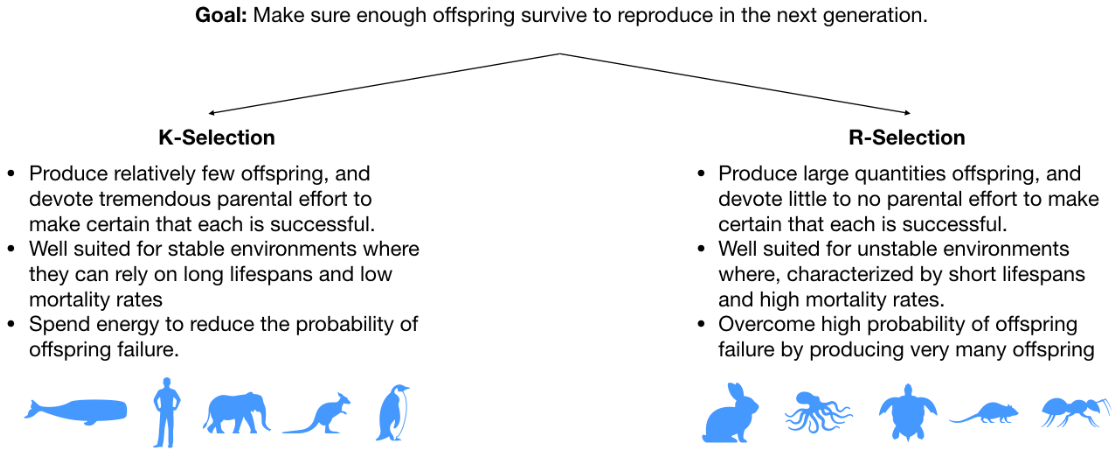
Conventional metrics rarely consider how confidence measures change for missions that include many identical spacecraft, since to date it has been cost prohibitive to do so except in rare cases [9, 27, 50, 13]. If one could launch thousands of identical spacecraft, confidence in any particular one might be extremely low while confidence that some critical number remains operational could remain high. This probabilistic model is a fundamental motivation for the Monarch. Rather than trading high cost for low risk, Monarchs trade high quantity for low risk. It is an idea that would have been impossible to realize until only a few years ago.

The notion of trading quantity for risk is not without precedent in nature, and is particularly apparent in reproductive strategies. Evolution has arrived at two general solutions to the problem of maintaining a viable population from one generation to the next. Some creatures, like humans or whales, employ K-selection. K-selected species produce a relatively small number of offspring and spend a tremendous amount of time and energy to make certain that each child is successful. These animals are well suited to stable environments where they can rely on long lifespans and a low mortality rates. This strategy is clearly a successful one, as evidenced by the existence of all creatures that use it. There is, however, an alternative solution.

Other creatures, like sea turtles, produce a relatively large amount of offspring and put extremely little investment into any one of them. The probability of survival for any particular offspring can be extremely low, but as long as enough are produced then the population remains healthy. These are R-selected species. R-selected species tend to have shorter lifespans than K-selected species, faster sexual maturation rates, and larger numbers of offspring.

They are far better suited to unstable environments than K-selected species [33]. See Fig. 1.1.

This dissertation asserts that spacecraft engineers have something to learn from nature in this regard. Every spacecraft that humans have launched has been K-selected. Engineers produce very few spacecraft in a lifetime, and they devote an extreme amount of time, money, and energy into each of these spacecraft to be as certain as possible that it will survive for as long as intended. See Fig. 1.2. K-selection has been the design paradigm for spacecraft strictly out of necessity. Solving the mission assurance problem in the statistical manner of R-selection requires spacecraft that can be manufactured at much lower cost than conventional, K-selected spacecraft, launched in much greater quantity, and with a much faster development cycle. This has not been possible until only very recently when other industries (mostly cell phone and gaming industries) drove down the cost of automated circuit board manufacturing and assembly, processors, and surface-mounted sensors. The world has not yet seen R-selected spacecraft because it has never before been possible to manufacture



Goal: Make sure enough offspring survive to reproduce in the next generation.

K-Selection

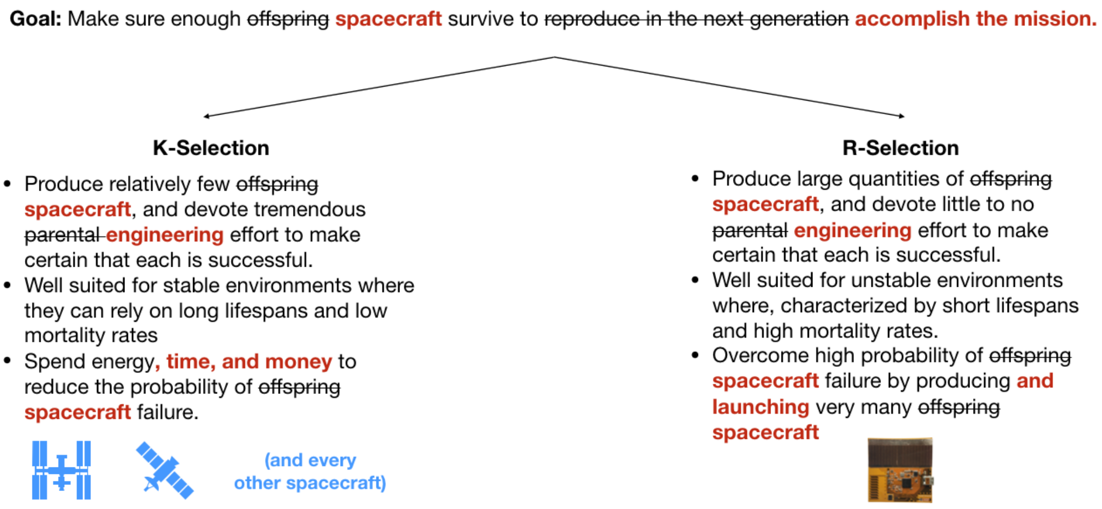
- Produce relatively few offspring, and devote tremendous parental effort to make certain that each is successful.
- Well suited for stable environments where they can rely on long lifespans and low mortality rates
- Spend energy to reduce the probability of offspring failure.

R-Selection

- Produce large quantities offspring, and devote little to no parental effort to make certain that each is successful.
- Well suited for unstable environments where, characterized by short lifespans and high mortality rates.
- Overcome high probability of offspring failure by producing very many offspring

Figure 1.1: R and K selection in nature.

R-selected spacecraft. The systems and programs that have come closest to this sort of architecture include Globalstar, Iridium, Orbcomm, and the Educational Launch of Nanosatellites (ELaNA). In these systems, the success of the mission or the program does not depend on the success of every individual satellite, and the system is robust to a small number of individual satellite failures [46]. These systems are best classified, however, as failure tolerant rather than failure reliant. In much the same way that a pride of lions may continue to survive after the loss of a few individuals, Globalstar, Iridium, Orbcomm, and ELaNA will continue to operate after the loss of a few spacecraft. This is a different sort of risk management technique than that used by a decidedly R-selected species, like ants, where loss is a critical and expected part of the survival strategy for the colony. It is now possible to build spacecraft with this decidedly R-selected approach to mission assurance. R-selected spacecraft represent a paradigm shift away from failure-tolerant systems and toward failure-reliant systems where mission assurance is based on the statistics of survival, rather than failure rates.



Goal: Make sure enough ~~offspring~~ **spacecraft** survive to ~~reproduce in the next generation~~ **accomplish the mission.**

K-Selection

- Produce relatively few ~~offspring~~ **spacecraft**, and devote tremendous ~~parental~~ **engineering** effort to make certain that each is successful.
- Well suited for stable environments where they can rely on long lifespans and low mortality rates
- Spend energy**, time, and money** to reduce the probability of ~~offspring~~ **spacecraft** failure.

(and every other spacecraft)

R-Selection

- Produce large quantities of ~~offspring~~ **spacecraft**, and devote little to no ~~parental~~ **engineering** effort to make certain that each is successful.
- Well suited for unstable environments where, characterized by short lifespans and high mortality rates.
- Overcome high probability of ~~offspring~~ **spacecraft** failure by producing **and launching** very many ~~offspring~~ **spacecraft**

Figure 1.2: R and K selection in spacecraft.

This ushers in an entirely new field of study within aerospace engineering.

The list of open research questions associated with building and utilizing R-selected spacecraft is nearly as extensive as the list of open research questions for conventional spacecraft was in the 1950s-60s. The open questions are fundamental ones about sending and receiving data to these systems, controlling the trajectory and orientation of each spacecraft and of the collection of spacecraft, and basic design principles. These are not issues of incremental improvement on existing technology, they are fundamental questions about construction and utilization of a new kind of space system. Answering these questions will bring space exploration and planetary science of an unprecedented variety. This dissertation describes the first space system designed according to this new philosophy.

The Monarch, shown in Fig. 1.3, is the first attempt to apply R-selection to spacecraft, and that brings with it the same advantages and disadvantages found in nature. Each satellite has a far higher probability of failure than any conventional K-selected spacecraft, but, just as in nature, that probability of failure is offset by the number of spacecraft that can be launched at a single time. Their quantity makes them well suited for unstable environments and dangerous missions, since they are not beholden to the probability of failure, like conventional spacecraft, but instead exploit the probability of failure. R-selected spacecraft have their own separate and unique set of use cases that are apart from those of conventional spacecraft. What follows is an overview of the challenges and advantages unique to gram-scale R-selected spacecraft. This dissertation also presents a number of use cases — involving distributed in-situ sensing and planetary science — that are unique to spacecraft of the Monarch's diminutive size and large quantity. Results from simulated lunar impact survival tests and a case-study planetary science mission are presented and discussed, suggesting one particular use case. Demonstrations of distributed sensing, leaderless cooperation, routing, actuation, GPS acquisition, and powering are presented to illustrate the viability of some entirely new mission concepts. In the final chapters, I anticipate future capabilities for Monarchs and present a method for extracting insights from the sorts of datasets which swarms of Monarchs will produce. The appendices discuss applications for distributed in-situ sensing in digital agriculture, and present datasets gathered by the Monarchs from vineyards and dairy calves.

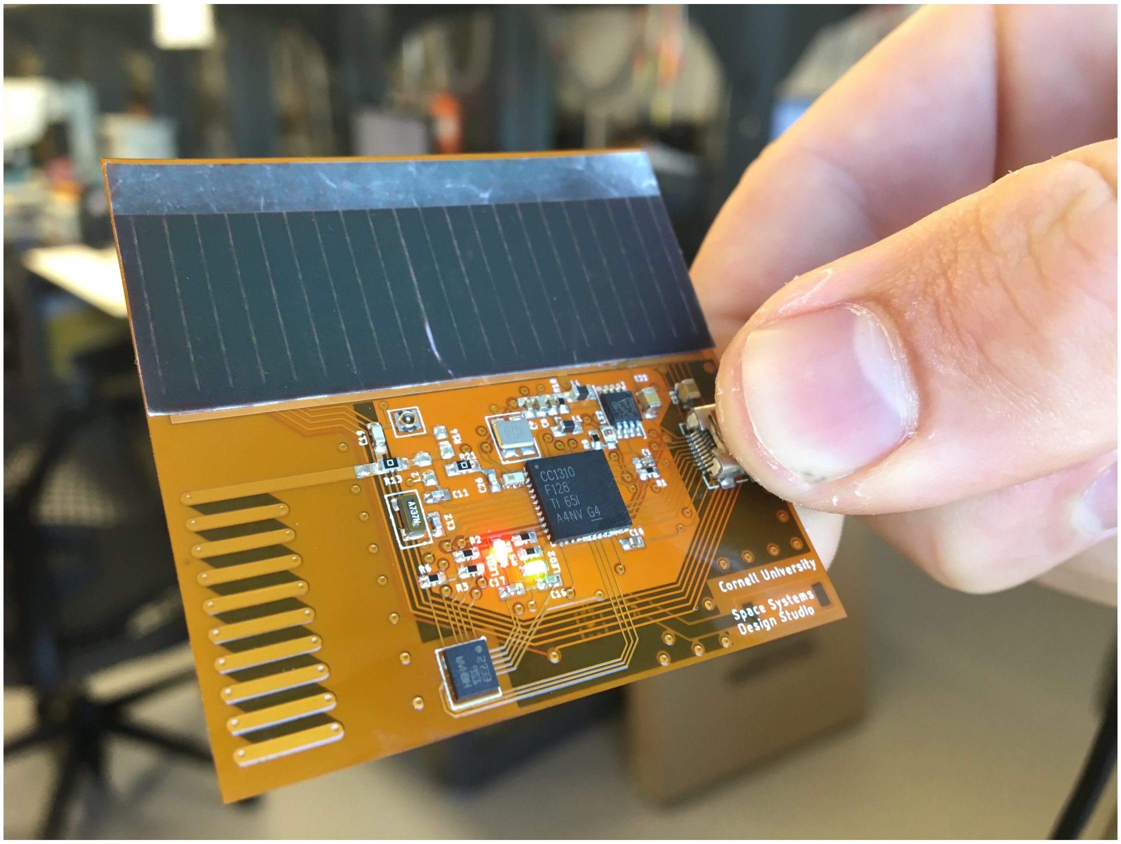


Figure 1.3: A Monarch spacecraft

CHAPTER 2

THE MONARCH SPACECRAFT

Designing a mission that trades high quantity for low risk rather than high cost for low risk requires a spacecraft that can be manufactured cheaply and in bulk, can launch and deploy in much greater quantities than conventional spacecraft, and can maintain the core capabilities required for it to be useful. These goals are now achievable. The economies of scale driven largely by the consumer-electronics industry (specifically, cell phones and gaming) have reduced the cost of surface-mounted processors, sensors, and radios to a tiny fraction of what they were just a decade ago [38]. This revolution has also driven down the cost and timeframe for manufacturing and assembly of printed circuit boards. The Monarch takes advantage of both of these trends. It is a spacecraft built through entirely automated processes, the same processes that build circuit boards for cell phones and other electronics. Monarchs use sensors and processors from game controllers, laptops, and other consumer-market electronics for which economies of scale have driven down component costs. The result is a 2.5 gram spacecraft that can be manufactured in bulk for less than $50 apiece, launched and deployed by the hundreds or thousands, and can go places and do things that conventional spacecraft cannot. Fig. 2.1 shows the front and rear of the spacecraft. The components are labeled. Monarchs are not small versions of large spacecraft, and they do not replace conventional spacecraft. Instead, they are a new way to access and explore space, and they have their own new and unique use cases.

The Monarch is an example of what has come to be known as a chipsat, a concept whose development began in earnest at Cornell University in 2007, al-

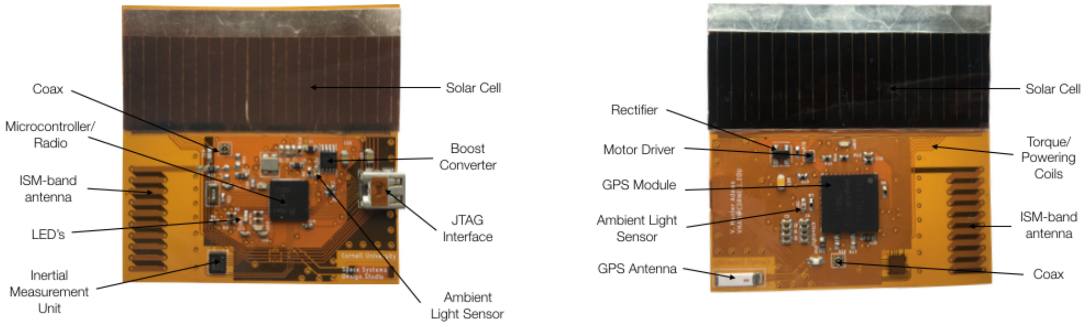


Figure 2.1: Monarch spacecraft with consumer-market electronics components labeled.

though earlier work at the Aerospace Corporation in 1999 offered insight into what might be possible at this scale [15]. The first publications by Atchison et al described spaceflight dynamics at the microscale. The surprising benefits of small scale, such as the importance of effects like solar pressure, drag, and the Lorentz force in Earth orbit to alter trajectories in unfamiliar ways, motivated the creation of a prototype, small-scale free-flyer to verify these effects experimentally. From 2007 through 2016, Cornell's research focused on Sprite, the name Atchison gave them. Sprites were 4-gram femtosatellites or chipsats, which have now flown four times (on the International Space Station in 2011, on Kicksat-1 in 2014, Venta-1 in 2016, and KickSat-II in 2019), with an additional mission planned in 2020. Kicksat-1 was the world's first crowd-funded spacecraft (via Kickstarter.com), almost singlehandedly designed and built by Zac Manchester, then a student at Cornell and now on the faculty at Stanford. Kicksat-1 took 104 early-generation Sprites to orbit [24]. Kicksat-2 carried 128. A Sprite on The Venta-1 mission — again, Manchester's work — established the feasibility of communicating across large distances with low power: 10 mW transmission reached over 1500 km with suitable forward-error correction, requiring only a laptop and HAM radio antenna. With their exceptionally low ballistic coefficient, atmospheric drag deorbits the chipsats in a matter of days, as shown in Chapter 3 and validated by the KickSat-2 mission. Different debris risk mitigation strategies must be employed at higher orbits. Such strategies may include building the spacecraft of a material that will sublime away, or giving them thrust capability for escaping or entering the atmosphere.

The Monarch has advanced well beyond these early efforts. Here, we describe the Monarch in terms of the subsystems associated with larger, conventional spacecraft, as labeled in Fig. 2.2. These subsystems include telemetry and command, power generation, attitude determination and control, navigation, and payload [50]. The size of the Monarch makes some of these subsystems dif-

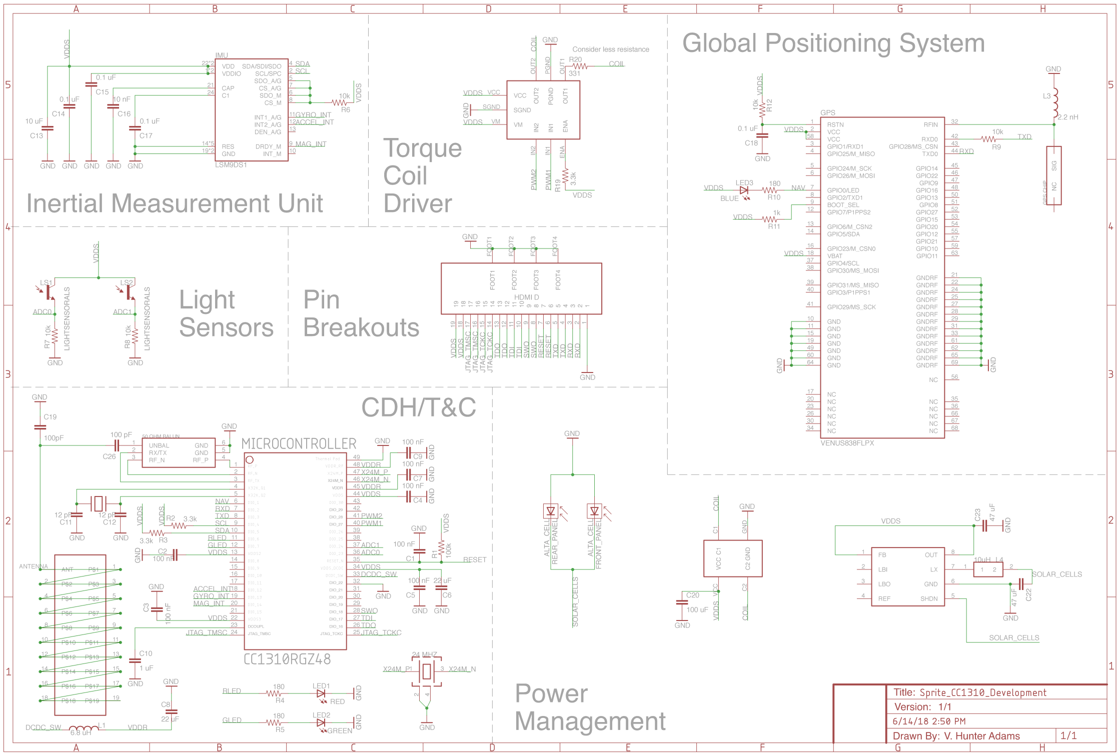


Figure 2.2: Electrical schematic for Monarch spacecraft.

ferent from their larger-spacecraft analogues, and it couples some subsystems that are not coupled in larger spacecraft. The fundamental concept of trading quantity for risk finds its way into each of these subsystems.

2.1 Telemetry and Command

Telemetry and command takes place via a 25 mW ISM-band radio and and embedded PCB antenna [17]. With such a low-power transmitter, and without the ability to accommodate a high-gain antenna, the data rate from any particular Monarch is substantially lower than larger spacecraft with more power availability and directed, high-gain antennas. With some reasonable assumptions on the parameters associated with the communication system (500 km transmission distance, isotropic transmission antenna, 915 MHz carrier frequency, 7dB receiver antenna, 64 kHz bandwidth), it can be shown that the Shannon Limit for a Monarch in Earth orbit is approximately 84 kilobits per second [12, 41, 6]. Thus, if a line rate less than 84 kbps is used, there exists a coding technique (involving error correction) that allows the probability of error at the receiver to be made arbitrarily small.

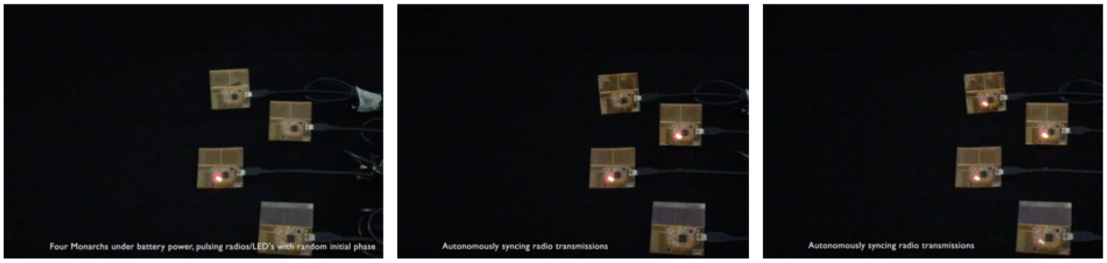

Four Monarchs under battery power, pulsing radios/LED's with random initial phase
Autonomously syncing radio transmissions
Autonomously syncing radio transmissions

Figure 2.3: Autonomous radio synchronization.

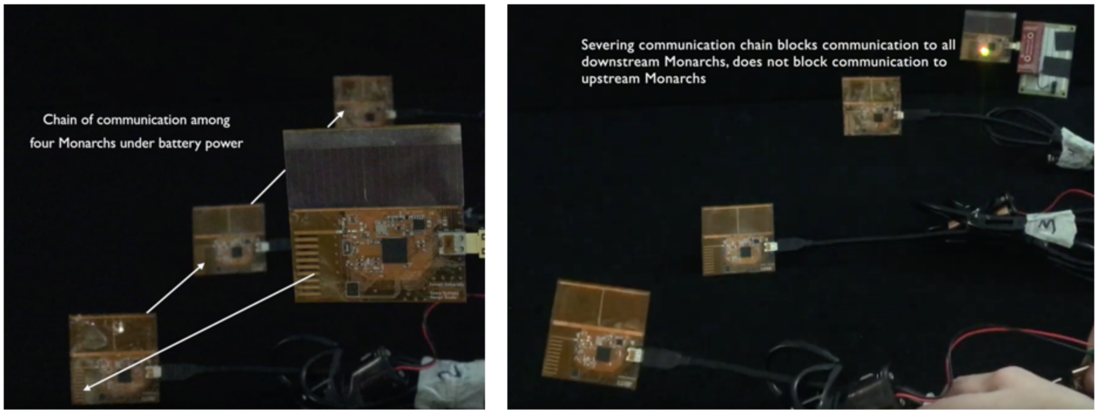

Chain of communication among
four Monarchs under battery power
Severing communication chain blocks communication to all
downstream Monarchs, does not block communication to
upstream Monarchs

Figure 2.4: Packet routing.

However, these comparatively low transmission rates per Monarch are not the proper metric to consider, since many hundreds or thousands of Monarchs may be deployed simultaneously, each of which may communicate data at this comparatively low transmission rate. This is how the notion of quantity vs. cost finds its way into this subsystem. The data rate from the entire collection is competitive with large, high-power spacecraft and, furthermore, the dataset is of an entirely different sort. Rather than receiving large amounts of data from a few sensors on a single spacecraft at a single location, a dataset from a collection of Monarchs comes from many thousands of sensors distributed across vast regions of space. This distribution creates the opportunity for entirely new sorts of missions.

A series of technical demonstrations proved the viability of many aspects of this subsystem. To demonstrate leaderless cooperation and networking, a collection of Monarchs were programmed as pulse-coupled oscillators. See Fig. 2.3. They began beaconing radio packets at random intervals and, over the course of a few minutes, synchronized their transmissions. In a second demonstration, the same collection of Monarchs was programmed to only send/receive transmissions from a single other Monarch. Breaking the chain of communication caused all downstream Monarchs to lose access to information from those upstream. See Fig. 2.4. This demonstrates the ability of the Monarchs to send data through specified paths in a collection.

2.2 Power and Thermal

Trading high quantity for low risk also affects the Monarchs power subsystem. On large, conventional spacecraft, a battery keeps the spacecraft awake when it passes through the shadow of the Earth. For missions involving a single high-end spacecraft, this necessity is inescapable, since power keeps the spacecraft thermally regulated [50]. With thousands of Monarchs, power can be handled differently. At only 2.5 grams (the mass of an American penny) and with a very flat shape, Monarchs reach thermal equilibrium much faster than larger spacecraft. At their size, it costs more energy to keep a battery warm in eclipse than that battery can store when in sunlight [50]. Thus, these spacecraft have no means of thermal regulation. Instead, all sensors, processors, and components are chosen based, among other things, on their operational temperatures. This precludes the use of any battery, the operational temperatures for which are exceeded while in orbit. With a small capacitor, one that is insensitive to the thermal environment, Monarchs can continue to function at low duty cycle in eclipse. Otherwise, they sleep when in eclipse and wake when in sunlight. Networking capability ensures that a swarm or cluster of Monarchs is always on, in a generalized sense, even when a single spacecraft is unpowered. So, collecting scientific data and communicating it to Earth can continue, regard-

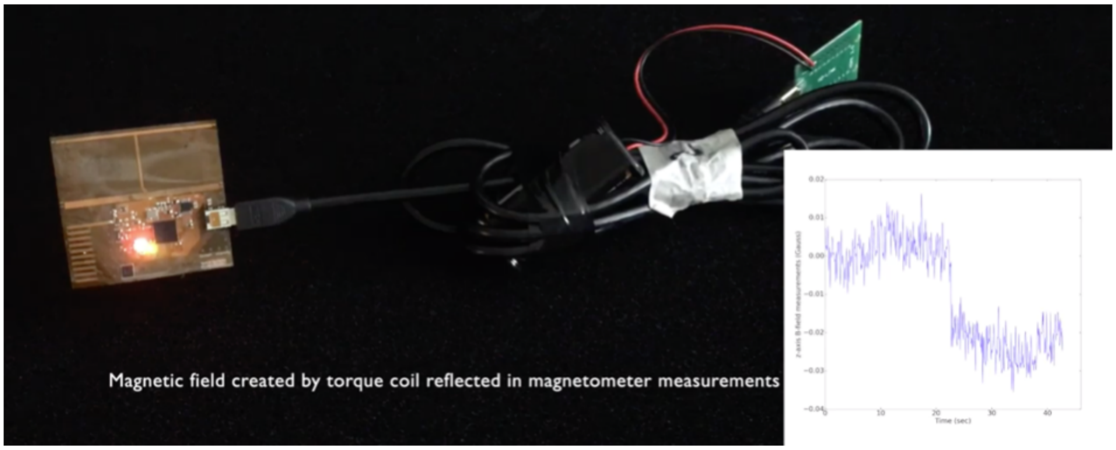


Figure 2.5: Magnetic field normal to Monarch surface, before/after torque coil is enabled.

less of the local solar flux. For missions involving monolothic spacecraft, such an operations concept would be far from optimal, and likely unacceptable. For Monarchs, however, quantity makes this arrangement perfectly adequate.

2.3 Attitude Determination and Control

Attitude determination looks very much the same on Monarchs as on large conventional spacecraft. In fact, the pointing agility (combining angular rate, acceleration, and so forth) is roughly independent of length scale. However, attitude and navigation are uniquely coupled for spacecraft of their size. Each Monarch carries a gyroscope, magnetometer, and light sensors that act as coarse Sun sensors. So, three-axis attitude determination is possible [26, 44, 48, 25]. Each Monarch also carries a GPS receiver and a GPS antenna, with which it may determine its location, velocity, and the absolute time when operating in Earth orbit. Attitude control is a bit more subtle on the Monarch than on a con-

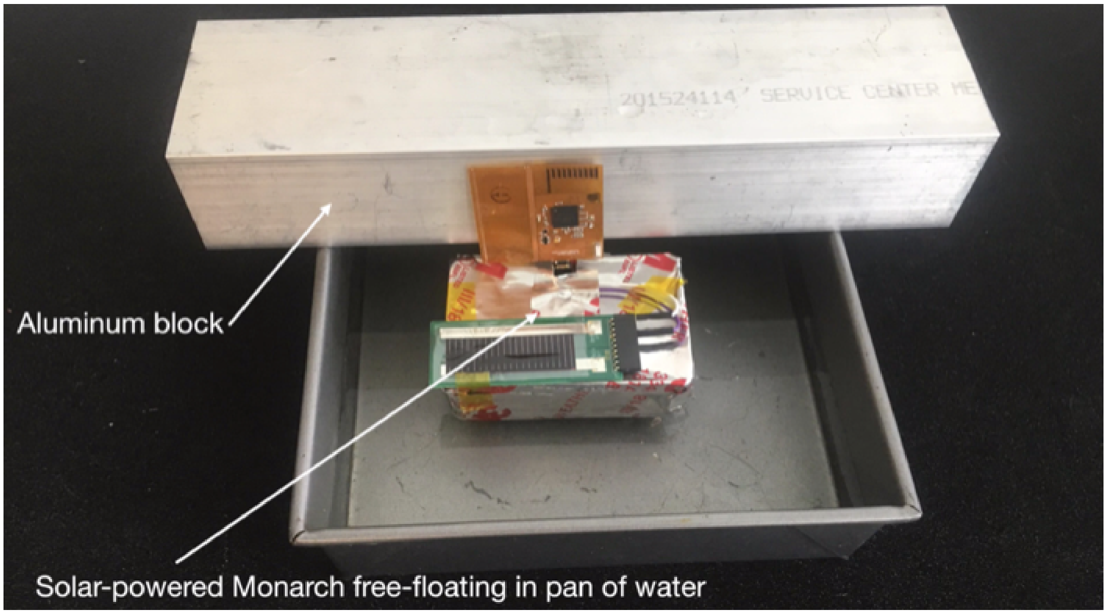


Figure 2.6: Test setup for torque coil actuation demo.

ventional spacecraft. Monarchs drive electrical current through a coil of wire embedded in their interior in order to create their own local magnetic field. Fig. 2.5 shows normal-axis magnetic field measurements from a Monarch's onboard magnetometer before and after the torque coil is turned on. This magnetic moment torques against the Earths magnetic field, thereby changing the orientation of the spacecraft. This technique is common in larger spacecraft [45, 22, 49], particularly CubeSats, but its implementation in Monarchs is unique in that the coils lie only in the plane of the printed circuit board. The inertia tensor of the Monarch is such that it is passively stable in spin about its normal axis [20]. Rather than requiring 3-axis control, Monarchs use their torque coils to induce and cease precession about the Earth's magnetic field vector during a stable spin, a 2-axis control solution. For spacecraft with area-to-mass ratios as high as that of the Monarchs, attitude and trajectory are highly coupled in low Earth orbit, where the dominant orbital perturbation is atmospheric drag [5]. As the

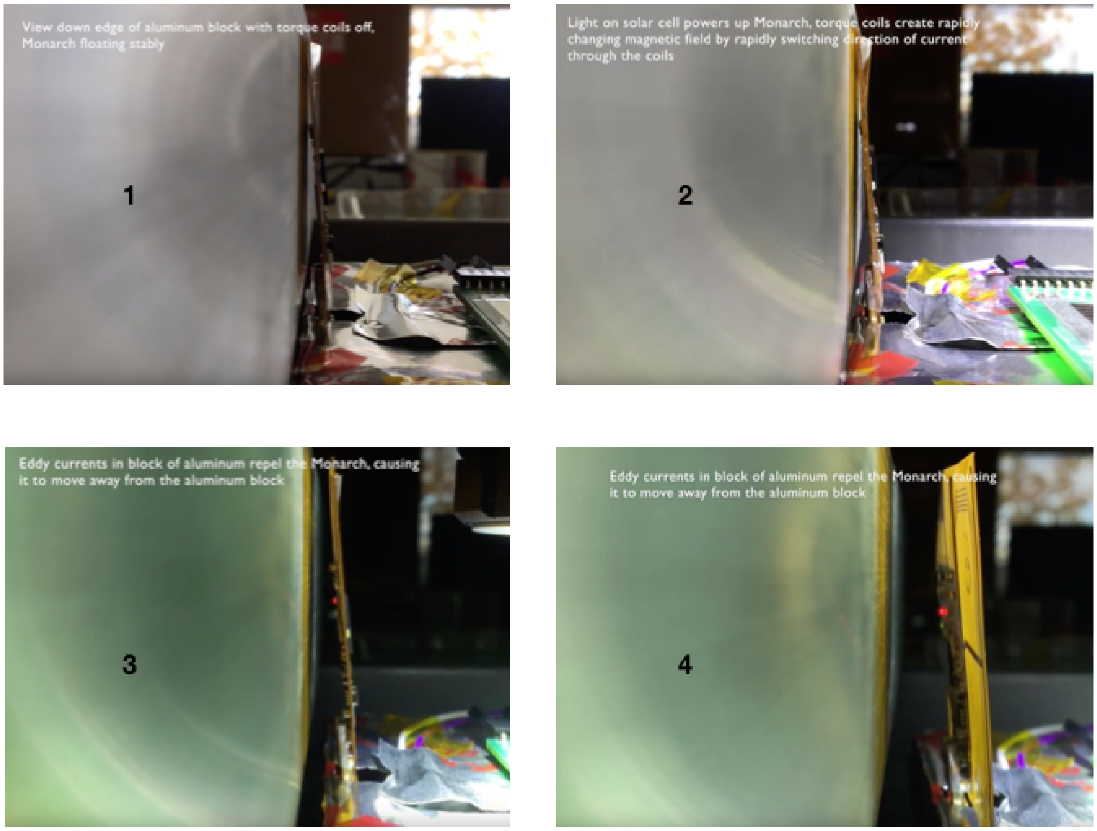

View down edge of aluminum block with torque coils off, Monarch floating stably
1
Light on solar cell powers up Monarch, torque coils create rapidly changing magnetic field by rapidly switching direction of current through the coils
2
Eddy currents in block of aluminum repel the Monarch, causing it to move away from the aluminum block
3
Eddy currents in block of aluminum repel the Monarch, causing it to move away from the aluminum block
4

Figure 2.7: Monarch actuating via torque coils.

Monarch leans its flat face into the velocity direction, drag slows it relative to other Monarchs whose thin edge faces the velocity direction. In changing their orientation, a swarm of Monarchs can both affect power generation and manage the shape of the swarm. This capability also has implications for the sorts of missions for which Monarchs are well suited. Figs. 2.6 and 2.7 show a demonstration of a Monarch actuating with its onboard torque coils. A solar-powered Monarch is free-floating in a pan of water, with its face against a block of aluminum. When powered via light, the Monarch sets up an oscillating current through its torque coils which induces eddy currents in the block of aluminum, repellng the Monarch. This is a different sort of actuation than will be used in space, but it shows that the torque coils can indeed actuate a Monarch.

2.4 Payload

Payloads for Monarchs are different from payloads for conventional spacecraft. Their size necessarily limits the aperture, which precludes remote-sensing payloads. Large spacecraft will always be better suited for remote sensing. Instead, Monarchs are well suited for carrying sensors that measure characteristics of the environment in the immediate vicinity of the spacecraft — quantities including temperature, pressure, electromagnetic fields, particle distribution, radiation, etc. It is best to think of a collection of Monarchs as a single radio-networked sensor, each node of which remotely reports its local in-situ measurements. Such a collection gathers data of the spatial breadth associated with remote sensors, but with the localized depth of in-situ sensors. Monarchs enable missions of two very broad types: those that involve spatially distributed in-situ measurement, and those that involve actions that pose extremely high risk to individual spacecraft. They offer in-situ measurements with remote delivery of data. Section 4.4 and the appendices present and explain some distributed in-situ datasets that the Monarchs gathered on Earth.

CHAPTER 3

ROUTING MECHANISM FOR DISTRIBUTED MONITORING MISSIONS

3.1 Background and Motivation

As discussed in Chapter 1, chipsats do not replace conventional spacecraft. They have their own new and unique set of use cases. These use cases include missions involving massively distributed sensing and distributed monitoring. In distributed sensing missions, many thousands of chipsats are distributed over a vast area and each communicates local measurements through the collection and back to an operator. In distributed monitoring missions, the chipsats are again spread out over a vast area, but each reports detection of some stim-

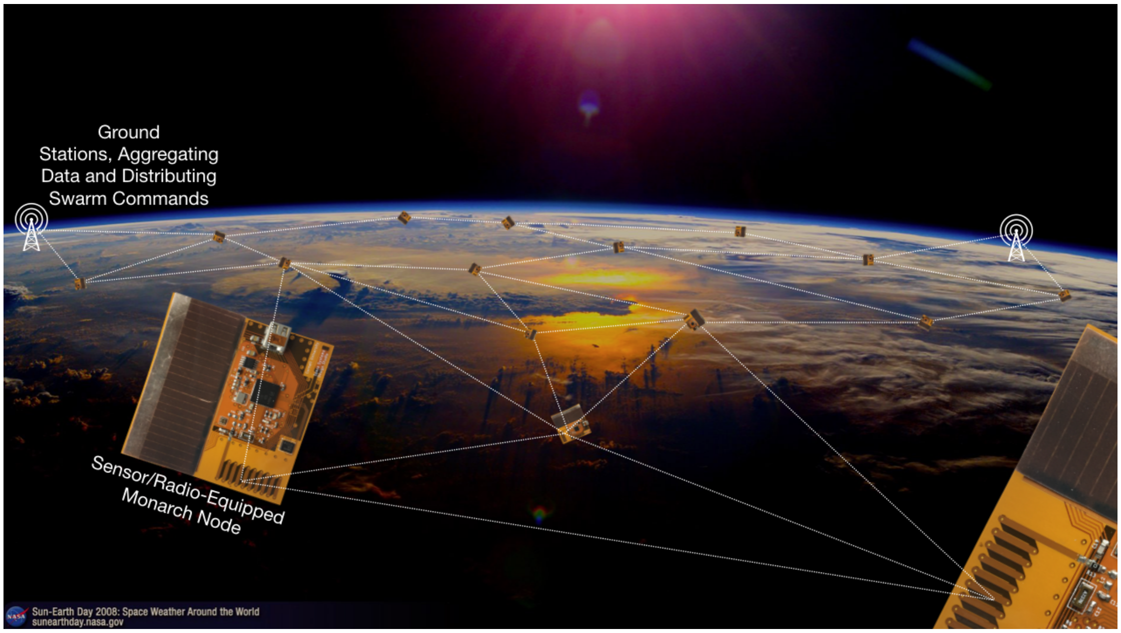

Ground Stations, Aggregating Data and Distributing Swarm Commands
Sensor/Radio-Equipped Monarch Node
Sun-Earth Day 2008: Space Weather Around the World
sunearthday.nasa.gov

Figure 3.1: Artistic representation of a swarm of networked Monarchs performing a distributed in-situ sensing mission in low-Earth orbit.

ulus (a solar event, activity of a spacecraft under surveillance, etc). Distributed sensing requires high-bandwidth communication of information through the network of chipsats, distributed monitoring requires comparatively low bandwidth. These constellations could be deployed in LEO for upper atmospheric or heliophysics studies, for spacecraft surveillance, or distributed monitoring for solar flares. Alternatively, they could be used in deep space for missions involving distributed plume sampling around Enceladus or planetary impact missions. All of these missions require exfiltration of data from the collection of chipsats to a receiver station. Fig. 3.1 shows an artistic representation of a distributed sensing/monitoring mission.

The conventional method for interacting with a spacecraft is to send it commands and receive telemetry when it travels within range of a ground station. Between passes, a conventional spacecraft will log data to onboard memory for later downlink [4]. There are rare examples of collections of satellites in which individual spacecraft that are out of range of a ground station can still communicate data to that ground station by sending them through an intermediate spacecraft. The Iridium constellation, with 66 satellites, is the best known example of one of these systems. Even for a dynamic network of 66 spacecraft, the number of nodes is small enough for estimates of the positions of every node in the network to be continuously maintained. Consequently, routes through the Iridium constellation and other constellations of conventional spacecraft may be precomputed for each packet. Routing tables are continually updated as the topology changes [34, 18]. This method will not work for swarms of hundreds of thousands of chipsats, for which it is entirely intractable to maintain continuous estimates of individual node positions or to continually update routing tables.

SpaceXs Starlink will ultimately be composed of nearly 12,000 cooperative spacecraft. Each spacecraft is 227 kg, and the collection is carefully arranged into three orbital shells. Each spacecraft carries thrusters and actuators for attitude control and for maintaining formation [1]. As a consequence, routing strategies through this network can exploit the determinism introduced by its careful arrangement and maintenance. Routing paths can be established open-loop, based on the known relative positions of all nodes [14]. The same method cannot be used for chipsats, which trade orbital control for size and expense.

Because of their exceptionally small size and mass, chipsats make spacecraft constellations of an unprecedented size eminently achievable. With the same payload mass as a single Iridium spacecraft, one could launch over 275,000 Monarch spacecraft [23]. It is entirely intractable to maintain continuous estimates of individual node positions or to continually update routing tables for dynamic networks of this size. Furthermore, Monarchs sacrifice much of the capability of large, conventional spacecraft in order to achieve their tiny form factor. A practical and scalable routing mechanism for communicating data through a network of chipsats must rest on realistic assumptions for the information available to each node in the network and the capabilities of each node in the network.

This chapter argues which information should and should not be assumed to be available to each node in a network composed of an arbitrary number of chipsats, and then derives the best achievable routing mechanism for a packet through an Earth-orbiting collection of chipsats under the argued assumptions. It does so by framing the problem as a series of optimal stopping problems and applying the dynamic programming equations. Routing data from any origin node in the network to any destination node in the network can be viewed as a series of decisions. The origin node will encounter a number of intermediate chipsats as it moves along its trajectory. At each of these encounters, it must decide whether the expected time to destination is minimized by relinquishing the data that it carries to the intermediate chipsat that it has encountered, or by retaining that data for itself. This decision is repeated for every encounter with an intermediate chipsat until the data reaches its destination. Because the derived mechanism does not consider bottlenecking at ground stations, the scope of application is limited to distributed monitoring missions, wherein a collection of chipsats route low-bandwidth indication of a stimulus through the network to a receiver station.

This chapter treats only two-dimensional networks of chipsats (i.e. networks for which all chipsats occupy the same plane). It does so for three reasons. The first is that restraining oneself to two dimensions leads to the simplest form of a generalizable routing mechanism. The second reason is that this is not an unreasonable assumption for a collection of chipsats deployed from a common mothership in low-Earth orbit. At these altitudes, atmospheric drag is by far the dominant perturbing force on spacecraft of the Monarch's area to mass ratio. All chipsats deployed in low-Earth orbit will deorbit after 3-7 days, which is a long enough period of time for extensive dispersion in the direction of travel, but not a long enough time for extensive dispersion in other directions [5]. See Fig. 3.2 for results from a simulated deployment of 500 Monarchs from the International Space Station with randomized orientations and under the influence of J2 gravity and drag, which substantiates this claim. Finally, restricting oneself to coplanar orbits allows for the performance of the routing mechanism to be evaluated on an exhaustive collection of all possible configurations of orbits.

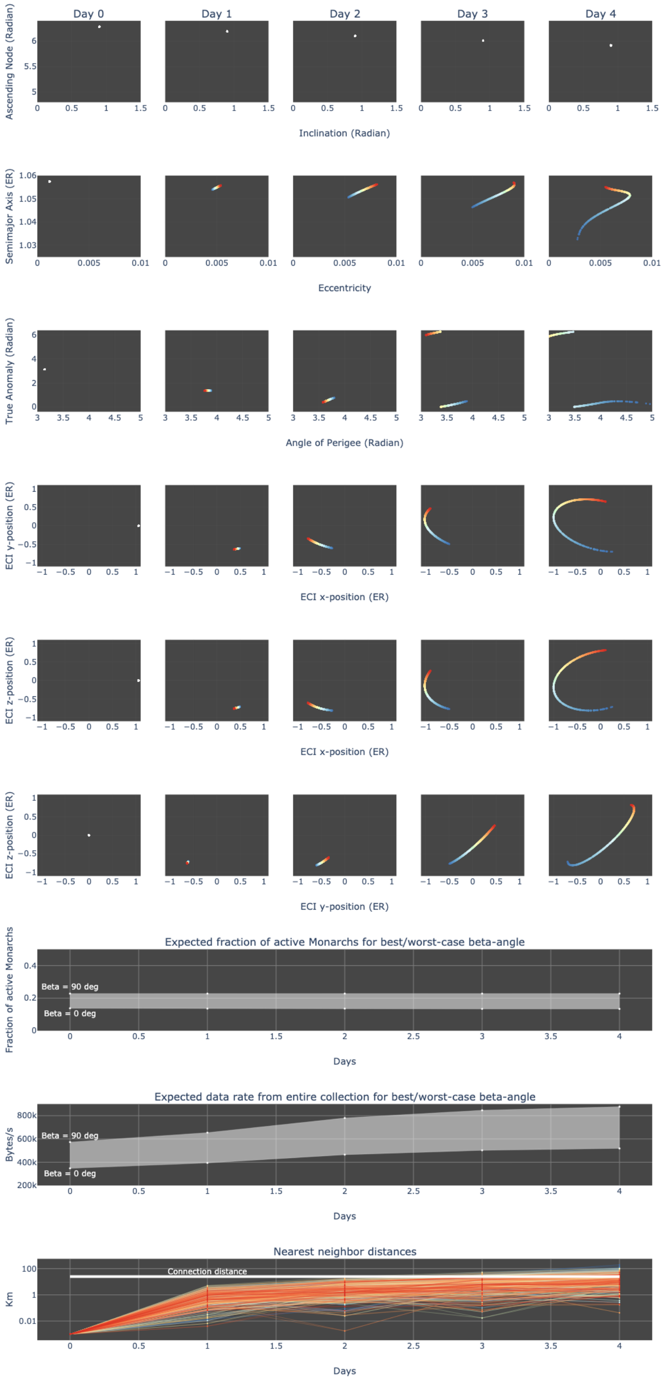


Figure 3.2: Simulation results for 500 Monarchs deployed from the International Space Station (individually colorized) with randomized orientations, under influence of J2 gravity and drag.

3.2 Assumptions

A routing mechanism's efficiency through any network is limited by knowledge of the topology of that network. In the case that one knows the exact relative positions of all nodes in a network, then one can (in principle, if not in practice) solve for the fastest path from any node in that network to any other node, or for the set of all best paths [7]. This is the case for the Iridium constellation. The speed with which one arrives at the optimal path is obviously also of relevance for any practical application. This chapter is interested exclusively in practical routing mechanisms of the sort that could be implemented on existing chipsat hardware in the immediate future, and that scale to networks of the size that chipsats enable. Deriving such a mechanism requires a realistic set of assumptions surrounding the capability of each node and the information available to each node. Each of these assumptions is stated and justified in this section.

3.2.1 Assumptions Regarding Information Availability

With respect to available information, it is assumed that each node knows its own position and velocity for all time, each node is able to measure absolute and elapsed time, and each node knows the angular rate of the Earth. The source for position, velocity, and absolute time information is the onboard GPS carried by each chipsat. The source for elapsed time is a timer in the onboard processor. The angular rate of the Earth is, of course, a parameter. This parameter is necessary for each chipsat to calculate its expected time to a ground station, which is co-rotating with the Earth, as explained in Section 3.4. It is assumed that the GPS does not enter a failure mode in which it reports erroneous positions and velocities, but is instead either totally functional or unresponsive. We also assume negligible drift in the onboard timer over the course of the chipsat's lifetime. The information that is available to each node does not require justification, given the suite of sensors with which each chipsat is equipped [2]. It is the information that is not available to each node (or to human operators or to ground-based equipment) that requires justification.

No node in the network, nor any ground-based equipment, is assumed to have knowledge of the number of nodes in the network. Chipsats are designed such that their high probability of failure (high relative to conventional spacecraft) is offset by the quantity that can be deployed at a single time. The consequence of this is that the number of functional nodes in the network will decrease from a known initial number to zero at a difficult-to-estimate rate. The efficiency of a routing mechanism through a network composed of nodes like these should not, for that reason, depend on knowledge of the number of nodes in the network. Furthermore, chipsats are so inexpensive to launch and deploy, a practical application will likely involve augmenting the network with fresh chipsats through subsequent launches. For this reason too, the routing mecha-

Table 3.1: Assumptions on information available to each node

Available	Not Available
Node's own position	Number of nodes in network
Node's own velocity	Topology of network
Time (absolute and elapsed)	Location of ground station
Angular rate of Earth	Position/velocity of any other nodes

nism should not depend on knowledge of the number of nodes that compose the network, and should instead have performance that is agnostic to this information. Section 3.6 discusses some of the practical considerations associated with replenishing a chipsat swarm from additional motherships.

The topology of the network is also assumed unknown to any of the nodes. Spacecraft with area to mass ratios as high as those of Monarchs and other chipsats are extremely susceptible to orbital perturbation in low-Earth orbit. Atmospheric drag is the most significant perturbing force, with solar pressure also contributing to alterations in trajectory. Both of these forces depend directly on the effective surface area of each chipsat, which in turn depends on the attitude of the chipsat [5]. For collections of thousands to hundreds of thousands of nodes, these perturbing forces will cause the topology of the network (i.e. which nodes communicate with which other nodes, and at which times) to change constantly and chaotically. The efficiency of the routing mechanism, therefore, should not rest on assumptions regarding the topology of the network.

The final assumption regarding information availability to each node is one that would not be strictly required but that is, in the estimation of the author, a good design decision for any practical application involving chipsats. No node is assumed to have information regarding the locations of ground stations. Unlike conventional spacecraft that will typically employ high-gain fixed antennas for high-bandwidth communication, chipsats instead communicate to the ground via handheld antennas that interface with a laptop computer [2, 24]. Rather than receiving high-bandwidth information from a single spacecraft, as is conventional, the model for chipsats is low-bandwidth communication from each of many nodes in a network. Considered as an aggregate, the data rates for a single conventional spacecraft and for a swarm of many chipsats are of the same order of magnitude, but the data rate from any particular chipsat is much lower. This enables handheld, portable receiver stations. Consequently, the performance of the routing mechanism should be agnostic to the location of the receiver station. This set of assumptions is summarized in Table 3.1.

3.2.2 Assumptions Regarding Capability

Each chipsat is equipped with a low power radio transceiver. They use these transceivers to communicate both to the ground stations and to one another. Substantial signal processing, which requires approximately the computational ability a commercially available laptop, is required by the ground stations in order to receive these transmissions [24]. Each chipsat's processor has significantly less computational ability than a laptop, and therefore the transmission distances from chipsat to chipsat are significantly shorter than from chipsat to ground [17]. The consequence is that, in a collection of very many chipsats, the individual chipsats will drift into and out of communication distance with one another. As discussed in the previous section, the particular chipsats that pass into communicable range with one another will change unpredictably as the topology of the network evolves. When two chipsats are within communicable range, they may share information with one another. When they are out of communicable range, they cannot share information and they do not store any information about the trajectory of the node with which they had previous contact (since these trajectories change so quickly, and since this too does not scale as the number of nodes increases). It is furthermore assumed that each of the chipsats uses Code Division Multiple Access (CDMA), which allows the ground station to differentiate signals from hundreds or thousands of chipsats.

3.3 Problem Statement

This chapter considers the situation in which an arbitrarily large number of chipsats are deployed from a common mothership in low-Earth orbit. The initial conditions imparted to each chipsat are randomized, with some being boosted to orbits with higher altitudes than the mothership and others landing on orbits lower than that of the mothership. This variation in eccentricity and semimajor axis, along with perturbations from atmospheric drag, solar pressure, and other higher-order effects, leads to rapid dispersion of the chipsats in the along-track direction and comparatively little dispersion in altitude, as shown in Figs. 3.3 and 3.2.

The probability density function for the position of any particular chipsat is a function only of its eccentricity and semimajor axis (see Section 3.7). After a sufficient amount of time has passed, the positions of all chipsats are well approximated by independent distributions [3]. Fig. 3.4 shows the randomly

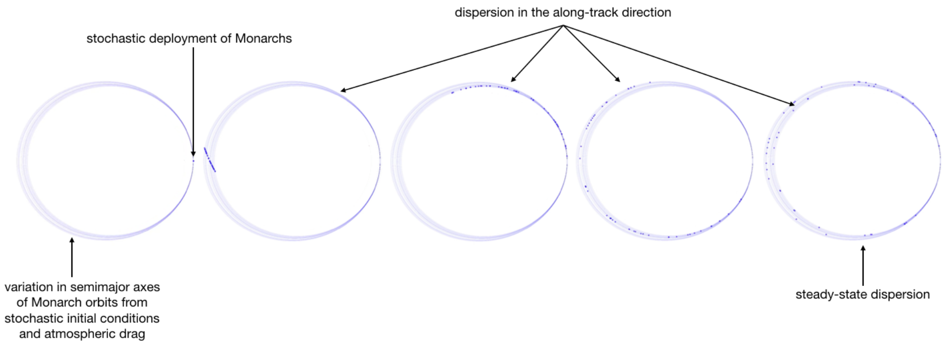
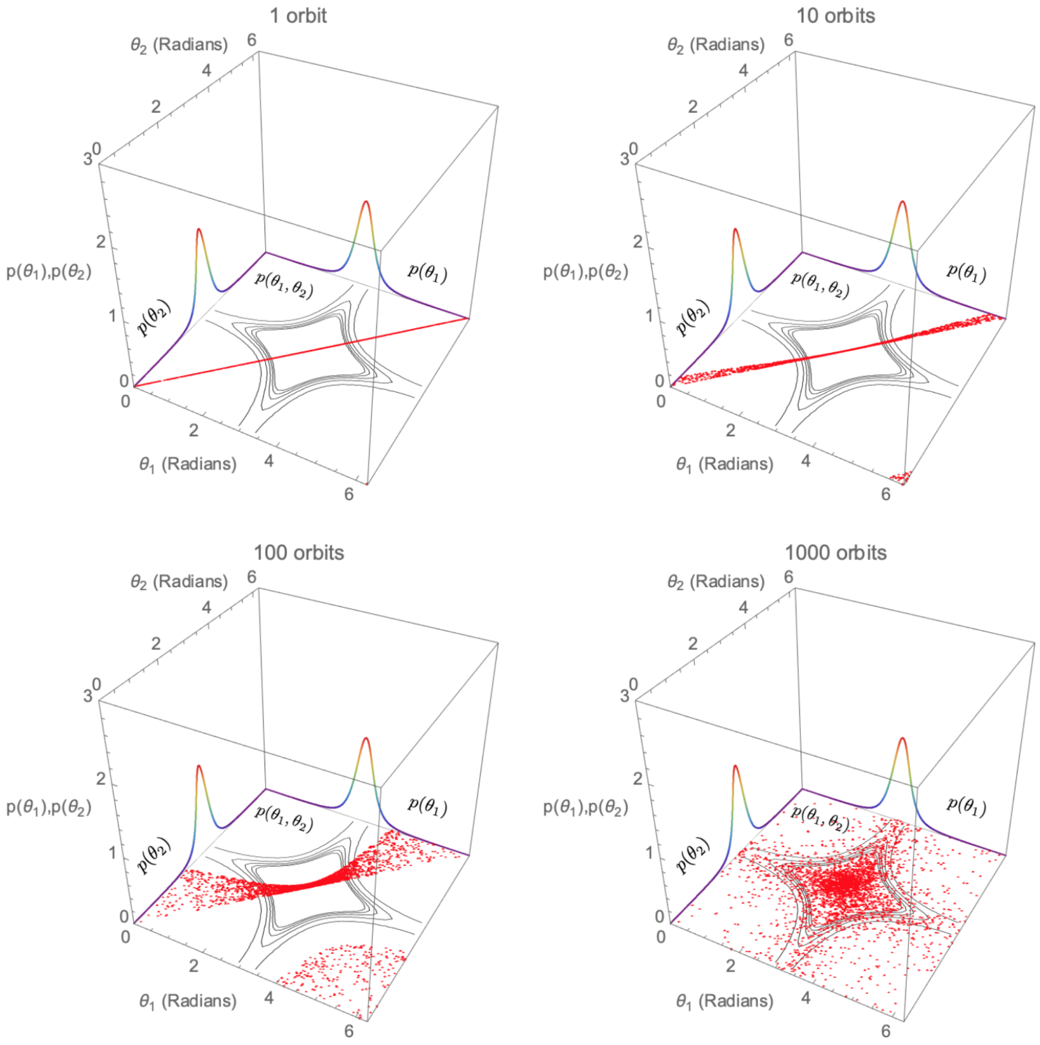

dispersion in the along-track direction
stochastic deployment of Monarchs
variation in semimajor axes of Monarch orbits from stochastic initial conditions and atmospheric drag
steady-state dispersion

Figure 3.3: Example of chipsat deployment and dispersion

sampled true anomalies of two spacecraft on orbits with slight variations in eccentricity and semimajor axis. It can be seen that, after sufficient time, the pairwise positions of these two spacecraft are well approximated by their joint probability density function, represented by the contours.


1 orbit

10 orbits

100 orbits

1000 orbits

Figure 3.4: True anomalies of two Monarchs deployed on orbits at geostationary altitude and inclination, after varying numbers of orbits. Their orbits are identical except of a variation in eccentricity by one tenth of a percent.

The objective is to develop a mechanism that will select the path which minimizes the expected time that it takes for the packet to reach the ground station, under the assumptions justified in Section 3.1. As explained in the introduction, this chapter only treats networks for which all chipsats occupy the same orbital plane (same inclination and longitude of ascending node). The network under consideration is an example of an opportunistic network, where edge connections are unpredictable and the topology of the network is not known to any of the constituent nodes. A path from a particular node to the destination may change or break during the routing process. There are two general strategies for routing through these networks: flooding-based approaches and forwarding-based approaches [16]. In flooding-based approaches, like epidemic routing, each node broadcasts the packet to every one of its neighbors until the packet reaches the destination. These approaches have the benefit of getting the data to its destination as quickly as possible, at the cost of bandwidth. Though strategies exist for reducing the overhead associated with epidemic routing [47], it still requires more resources than forwarding-based approaches.

Forwarding-based approaches select a single path to the destination, rather than letting all possible paths compete. In a forwarding-based approach, the node carrying the packet chooses which neighbor to which to handoff the data (or whether to retain the data) based on some piece of information. This information may be the proximity of each neighbor to the destination, or local knowledge of the network. This strategy has the advantage of increased bandwidth because more packets may be routed at once, since fewer nodes are occupied with a single packet than in an epidemic approach. The cost is potentially choosing a sub-optimal path [16]. This chapter presents a forwarding-based approach through an opportunistic network of chipsats, where routing decisions are based on the instantaneous orbital mechanics of neighboring chipsats. This approach has the advantage of allowing more packets to be transmitted at once than would be allowed by an epidemic approach, and the disadvantage of placing packet delivery at risk. If the chipsat carrying the packet fails, then the packet fails to reach its destination. For many distributed monitoring applications, this is an appropriate trade-off. The mechanism by which these decisions are made is derived through dynamic programming.

3.4 The Dynamic Programming Equations

The dynamic programming technique is generally useful for solving problems that involve a series of decisions. The objective is to make each decision such that the cost (some quantifiable metric for the undesirableness of an outcome) over a given number of stages is minimized. Typically, each of these decisions involves a tradeoff between the immediate cost incurred at each particular stage and the anticipated future cost incurred as a consequence of each decision. A routing mechanism that optimizes only over each immediate step without consideration for future incurred costs is labeled a myopic mechanism. [7]

Problems exist for which the optimal mechanism is a myopic one (i.e. the cost over all stages is minimized by minimizing the cost at each stage). There exists another class of problems for which a myopic mechanism is suboptimal, but a lack of information about the number of stages in the problem or of the structure of the problem makes such a mechanism the only option. The on-orbit routing problem considered in this chapter falls into both categories, depending on the configuration of orbits. Each chipsat optimizes over each decision with-

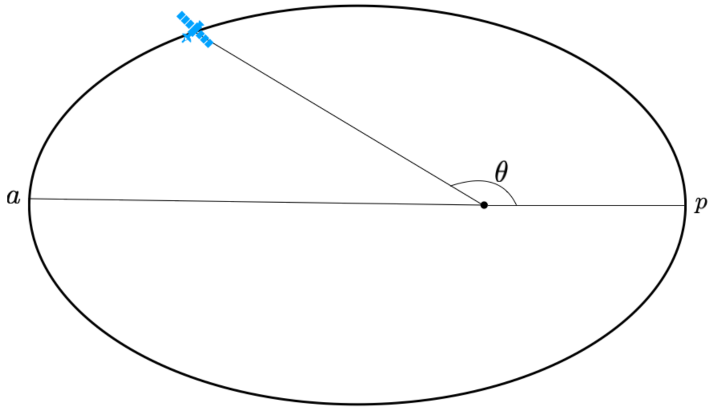


Figure 3.5: Illustration of state variables

out consideration for future cost incurred as a consequence of that decision. It does so because no chipsat has enough system knowledge to estimate future incurred cost. It will be shown that, for some configurations of orbits, this myopic mechanism yields the optimal mechanism (the mechanism that generates a route with the shortest expected time to ground station). For other configurations of orbits this myopic mechanism is suboptimal, but it is the best that can be achieved under the necessary assumptions for grounding this problem in reality, as described in Section 3.2. Solving a problem using the dynamic programming technique requires a state representation and state update equation, a representation for control input, representations for stopping and stage costs, and an optimal value function. With these defined, one can solve for the optimal control mechanism for minimizing cost over a number of stages.

The state of the system is specified by the instantaneous perigee altitude of the chipsat carrying data to be routed, the apogee altitude of the chipsat, the angular position of the chipsat measured from its perigee position, and an identifier for which chipsat is presently carrying the data, as shown in Eqn. 3.1 and Fig. 3.5. All of these quantities can be found directly from the GPS data available to each chipsat. In most dynamic programming problems, the state is indexed by time, and the state update equations evolve each state variable from a timestep k to the next timestep k+1. For this particular problem, the state is indexed by a quantity other than time. Instead, the state is indexed by swept Earth angle, as shown in Fig. 3.6. As explained later in this section, each chipsat maintains an estimate of its own expected time to a ground station, the precise location of which is unknown to any chipsat. As each chipsat sweeps more of the Earth without discovering the ground station, it becomes increasingly confident that it will find it in the near future. In the case of the 2-dimensional problem considered in this chapter, a chipsat can be completely confident that it will discover a ground station as it approaches a swept angle of 2π radians without having yet discovered it. These estimated times to ground station, which incorporate the orbital mechanics associated with each chipsat's apogee, perigee, and true anomaly, form the basis of routing decisions.

Each time the data-carrying chipsat encounters another chipsat, it updates its state as shown in Eqns. 3.4-3.5. It does so by measuring its current apogee

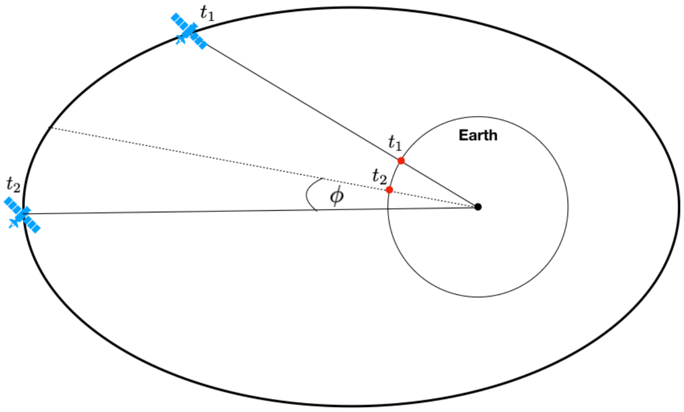


Figure 3.6: Illustration indexing variable, ϕ

altitude, perigee altitude, and angular position from its onboard GPS, and scaling its angular distance traveled by the angular rate of the Earth to determine its updated swept Earth angle, which indexes the state. This update equation is shown in Eqn. 3.5. Note that the chipsats angular position, θ, is a form of true anomaly that may exceed 2π radians. It is a measure of angular distance traveled since routing began. The control input to the system is very simple. At each encounter with an intermediate chipsat, the chipsat carrying data to be routed may take one of two actions. It may either relinquish its data to the encountered chipsat, or it may retain the data for itself, as shown in Eqn. 3.3. It makes this decision on the basis of the stopping cost, terminal cost, and optimal value function.

State representation:

$$x_{\phi_-} = \begin{cases} \gamma & \text{chipsat identifier} \\ p & \text{perigee altitude (km)} \\ a & \text{apogee altitude (km)} \\ \theta & \text{angular position (rad)} \end{cases} \tag{3.1}$$

Initial condition:

$$x_{\phi_0} = \begin{bmatrix} \gamma_0 \\ p_0 \\ a_0 \\ \theta_0 \end{bmatrix} \tag{3.2}$$

Control input:

$$u_{\phi_-} \in \begin{cases} 1 & \text{Relinquish data} \\ 0 & \text{Retain data} \end{cases} \tag{3.3}$$

State update equation:

$$x_{\phi_+} = f(x_{\phi_-}, u_{\phi_-}) = \begin{bmatrix} \gamma_+ \longrightarrow \text{from routing decision} \\ p_+ \longrightarrow \text{from GPS} \\ a_+ \longrightarrow \text{from GPS} \\ \theta_+ \longrightarrow \text{from GPS} \end{bmatrix} \tag{3.4}$$

Index update equation:

$$\phi_+ = (\theta_+ - \theta_-) \frac{T_{Earth}}{T_{Earth} - T_{Node}} \tag{3.5}$$

The stopping cost is the cost incurred if the chipsat carrying data relinquishes that data to the encountered chipsat. This cost is the optimal expected time to a ground station for the chipsat to which the data is relinquished. Note that, with global knowledge, this calculation would include a term that incorporates the probability of encountering another chipsat with a faster expected time to the ground station in the future. Under the assumptions required for making this routing mechanism a practical one, described in Section 3.2, no chipsat has access to the global information required to arrive at these probabilities. As a consequence, the optimal expected time to the ground station (which would include information about probability of future encounters) is approximated by Eqn. 3.8. Eqn. 3.8 gives the expected time to the ground station without consideration of the possibility of future encounters. Section 3.5 shows that this approximation still yields the optimal route for particular configurations of orbits, since it is only the relationship between the stopping cost and the optimal cost to go that is of consequence for decision making, and not the particular values of each, as shown in Eqn. 3.12.

Note also that the stopping cost optimizes only over time, rather than jointly optimizing over energy and time. This is a consequence of the chipsats' architecture. The chipsats do not have any means of propulsion, nor do they have any power storage in the form of batteries. All of the electronics are powered directly from a 300 mW solar cell. The power availability from the solar cell significantly exceeds the power draw from the electronics. Thus, when a chipsat is illuminated, it has a continuous supply of more-than-ample power. It is for this reason that the stopping cost optimizes over time, and not over energy and time. If the chipsats were storing energy and strategically meeting it out, or if they were using an expendable resource for propulsion, then such a joint optimization would be the prudent choice.

From the expression for the stopping cost, it can be seen that the terminal cost (the cost incurred at a swept angle of 2π radians) is the expected time from the initial swept angle, ϕ_0, if no handoff has occurred. If a handoff has occurred, then the terminal cost is 0. The optimal value function is shown in Eqn. 3.11, which again involves an approximation of the optimal expected time to ground station of the same sort used for the stopping cost. This yields the simple thresholding policy shown in Eqn. 3.12. At each encounter, the chipsat carrying data uses its onboard GPS unit to update its state and state index (swept angle). It then shares this state index with the encountered chipsat, and both calculate their expected time to the ground station by integrating their own probability density functions for position, as shown in Eqn. 3.8. The myopic routing mechanism then simply chooses whichever chipsat has the shortest expected time to the ground station, and the process repeats until the ground station is encountered.

The routing mechanism is optimal if and only if the approximations for the expected time to ground station involved in the calculation of the optimal time to go, $V_{\phi_+}(x_{\phi_+})$, and the stopping cost, c_s, are such that the relationship between these approximations always yields a correct decision. Eqn. 3.12 makes it clear that it is the relationship between $V_{\phi_+}(x_{\phi_+})$ and c_s that is of consequence for decision making, and not the values themselves. For some configurations of orbits, this is the case. For others, it is not.

Stopping cost:

$$\begin{aligned} c_s = {} & \text{optimal expected time to destination} \\ & \text{for encountered chipsat} \end{aligned} \tag{3.6}$$

Stage cost:

$$c_{\phi_-}(x_{\phi_-}, u_{\phi_-}) = \begin{cases} 0, & u_{\phi_-} = 0 \\ c_s, & u_{\phi_-} = 1 \end{cases} \tag{3.7}$$

Optimal expected time to ground station (see appendices):

$$\begin{aligned} E_{\phi_-}[t](x_{\phi_-}) \approx \frac{1}{2\pi - \phi_-} \int_{\phi_-}^{2\pi} \Bigg[& \frac{T_{node} T_{Earth}}{2\pi \left(T_{Earth} - T_{node}\right)} \\ \int_{\phi_-}^{y} & \frac{(1-e)^{\frac{3}{2}}}{\left(\frac{1}{1+e}\right)^{\frac{3}{2}} \left(e \cos\left((\theta_- + \phi) \frac{T_{Earth}}{T_{Earth} - T_{node}}\right) + 1\right)^2} d\phi \Bigg] dy \end{aligned} \tag{3.8}$$

Terminal cost:

$$c_{\phi=2\pi} = \begin{cases} E_{\phi_0}[t](x_{\phi_0}) & \gamma_{\phi=2\pi} = \gamma_0 \\ 0 & \gamma_{\phi=2\pi} \neq \gamma_0 \end{cases} \tag{3.9}$$

Terminal value function:

$$V^*_{\phi=2\pi}(x_{\phi=2\pi}) = \begin{cases} E_{\phi_0}[t](x_{\phi_0}) & \gamma_{\phi=2\pi} = \gamma_0 \\ 0 & \gamma_{\phi=2\pi} \neq \gamma_0 \end{cases} \tag{3.10}$$

Optimal value function:

$$\begin{aligned} V^*_{\phi_-} &= \min_{u \in \{0,1\}} E\left[c_{\phi_-}(x_{\phi_-}, u_{\phi_-} \in \{0,1\} + V^*_{\phi_+}(x_{\phi_+})\right] \\ &= \min\left[c_s, V^*_{\phi_+}(x_{\phi_+})\right] \\ &\approx \min\left[c_s, E_{\phi_+}[t](x_{\phi_+})\right] \end{aligned} \tag{3.11}$$

Optimal routing mechanism:

$$g^*_{\phi_-} = \begin{cases} 1 & c_s < V_{\phi_+}(x_{\phi_+}) \\ 0 & c_s \geq V_{\phi_+}(x_{\phi_+}) \end{cases} \tag{3.12}$$

3.5 Performance

The relationship between the optimal time to go, $V_{\phi_+}(x_{\phi_+})$, and the stopping cost, c_s, will be such that the relationship between these approximations will always yield a correct decision, as shown in Eqn. 3.12, if and only if the following conditions are met:

1. After a routing decision among two chipsats, it is impossible that the chipsat that had a longer expected time to the ground station (and therefore relinquished the data) at the time of the routing decision will both a) later attain a shorter expected time to ground station than the other chipsat and b) overtake the other chipsat at a distance that exceeds the node-to-node communication distance before the entire Earth has been swept.

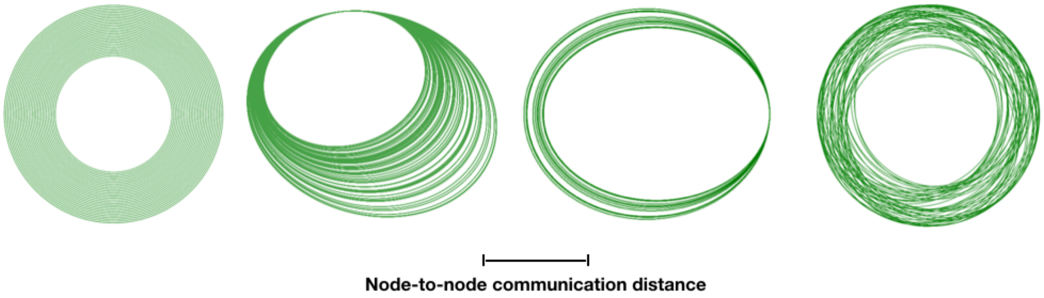


Figure 3.7: Examples of fully connected orbit configurations

2. After a routing decision has been made, it is impossible that the chipsat which relinquished the data will encounter another chipsat that both a) has a shorter expected time to ground station than the first chipsat to which the data was surrendered and b) will not come into communicable range with that chipsat before the entire Earth has been swept.

If these conditions are met, then the routing mechanism will choose the path which minimizes the expected time to the ground station. These conditions are examined for each of a series of orbital configurations.

3.5.1 Performance on Fully Connected Configurations

We consider configurations of orbits for which the maximum altitude separation for all nodes is within the node-to-node communication distance, as shown in Fig. 3.7. For this particular case, any two chipsats that are at the same angular position are capable of communicating. For such configurations of orbits, it is impossible for any chipsat to overtake any other chipsat without passing within a communicable distance. Any chipsat that overtakes another chipsat will be able to communicate with the chipsat that it is overtaking. Thus, there is never cost incurred by relinquishing data to a chipsat with a faster estimated time to ground station (even if the origin chipsat encounters an even-faster chipsat after handoff has occurred, or if the original chipsat later attains a faster expected time to ground station). The optimality conditions therefore hold and the derived mechanism chooses the path which minimizes the expected time to the ground station for these configurations of orbits. Section 3.5.2 shows that, in the special case that all orbits are circular, the mechanism not only chooses the path which minimizes the expected time to the ground station, it chooses the optimal path which minimizes absolute time to the ground station.

3.5.2 Performance on Nested, Circular Configurations

A second configuration of interest is one composed of concentric circular orbits with altitude separations that may or may not prevent complete connectedness,

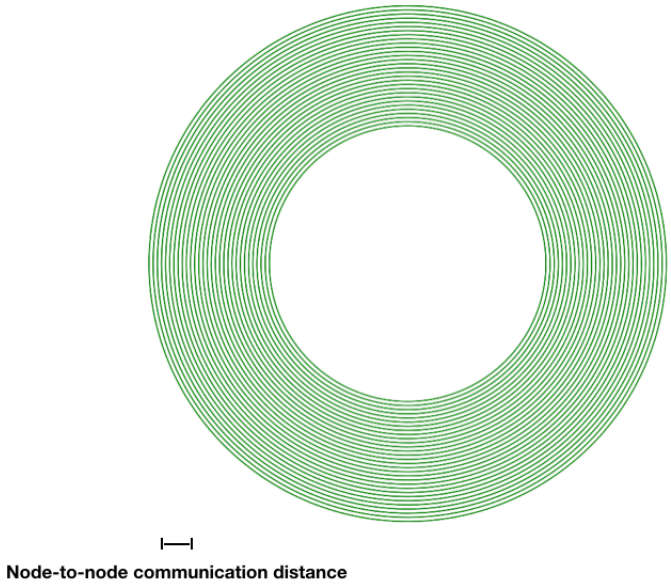


Figure 3.8: Nested, unconnected configuration of circular orbits

as shown in Fig. 3.8. Chipsats on circular orbits travel at a constant velocity and have zero eccentricity. The equation for the expected time to ground station, eqn. 3.8, reduces to the form shown in eqn. 3.13.

$$E_{\phi_-}[t](x_{\phi_-}) = \frac{1}{2\pi - \phi_-} \int_{\phi_-}^{2\pi} \left[\frac{T_{node} T_{Earth}}{2\pi \left(T_{Earth} - T_{node}\right)} \int_{\phi_-}^{y} d\phi \right] dy \tag{3.13}$$

The consequence is that chipsats on lower altitude orbits will always have a shorter expected time to ground station, as shown in Fig. 3.9. Chipsats on lower orbits are always traveling more quickly than chipsats on higher orbits and, as a result, always have a shorter period, T_{node}. All optimality conditions therefore hold for this configuration of orbits. The myopic routing mechanism always chooses to pass data down in altitude, which always results in not just the optimal expected path to the ground station, but the time-optimal path. For both connected and disconnected configurations of circular orbits, the derived myopic routing mechanism chooses the time-optimal path to the ground station.

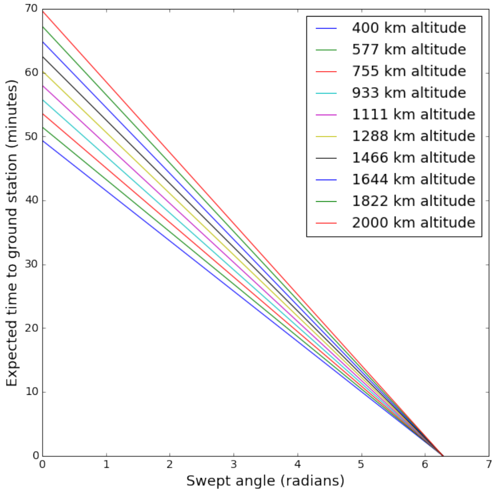


Figure 3.9: Expected time to ground station for circular orbits of differing altitude

Optimizing over each stage leads to a time-optimal path over all stages.

3.5.3 Performance on Nested, Unconnected Elliptical Configurations

The configuration of particular interest for practical applications is one composed of nested elliptical orbits that are separated by altitudes that exceed the node-to-node communication distance, as shown in Fig. 3.10. Each chipsat may communicate with other chipsats that occupy orbits of similar altitudes, but not those that occupy orbits of significantly higher or lower altitude. This configuration is of particular practical interest because it is the one that, to good approximation, chipsats deployed from a common mothership will achieve.

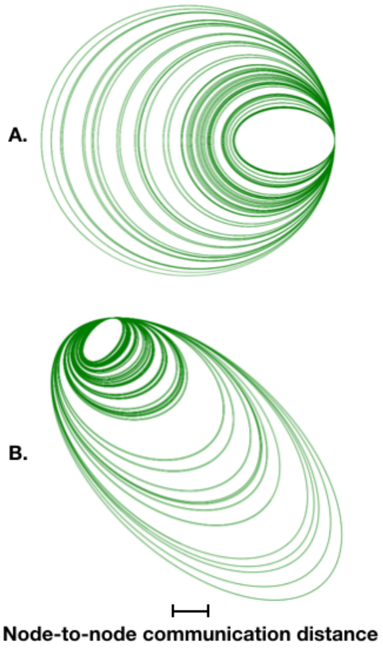


Figure 3.10: Nested, unconnected configurations of elliptical orbits

Because all chipsats are deployed from a common mothership, the collection of all attainable orbits by each deployed chipsat can be found by adding some amount of along-track velocity (in the forward or reverse direction) that is in the range of possible deployment velocities from the mothership. The important property of this collection of attainable orbits is that they do not cross over one another. All are nested inside of one another, intersecting at (at most) one location. If the mothership deploys the chips at its own perigee position, then all chips will have varying apogee altitudes (all at the same angular position) and an instantaneously identical perigee altitude and position, as shown in Fig. 3.10A. If the mothership deploys the chips at its own apogee position, then all chips will have varying perigee altitudes (all at the same angular position) and an instantaneously identical apogee altitude and position. If the mothership deploys the chips between apogee and perigee, then all nodes will land on a family of orbits in which each constituent orbit has either higher or lower apogee and perigee than all other orbits, as shown in Fig. 3.10B. None of these, however, crossover one another. All orbits are strictly losing energy and, as a consequence, their perigee and apogee altitudes are continuously decreasing. Orbits with lower average altitudes will lose energy more quickly than orbits with higher average altitudes because of the increased amount of atmospheric drag that they experience. The result is that the interior orbits in the initially nested configuration shrink away from higher altitude orbits more quickly than the higher altitude orbits approach them. The nested configuration is therefore maintained. A thought experiment can be used to show that the myopic mechanism is suboptimal (it does not necessarily choose the path which minimizes the expected time to the ground station) on these configurations of orbits.

Consider chipsats A, B, and C that are orbiting on nested, elliptical orbits as shown in Fig. 3.10A. Chipsat A carries data to be routed to a ground station. It is overtaken by chipsat B, which has a shorter expected time to the ground station than chipsat A and, therefore, receives the data from A. Later, chipsat A is overtaken by chipsat C, which has a shorter expected time to the ground station than A or B. If it is possible for chipsat C to overtake chipsat B at a distance greater than the distance at which it overtook chipsat A, then the routing mechanism is suboptimal because a shorter expected time to ground station could have been achieved if A had waited to handoff to C. Optimality condition 2 would not hold. This is only possible if the rate at which the perpendicular separation between the orbits of B and C increases is greater than the rate at which the angular separation between B and C decreases.

A simple argument proves that this is the case for parts of the orbits of A, B, and C. Between perigee and apogee, the distance between orbits in the nested configuration shown in Fig. 3.10A monotonically increases. Between apogee and perigee, the distance between orbits monotonically decreases. Thus, the rate of change of the distance between orbits in the nested configuration is zero at apogee and zero at perigee, with a maximum rate of change at a location somewhere between apogee and perigee. The rate of angular separation behaves differently. For orbits in the configuration under consideration, the velocity of chipsats with higher apogee altitudes is greater at perigee than the speed of chipsats with lower apogee altitudes. At apogee, however, chipsats with lower apogee altitudes have greater velocity than those with higher apogee altitudes (see Fig. 3.11). Therefore, the rate of change of angular separation between chipsats is negative at perigee and positive at apogee. There must then be a location, somewhere between perigee and apogee, for which the rate of change of angular separation is zero.

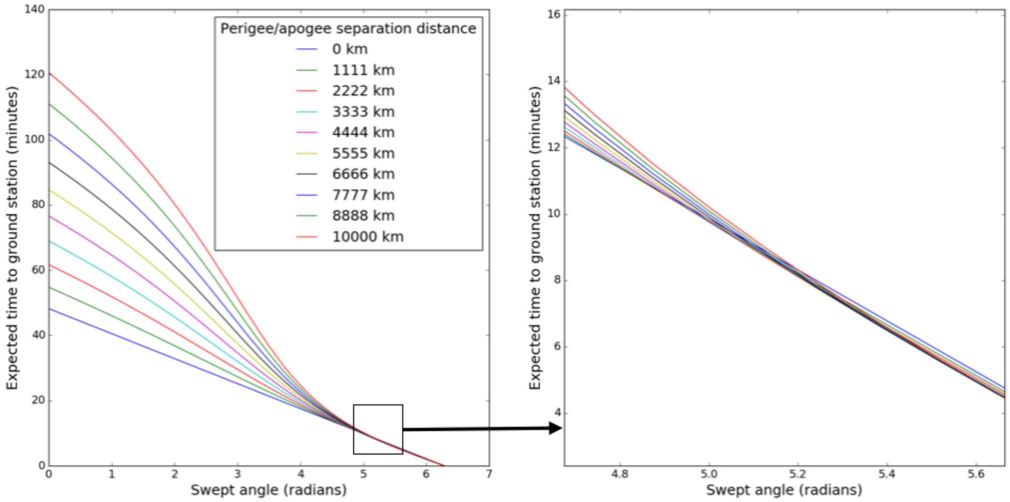


Figure 3.11: Expected time to ground station for nested elliptical orbits, as in Fig. 3.10A.

This proves that there must be a range of locations for which the rate of change of separation between orbits exceeds the rate of change of angular separation between chipsats on those orbits. It is possible, therefore, for an incorrect routing decision to be made. A chipsat may handoff to another chipsat before encountering a third that has a shorter expected time to ground station. It has been shown that it is not always the case that this third chipsat will pass within communicable distance of the first, and therefore the routing mechanism does not always yield an optimal expected route. Optimality condition 2 does not hold for nested configurations of elliptical orbits like that shown in Fig. 3.10A. Through a nearly identical argument, it can be shown optimality condition 2 does not hold for configurations like that shown in Fig. 3.10B either.

Chipsat deployment between apogee and perigee leads to the skewed configuration of orbits shown in Fig. 3.10B, in which all orbits have higher or lower apogee and perigee than all other orbits. The consequence is that the rate of change of separation between orbits is increased for one half of the network and decreased on the other half of the network. Using precisely the same reasoning as was employed for the previous configuration, it can be shown that there must exist a range of locations in this network for which the rate of change of separation between orbits exceeds the rate of change of angular separation between chipsats on those orbits. Therefore, for this configuration also, it is possible for incorrect routing decisions to be made and the myopic mechanism is sub optimal. It may choose a path which does not minimize the expected time to the ground station. Sub optimal, however, does not necessarily mean not worth doing. The expected time to the ground station is still reduced by performing the handoffs as prescribed by the routing mechanism. Sub-optimality only means that a better handoff could have been made.

3.5.4 Performance on Stochastic, Unconnected Configurations

For a stochastic collection of unconnected orbits, like that shown in Fig. 3.12, no guarantees whatsoever may be made for optimality conditions 1 or 2. For this configuration, therefore, the myopic routing mechanism is suboptimal. As in the nested, unconnected configuration, the expected time to the ground station is still reduced by performing the handoffs as prescribed by the routing mechanism. Sub-optimality only means that a better handoff could have been made.

3.6 Practical Considerations

Implementation of the system described in this chapter will lead to a number of off-nominal situations that would need to be accommodated. These include situations in which a collection of chipsats is replenished from a secondary mothership, situations in which a packet misses a communication opportunity with the ground station, and non-catastrophic GPS failure modes. Each is considered in this section.

The only assumption on the collection of chipsats is that they all occupy the same orbital plane. As long as subsequent motherships all occupy the same orbital plane (which is not difficult, practically, to achieve), then the assumptions are not affected. A replenishment may, however, affect the configuration of the collection of orbits and consequently the optimality of the routing mechanism. If, for example, the original collection of chipsats occupied a fully connected nested configuration of orbits, then the routing mechanism would be routing optimally over that collection chipsats (Section 3.5.1). A replenishment may

Table 3.2: Summary of optimality for each orbit configuration

Orbit Configuration	Routing Mechanism Performance
Nested, circular, connected	Chooses optimal path
Nested, circular, disconnected	Chooses optimal path
Fully connected, non-circular	Chooses optimal expected path
Nested, elliptical, disconnected	May choose suboptimal expected path
Stochastic, disconnected	May choose suboptimal expected path

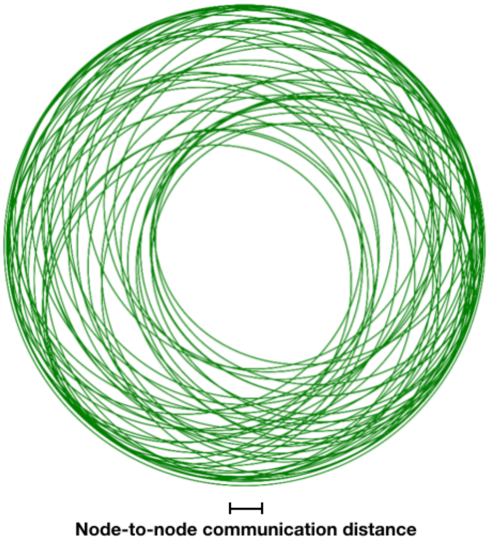


Figure 3.12: Stochastic, unconnected configurations of orbits

change the configuration from nested and fully connected to nested and unconnected, or stochastic. The routing mechanism still works over such collections of orbits but, as explained in this chapter, it is not optimal over such configurations.

It is also possible that a chipsat will fail to find a ground station during a single pass. The action taken in this case would depend on the mission and the importance of the packet being routed. For missions in which other chipsats can be expected to be routing similar information (a solar activity monitoring mission, for example), then the chipsat might retire the data after it has swept its entire search space (θ, Section 3.4). Alternatively, for missions requiring a greater guarantee on each packets delivery, the chipsat may make the conservative assumption that a technical problem has prevented it in particular from communicating, and it will surrender its packet to a neighboring chipsat.

The ultimate practical consideration is whether latency benefit from the routing mechanism is worth the implementation burden, as opposed to having each chipsat wait for direct downlink to a ground station. Chipsats enable swarms of hundreds of thousands of spacecraft. In swarms of such number, the connectivity of the network will remain intact until the chipsats begin to deorbit. The neighboring nodes will change but, if enough are launched, a path will remain from each node in the network to each of the other nodes in the network through intermediate nodes. Consequently, the speed with which a packet can be routed from any origin node in this network to the destination is limited only by the time that it takes to decide on a handoff, and the light-travel time of the packet. In this limiting case of very large constellations, the packet transmission time savings will be radically reduced by using the strategy described in the chapter as opposed to the non-strategy of waiting for direct downlink.

3.7 Derivations

This section derives the probability density function for the true anomaly of a spacecraft on an elliptical, Keplerian orbit, as used in Eqn. 3.8.

3.7.1 Swept Area from Angular Position

Consider an elliptical orbit with semimajor axis a and semiminor axis b, as shown in Fig. 3.13. Earth sits at focal point F, with perigee at A. As the spacecraft traverses the orbit, it sweeps out the area AFP. The position of the spacecraft is specified by its distance from the Earth (ρ) and the angle from perigee (θ). Some of the geometric relationships among the above quantities are given

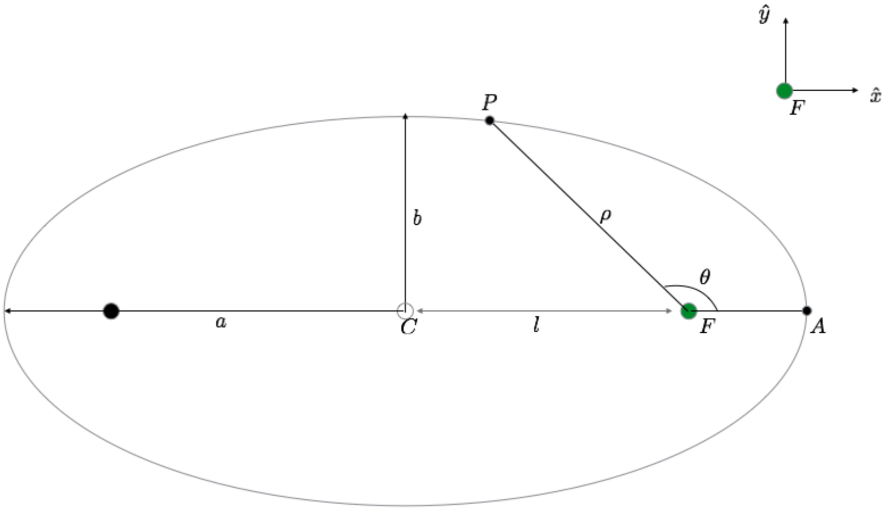


Figure 3.13: Swept area geometry, origin at focus

by Eqns. 3.14-3.16.

$$e = \text{eccentricity} = \sqrt{1 - \frac{b^2}{a^2}} \tag{3.14}$$

$$l = \text{linear eccentricity} = ae \tag{3.15}$$

$$\rho(\theta) = \frac{a(1 - e^2)}{1 + e\cos\theta} \tag{3.16}$$

The location of the spacecraft (in Cartesian coordinates, with the origin on the Earth) can be parametrized as shown in Eqns. 3.17-3.18.

$$ {}^{F}\mathbf{P} = \begin{bmatrix} \rho\cos\theta \\ \rho\sin\theta \end{bmatrix} \tag{3.17} $$

$$ = \begin{bmatrix} \frac{a(1-e^2)}{1+e\cos\theta}\cos\theta \\ \frac{a(1-e^2)}{1+e\cos\theta}\sin\theta \end{bmatrix} \tag{3.18} $$

Consider scaling the y-coordinate such that the trajectory is a circle of radius a (the semimajor axis), is shown in Fig. 3.14. This can be done by scaling the y-axis by $\frac{a}{b} = \frac{1}{\sqrt{1-e^2}}$. The point P gets mapped to Q, and the new position can be parametrized as shown in Eqns. 3.19 and 3.20. Moving the origin to the center of the circle, as shown in Fig. 3.15, moves the position of Q to that given by Eqns. 3.21 and 3.22.

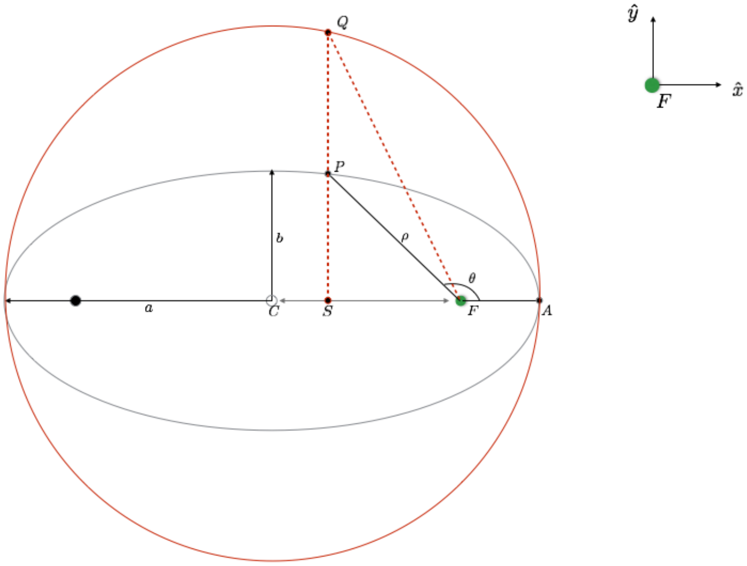


Figure 3.14: Swept area geometry, scaled to circle

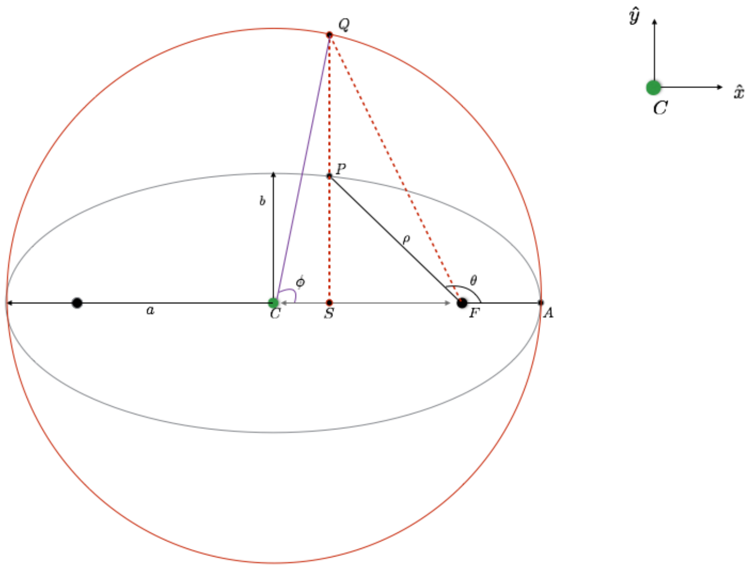


Figure 3.15: Swept area geometry, scaled to circle, origin at center

$$
{}^{F}\mathbf{Q} = \begin{bmatrix} \rho \cos\theta \\ \frac{1}{\sqrt{1-e^2}} \rho \sin\theta \end{bmatrix} \tag{3.19}
$$

$$
= \frac{a}{1 + e\cos\theta} \begin{bmatrix} (1-e^2)\cos\theta \\ \sqrt{(1-e^2)}\sin\theta \end{bmatrix} \tag{3.20}
$$

$$
{}^{C}\mathbf{Q} = \begin{bmatrix} \rho \cos\theta + ae \\ \frac{1}{\sqrt{1-e^2}} \rho \sin\theta \end{bmatrix} \tag{3.21}
$$

$$
= \frac{a}{1 + e\cos\theta} \begin{bmatrix} (1-e^2)\cos\theta + e(1 + e\cos\theta) \\ \sqrt{(1-e^2)}\sin\theta \end{bmatrix} \tag{3.22}
$$

The angle ϕ is the eccentric anomaly. The geometry of Fig. 3.15 yields Eqn. 3.23, which can be simplified to Eqn. 3.14 using a Wiererstrauss substitution. Solving Eqn. 3.24 for ϕ yields Eqns. 3.25-3.26. Solving Eqn. 3.24 for θ yield Eqn. 3.27.

$$\tan\phi = \frac{\sqrt{1-e^2}\sin\theta}{(1-e^2)\cos\theta + e + e^2\cos\theta} \tag{3.23}$$

$$\tan\frac{\phi}{2} = \sqrt{\frac{1-e}{1+e}}\tan\frac{\theta}{2} \tag{3.24}$$

$$\begin{aligned} \phi &= \operatorname{atan2}\left(\sqrt{1-e^2}\sin\theta,\quad (1-e^2)\cos\theta + e + e^2\cos\theta\right) && (3.25)\\ &= 2\tan^{-1}\left[\sqrt{\frac{1-e}{1+e}}\tan\frac{\theta}{2}\right] && (3.26) \end{aligned}$$

$$\theta = 2\tan^{-1}\left[\sqrt{\frac{1+e}{1-e}}\tan\frac{\phi}{2}\right] \tag{3.27}$$

With this angle ϕ, one can calculate the area of sector ACQ using Eqn. 3.28. This sector includes the region of interest.

$$A_{ACQ} = \frac{1}{2}a^2\phi \tag{3.28}$$

To obtain the area of the region of interest, subtract and scale Eqn. 3.28. To begin, subtract off the area of triangle FCQ, leaving the region AFQ remaining, as shown in Eqns. 3.29-3.31.

$$\begin{aligned} A_{AFQ} &= A_{ACQ} - A_{FCQ} && (3.29)\\ &= \frac{1}{2}a^2\phi - \frac{1}{2}a^2 e\sin\phi && (3.30)\\ &= \frac{1}{2}a^2\left[\phi - e\sin\phi\right] && (3.31) \end{aligned}$$

This is still the area for a circle. To get back to the area for the ellipse, undo the initial scaling by multiplying by $\frac{b}{a}$ (since scaling in the y-direction scales the area by the same factor). Doing so yields Eqn. 3.32.

$$A_{AFP} = \frac{1}{2}ab\left[\phi - e\sin\phi\right] \tag{3.32}$$

3.7.2 Probability Density from Swept Area

Because the spacecraft sweeps equal areas in equal times, the liklihood of finding the spacecraft in this angular region is given by Eqns. 3.33-3.34. This leads directly to the probability distribution function in ϕ, shown in Eqn. 3.35. Substituting the expression for θ yields the probability distribution in θ, shown in Eqn. 3.36.

$$\begin{aligned} P(AFP) &= \frac{A_{AFP}}{A_{total}} \qquad (3.33)\\ &= \frac{\frac{1}{2}ab\left[\phi - e\sin\phi\right]}{\pi ab} \qquad (3.34)\end{aligned}$$

$$P(\phi) = \frac{\phi - e\sin\phi}{2\pi} \tag{3.35}$$

$$P(\theta) = \frac{2\tan^{-1}\left(\sqrt{\frac{1-e}{e+1}}\tan\left(\frac{\theta}{2}\right)\right) - e\sin\left(2\tan^{-1}\left(\sqrt{\frac{1-e}{e+1}}\tan\left(\frac{\theta}{2}\right)\right)\right)}{2\pi} \tag{3.36}$$

The probability density function is obtained by taking the derivative of the distribution function with respect to ϕ (or θ in the case of $P(\theta)$). The result is given by Eqns. 3.37-3.38. Eqn. 3.38 is the probability density function for the true anomaly of a spacecraft on an elliptical, Keplerian orbit.

$$p(\phi) = \frac{\partial P(\phi)}{\partial \phi} = \frac{1 - e\cos(\phi)}{2\pi} \tag{3.37}$$

$$p(\theta) = \frac{\partial P(\theta)}{\partial \theta} = \frac{(1-e)^{3/2}}{2\pi\left(\frac{1}{e+1}\right)^{3/2}(e\cos(\theta)+1)^2} \tag{3.38}$$

CHAPTER 4

HIGH-RISK PLANETARY IMPACT MISSIONS

4.1 Motivation

As with R-selected species, one of the key advantages to employing high quantity for low risk rather than high cost for low risk is that mission success does not depend on any constituent member of the group. As a consequence, Monarchs can go places and take actions that would be prohibitively dangerous for large, conventional spacecraft. And individual Monarchs are disposable. So, in addition to the favorable impact mechanics associated with their low size and mass, Monarchs are extraordinarily well suited for high-risk planetary science and atmospheric reentry missions. Monarchs can be used to descend to the surfaces and through atmospheres of celestial bodies, such as Venus, Titan, or Europa. Their small size makes entry, descent, and landing (EDL) methods significantly different for Monarchs than for conventional spacecraft. Importantly, one does not need to guarantee survival of every Monarch throughout EDL, only to guarantee the statistics of survival. This mindset is entirely new in the field of planetary exploration.

The Monarchs' size makes them better equipped for surviving impacts, turbulence, and other shock-related effects than large conventional spacecraft. Scaling benefits the robustness of small spacecraft, since mass scales with approximately the cube of length, and strength with approximately the square of length. Smaller things exhibit higher natural structural frequencies and approach crystal-lattice stiffness. They are therefore stronger and can take a greater beating. This fact is also apparent in nature, where insects have propor-

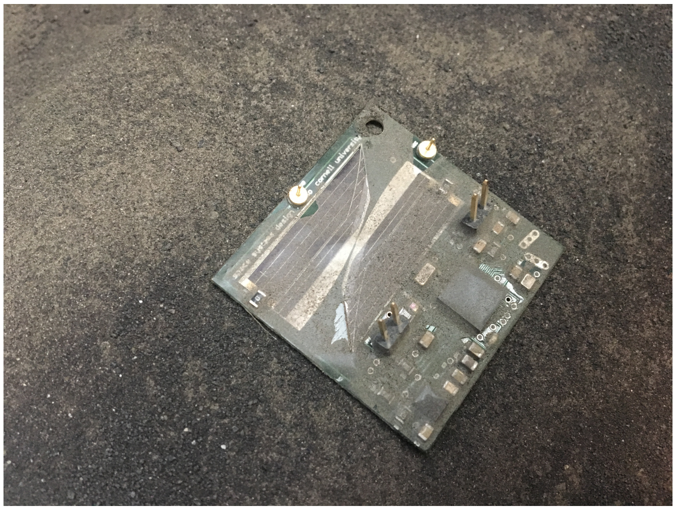


Figure 4.1: Impact test article on bed of lunar regolith simulant after exposure to 27,000 g's of acceleration to simulate impact with the lunar surface.

tionally greater strength than larger creatures and are capable of withstanding shocks that larger animals could not survive [43, 40, 32, 28]. Monarchs are the insects of spacecraft. Their resilience has significant implications for the entry, descent, and landing technology required to give Monarchs a chance of survival. There is evidence, in fact, that no such technology is required at all and that Monarchs may survive impacts with no additional protection.

4.2 Impact Durability Study

An impact durability study in 2017 exposed 12 Monarch precursors (printed circuit board test articles) to 5000-27,000 g's of acceleration normal to the board surface via an elastically loaded drop table. The drop table is described at length in [29, 11]. Each board carried the same inertial measurement unit (IMU) as the Monarch, the internal mechanics of which make it the most shock-sensitive component on the spacecraft. Lunar regolith simulant was placed underneath each test article in order to simulate impact with the lunar surface, as shown in Fig. 4.1. Prior to impact, each board was placed in a static testbed and a batch of measurements was gathered from the accelerometer, magnetometer, and gy-

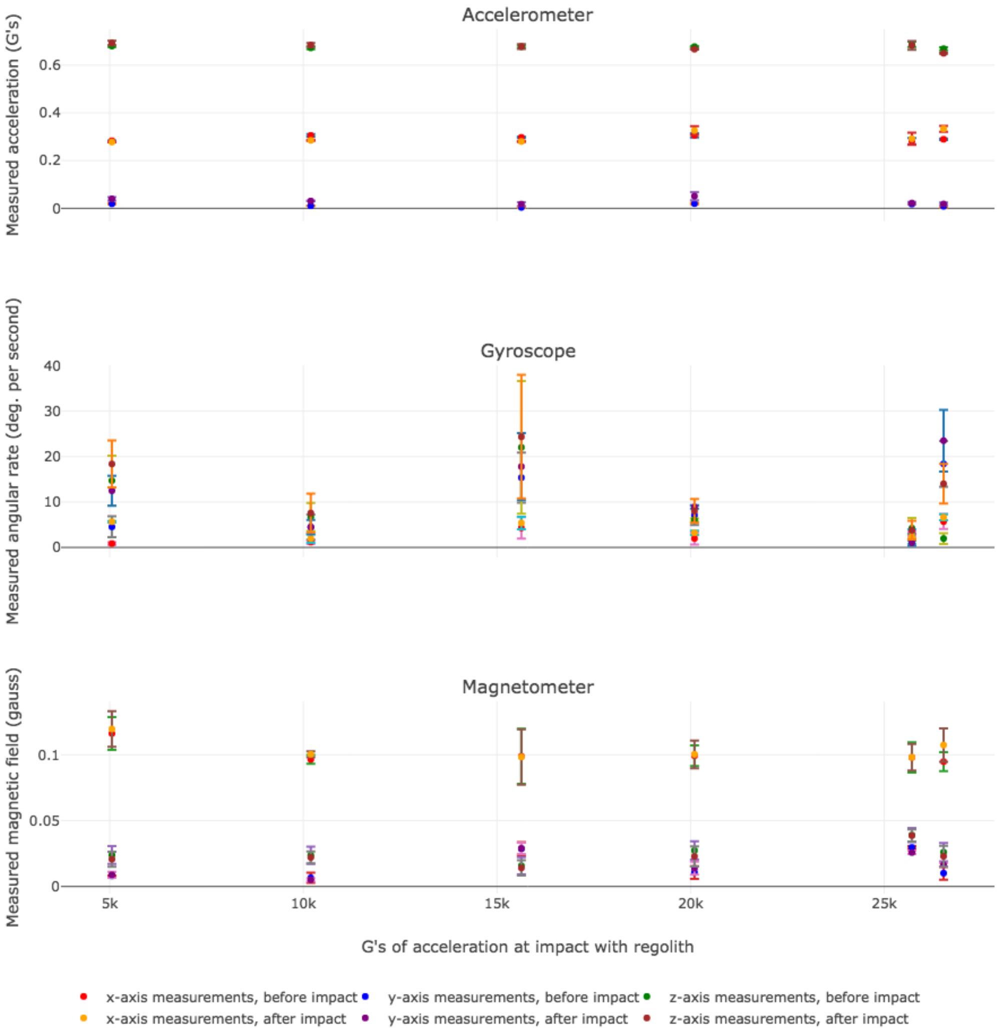


Figure 4.2: IMU measurements before and after impact with lunar regolith simulant, showing that the MEM's sensor survives and continues to operate to within the specifications of the datasheet. [30]

roscope. This step verified that the IMU on each board was operating to within the specifications of the datasheet, and characterized each sensor before impact. After impact, each test article was placed in the same testbed and measurements were gathered again from the same set of sensors in order to characterize degradation. As shown in Fig. 4.2, each IMU continued to operate to within manufacturer specifications for zero-g, zero-Gauss, and zero-rate levels after impact with the lunar regolith simulant [30]. This empirical assessment by no means guarantees that every Monarch would survive impact with a celestial body, but it suggests that they have a chance of surviving. If some number k Monarchs are required for mission success and one deploys $N > k$, then up to $\frac{N-k}{N}$ percent may fail on impact before the mission itself becomes unsuccessful. Mission assurance for Monarchs is statistical, and mission assurance equations can be derived from binomial distributions.

4.3 Statistical Mission Assurance for Impact Missions

Suppose that N chipsats are deployed to the surface of a celestial body, each with a probability p_1 of surviving impact. The probability of having any $k \leq N$ survive that impact is given by Eqn. 1. Put alternatively, this expression yields the probability of $k \leq N$ chipsats surviving 0 days on the surface. Each chipsat that survives impact then faces the threats associated with existing on the surface, including radiation. If one lets the probability of surviving each day be p_2 and makes the simplifying assumptions that this probability does not change with time, and that failures among chipsats are not correlated, then eqn. 2 gives the probability that $j \leq k$ chipsats survive for M days. These expressions can be used to find the probability of mission success.

$$p(k \leq N \text{ surviving impact})\Big|p_1, N = \frac{N!}{k!\,(N-k)!} p_1^k \,(1-p_1)^{N-k} \tag{4.1}$$

$$p(j \leq k \text{ surviving } M \text{ days on surface})\Big|p_2, M, k = \frac{k!}{j!\,(k-j)!} \left(p_2^M\right)^j \left(1 - p_2^M\right)^{k-j} \tag{4.2}$$

For a mission like the one under consideration, mission success is defined as at least some specified number $j \leq N$ of chipsats remaining alive on the surface for a specified number of days, M. Eqn. 3 yields the probability of success provided the number of chipsats deployed to the surface (N), the number of days associated with the mission success criterion (M), the number of remaining chipsats associated with the success criterion (j), the probability of any individual chipsat surviving impact (p_1), and the probability of any individual chipsat surviving each day on the celestial body (p_2).

$$\begin{gathered} p(j \geq (k \leq N) \text{ surviving impact and } M \text{ days on surface})\Big|N, M, p_1, p_2 = \\ \sum_{k=j}^{N} \left[\frac{N!}{k!\,(N-k)!} p_1^k \,(1-p_1)^{N-k} \cdot \sum_{i=j}^{k} \frac{k!}{i!\,(k-i)!} \left(p_2^M\right)^i \left(1-p_2^M\right)^{k-i} \right] \end{gathered} \tag{4.3}$$

Eqn. 3 represents a general model for evaluating the likelihood of success for any of these high-risk planetary missions. The variables within this equation must be populated with values specific to the particular mission being performed. The values for p_1 and p_2 will vary substantially from one celestial body to the next and must be determined via testing. The values for j, M, and N will depend on mission requirements. Fig. 4.3 shows a heatmap for the probability of mission success for a range of impact survival probabilities and daily survival probabilities. This heatmap is generated for the particular case where success is

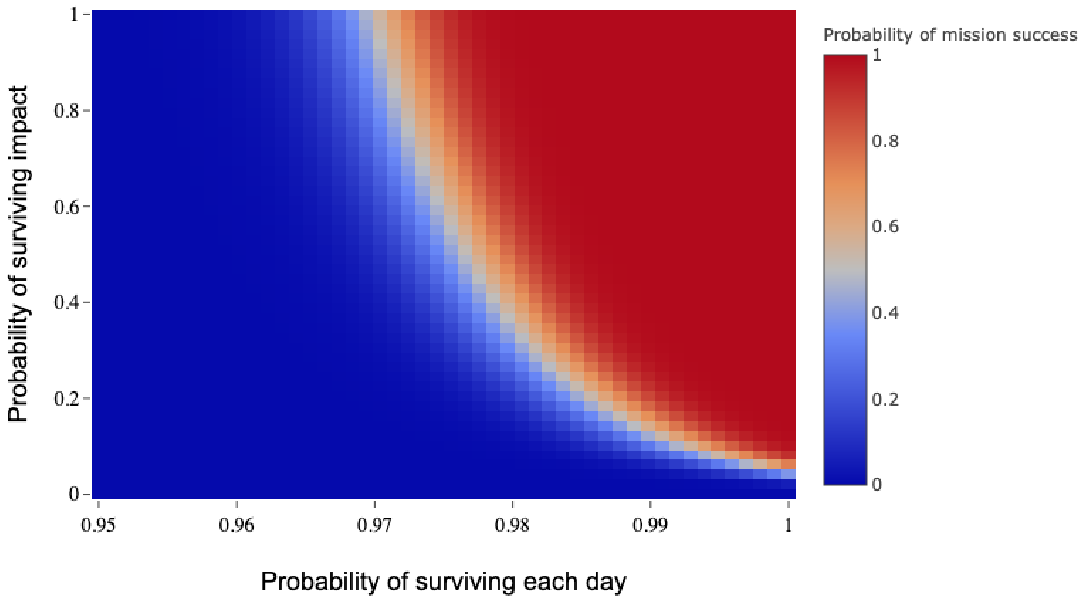


Figure 4.3: The probability of mission success, defined as 5 of 100 chipsats surviving on the the surface of a celestial body for 100 days, for a range of impact survival probabilities and daily survival probabilities.

defined as at least 5 of 100 chipsats surviving for 100 days on the surface. This paradigm in mission assurance places value on the confidence bounds, achieved by the quantity rather than quality of individual spacecraft. The nature of the data that Monarchs can gather once on the surface is best illustrated through a case study.

4.4 Case-Study Planetary Science Mission

A case-study planetary science mission was performed in order to gather a representative dataset. 20 Monarchs were deployed to the surface of Earth for 24 hours, during which time they remotely reported in-situ data from their payload sensor suite, which included a temperature and humidity sensor. Each also reported its location, as measured by its onboard GPS. The locations of the deployed Monarchs and the data that they gathered is shown in Fig. 4.4. For this case study, the planet in question happens to be Earth and the sensory payload happens to include temperature and humidity sensors, but there is nothing special, from a technical perspective, about that particular celestial body or sensory payload. For other celestial bodies, the payload may include a different suite of sensors.

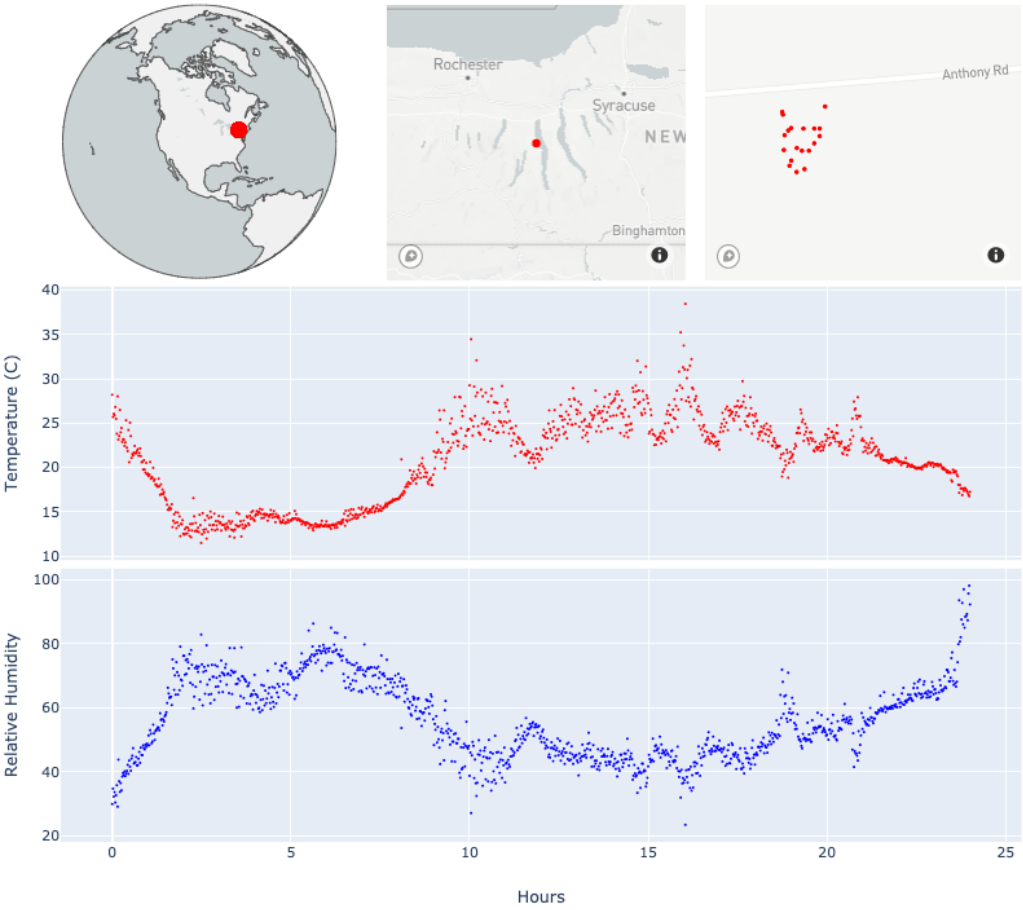


Figure 4.4: Dataset from a 24-hour case-study planetary science mission conducted on Earth.

CHAPTER 5
ANTICIPATING FUTURE CAPABILITIES: OPTICAL TRAJECTORY RECOVERY

5.1 Monarchs with cameras lost in space and time

As surface-mounted sensors and processors continue to improve, one can expect for the capabilities of the Monarch to continue to expand. Cameras are developing particularly rapidly, thanks largely to the cell phone industry. One can expect camera-equipped chipsats in the coming years as a consequence of this development. The addition of a camera will not only enable different sorts of scientific missions, it will also enable optical navigation in deep space. This chapter anticipates a Monarch equipped with a camera and offers a method for solving a problem that will likely be a common one for chipsats in deep space: being lost. This chapter considers a chipsat adrift in cislunar space with no knowledge of its trajectory, its attitude, or of the time. It presents a method by which such a chipsat may use onboard cameras, an onboard relative clock, and an onboard ephemeris table to autonomously recover the set of all trajectories which agree with camera observations. It does so in two steps. It first uses a batch of measurements to coarsely reduce the search space of possible positions and times. For each position/time pair, it then instantiates a particle filter to recover velocities and improve estimates of associated positions and times. The result is a family of trajectories that all produce measurements which agree with those gathered by the onboard cameras.

5.2 Background

Various aspects of spacecraft optical navigation have been studied by a number of researchers. Liounis, Daniel, and Christian consider a manned spacecraft in the Earth-Moon system that must autonomously navigate back to Earth using optical observations of the Earth, Moon, and stars [21]. Their algorithm involves an image filter on features of the celestial bodies that feeds an Extended Kalman Filter (EKF), which converges to the spacecraft trajectory. The filter converges to within 66 meters of true spacecraft position and 1.5 cm/sec of true velocity for a 100 km circular lunar orbit. For the situation treated in this chapter, the lack of time knowledge precludes the use of a similar EKF, which requires a known dynamics model. Furthermore, this analysis assumes cheap, COTS cameras that lack the resolution and contrast required to disambiguate any features on any celestial body, or to gather any information at all about stars other than the Sun. In [19], Lightsey and Christian develop an image-processing algorithm for an onboard optical system. The algorithm calculates the apparent centroid and diameter of a celestial body and calculates the relative angle between a body horizon and a reference star. The present study assumes that a similar sort of image processing procedure has already taken place and uses this a priori information to determine spacecraft trajectory and time.

A similar problem to that treated in this chapter was addressed for Deep Space 1, the first interplanetary spacecraft of NASA's New Millennium program. The spacecraft used onboard cameras to sight up to 12 asteroids at a time and used those sightings in a least-squares filter to estimate the spacecraft orbital parameters. The ephemerides for each body are assumed to be known a priori, and errors in the ephemerides are minimized by combining information from all observed asteroids. Monte Carlo simulation shows convergence to true position to within approximately 95 km, and convergence to true velocity to within approximately 0.2 m/s [8]. As with other previous research, the authors assume knowledge of time which provides a known dynamics model. The content of this chapter departs from existing research in that it considers the situation in which the spacecraft must recover not only its position and velocity but also the absolute time. Since time dictates the position of the gravitational bodies and thus the dynamic equations for the spacecraft, conventional filters like the EKF and sigma-point filter cannot be immediately employed.

5.3 Assumptions

This chapter assumes that the spacecraft is adrift in the vicinity of the Earth and Moon. The spacecraft possesses no knowledge of its trajectory, attitude, or the time, and it cannot communicate with the Deep Space Network or other Earth-based ranging facilities. It carries onboard cameras that provide digital images of the Earth, Sun, and Moon. I assume a camera arrangement which guarantees that all celestial bodies always remain in the field of regard of the camera system, and that they are not occluding one another. The camera system is not assumed to be able to resolve any features on any of the celestial bodies, but it is able to discriminate each from the others. An onboard clock measures time elapsed since starting the recovery procedure, and an onboard ephemeris table for each celestial body provides high-precision knowledge of its position for each time included in the table. I do not treat the image-processing aspect of this problem. Instead, this chapter assumes that an algorithm similar to that in [19] provides the necessary information from an image. I assume zero-

Table 5.1: Optical trajectory recovery assumptions

Available Information	Ephemerides for Earth, Sun, and Moon Elapsed time Measured size of celestial bodies Measured separation among celestial bodies
Unavailable Information	Present position Present velocity Present time Spacecraft attitude Features on celestial body surfaces
Measurement Assumptions	Persistent view of Earth, Sun, and Moon No occlusion of any celestial body by another Disambiguation among Earth, Sun, and Moon Zero-mean, Gaussian measurement noise

mean, Gaussian, uncorrelated measurement noise. Section 5.9 of this chapter discusses practical considerations associated with the described trajectory recovery method, including the consequences of lifting these assumptions. Table 5.1 summarizes all assumptions.

5.4 State Representation and Dynamics

This chapter describes a method for recovering the set of all spacecraft states which agree with camera observations of the Earth, Sun, and Moon. The state of the spacecraft includes its position and velocity in an Earth-centered inertial frame, and the absolute time. These states are coupled by the set of differential equations that govern how they evolve from time-step to time-step. Changes in position depend on the spacecraft's velocity, and changes in velocity depend on the gravitational landscape that the spacecraft occupies. This gravitational landscape depends not only on the position of the spacecraft, but also on the positions of the celestial bodies and, thus, the absolute time. See eqns. 5.1-5.4. I chose not to model J2 and higher-order effects, nor did I model perturbing forces like drag, because they are much less significant in cislunar space than at lower altitudes.

$$\begin{aligned}\mathbf{x} &= \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} & t \end{bmatrix}^T && (5.1)\\ &= \begin{bmatrix} \mathbf{r}_{ec} & \dot{\mathbf{r}}_{ec} & t \end{bmatrix}^T && (5.2)\end{aligned}$$

$$\begin{aligned}\ddot{\mathbf{r}}_{ec} &= -\frac{\mu_e}{(\mathbf{r}_{ec}^T\mathbf{r}_{ec})^{\frac{3}{2}}}\mathbf{r_{ec}} + \mu_m\left(\frac{\mathbf{r}_{em}-\mathbf{r}_{ec}}{(\mathbf{r}_{cm}^T\mathbf{r}_{cm})^{\frac{3}{2}}} - \frac{\mathbf{r}_{em}}{\rho_{em}^3}\right) + \mu_s\left(\frac{\mathbf{r}_{es}-\mathbf{r}_{ec}}{(\mathbf{r}_{cs}^T\mathbf{r}_{cs})^{\frac{3}{2}}} - \frac{\mathbf{r}_{es}}{\rho_{es}^3}\right) && (5.3)\\ \dot{t} &= 1 && (5.4)\end{aligned}$$

5.5 Measurement Representation

As discussed in Section 5.3, this chapter assumes that the Earth, Sun, and Moon all remain in the field of regard of the camera system, and that they do not occlude one another. The spacecraft may therefore gather measurements of the sizes and relative positions of each of the celestial bodies at each moment in

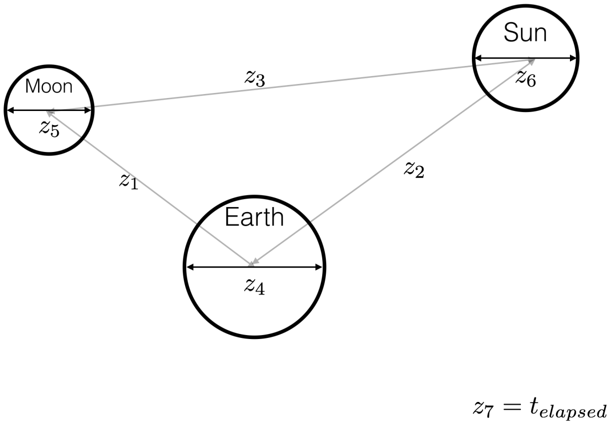


Figure 5.1: Measured quantities

time. Specifically, these measurements include the width, in pixels, of each celestial body and the separation, in pixels, among each of the celestial bodies. Fig. 5.1 illustrates these measurements. The onboard clock enables the spacecraft to also measure elapsed time. In order for these measurements to provide information about the spacecraft state (eqn. 5.2), these measurements must be represented in terms of the state variables. Doing so yields the nonlinear measurement model given by eqns. 5.5, and simplified in eqn. 5.6.

$$
\begin{aligned}
z_1 &= \cos^{-1}\left[\frac{-xd_{\mathrm{mx}} - yd_{\mathrm{my}} - zd_{\mathrm{mz}} + (-x)^2 + (-y)^2 + (-z)^2}{\sqrt{(-x)^2 + (-y)^2 + (-z)^2}\sqrt{(d_{\mathrm{mx}} - x)^2 + \left(d_{\mathrm{my}} - y\right)^2 + (d_{\mathrm{mz}} - z)^2}}\right]\frac{P}{\Theta} \\
z_2 &= \cos^{-1}\left[\frac{-xd_{\mathrm{sx}} - yd_{\mathrm{sy}} - zd_{\mathrm{sz}} + (-x)^2 + (-y)^2 + (-z)^2}{\sqrt{(-x)^2 + (-y)^2 + (-z)^2}\sqrt{(d_{\mathrm{sx}} - x)^2 + \left(d_{\mathrm{sy}} - y\right)^2 + (d_{\mathrm{sz}} - z)^2}}\right]\frac{P}{\Theta} \\
z_3 &= \cos^{-1}\left[\frac{d_{\mathrm{mx}}(d_{\mathrm{sx}} - x) + d_{\mathrm{my}}\left(d_{\mathrm{sy}} - y\right) + d_{\mathrm{sz}}(d_{\mathrm{mz}} - z) - zd_{\mathrm{mz}} - xd_{\mathrm{sx}} - yd_{\mathrm{sy}} + (-x)^2 + (-y)^2 + (-z)^2}{\sqrt{(d_{\mathrm{mx}} - x)^2 + \left(d_{\mathrm{my}} - y\right)^2 + (d_{\mathrm{mz}} - z)^2}\sqrt{(d_{\mathrm{sx}} - x)^2 + \left(d_{\mathrm{sy}} - y\right)^2 + (d_{\mathrm{sz}} - z)^2}}\right]\frac{P}{\Theta} \\
z_4 &= \frac{2P}{\Theta}\tan^{-1}\left(\frac{r_e}{\sqrt{(-x)^2 + (-y)^2 + (-z)^2}}\right) \\
z_5 &= \frac{2P}{\Theta}\tan^{-1}\left(\frac{r_m}{\sqrt{(-x + d_{mx})^2 + (-y + d_{my})^2 + (-z + d_{mz})^2}}\right) \\
z_6 &= \frac{2P}{\Theta}\tan^{-1}\left(\frac{r_s}{\sqrt{(-x + d_{sx})^2 + (-y + d_{sy})^2 + (-z + d_{sz})^2}}\right) \\
z_7 &= t - t_0
\end{aligned}
\tag{5.5}
$$

In eqn. 5.5, d_{ij} (where $i = m, s$ and $j = x, y, z$) represents the time-dependent x, y, and z position of the Moon and the Sun in an Earth-centered inertial frame. These quantities may be pulled directly from the onboard ephemeris table for each body, but knowledge of the absolute time is required to pull from the correct row of the table. z_1, z_2 and z_3 are the pixel separations among the three celestial bodies. z_4, z_5, and z_6 are the pixel widths of the three celestial bodies. z_7 is the elapsed time since beginning to recover time and trajectory. There is noise associated with each measurement. As stated in Section 5.3, this chapter assumes zero-mean, Gaussian, uncorrelated noise for all measurements. The measurement error covariance matrix therefore takes the form shown in eqn. 5.7.

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \end{bmatrix} = \begin{bmatrix} h_1(\mathbf{x}) \\ h_2(\mathbf{x}) \\ h_3(\mathbf{x}) \\ h_4(\mathbf{x}) \\ h_5(\mathbf{x}) \\ h_6(\mathbf{x}) \\ h_7(\mathbf{x}) \end{bmatrix} = \mathbf{h}(\mathbf{x}) \tag{5.6}$$

$$R = \begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_4^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_5^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_6^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_7^2 \end{bmatrix} \tag{5.7}$$

5.6 Search Space Reduction

The recovery algorithm aims to find all relative minima of the cost function shown in eqn. 5.8. The search space for this problem is extensive. The spacecraft could be anywhere within 800,000 km of the Earth, traveling anywhere between zero to tens of km/sec in any direction, at any moment in the mission lifetime. Properly populating that search space with particles for a particle filter would be prohibitively expensive, especially for the sorts of processors used by small spacecraft. Instead, one can use a batch of measurements to coarsely reduce the search space. This batch of measurements includes a rapidly gathered collection of width measurements for Earth, Moon, and Sun (in pixels), and a rapidly gathered collection of angular separation measurements among Earth, Moon, and Sun (in pixels).

$$J(\mathbf{x}) = \frac{1}{2}\left[\mathbf{z} - h(\mathbf{x})\right]^T R^{-1} \left[\mathbf{z} - h(\mathbf{x})\right] \tag{5.8}$$

5.6.1 Reducing search space in the temporal dimension

The batches of Earth-width measurements, Moon-width measurements, and Earth-Moon angle measurements are used to reduce the search space through time. The spacecraft calculates the mean and variance of the Earth-width measurement batch, the Moon-width measurement batch, and the Earth-Moon angle measurement batch. From these means and variances, a normal distribution is generated for each set of measurements that includes many more samples than batched measurements. Each of these samples is then transformed using eqns. 5.9-5.11. The Earth-width samples and Moon-width samples are transformed from pixels to distance measurements (in kilometers) from the spacecraft to each celestial body (eqn. 5.9 and eqn. 5.10, respectively). The angular separation measurements are transformed from pixels to radians (eqn. 5.11). This yields a new set of distributions for distance to Earth (in km), distance to Moon (in km), and separating angle between Earth and Moon (in radians).

$$\rho_{ce} \approx \frac{r_e}{\tan \frac{z_4 \Theta}{2P}} \tag{5.9}$$

$$\rho_{cm} \approx \frac{r_m}{\tan \frac{z_5 \Theta}{2P}} \tag{5.10}$$

$$\theta_{ecm} \approx z_1 \frac{\Theta}{P} \tag{5.11}$$

Eqn. 5.12 converts each set of Earth-distance, Moon-distance, and Earth-Moon angle samples to a separation distance between Earth and Moon. For each row of the onboard ephemeris table, the true separating distance between the Earth and Moon is computed. Fig. 5.2 shows the true and calculated Earth/Moon separated distances over 75 days as calculated by a spacecraft on a lunar flyby orbit. The spacecraft in question is assumed to gather pixel measurements with a standard deviation of error of 0.1 pixels from a camera with 3280 x 2464 pixel resolution. By subtracting the approximated separating distance

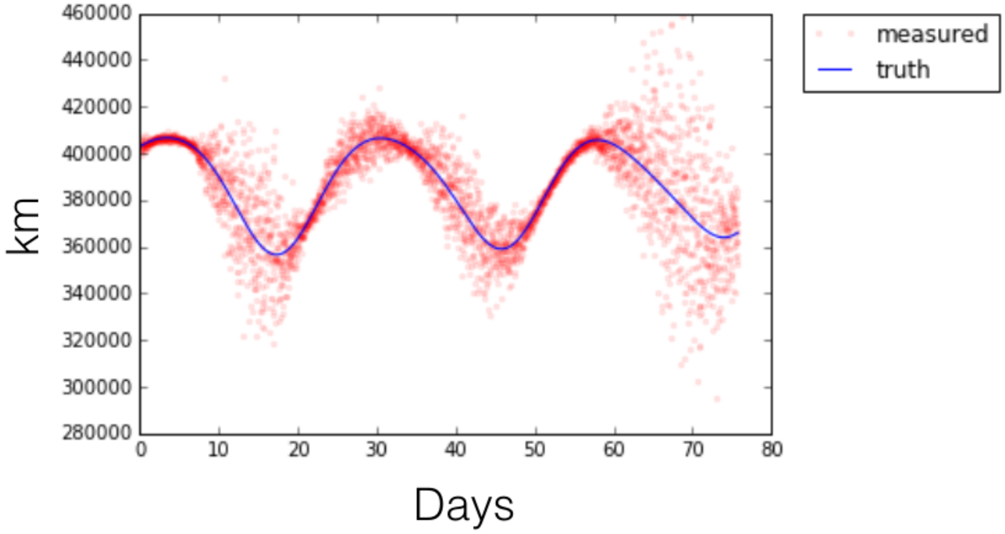


Figure 5.2: True and measured separation between Earth and Moon

from the true separation distances and finding all of the nearest approaches to zero, the spacecraft arrives at a probabilistic distribution for time. Section 5.8 contains an example of this. A set of maximum likelihood estimates is obtained by retaining each relative maximum in the sampled distribution. These preliminary estimates need not and will not be accurate. The goal is simply to reduce the search space to a discrete number of guesses.

$$\rho_{em} \approx (\rho_{ce}^2 + \rho_{cm}^2 - 2\rho_{ce}\rho_{cm}\cos\theta_{ecm})^{\frac{1}{2}} \tag{5.12}$$

5.6.2 Reducing search space in the spatial dimensions

The sampled distances from spacecraft to Earth place the spacecraft anywhere on the surface of one of a family of spheres which surround the Earth, each of which with a radius equal to a sampled Earth distance. Similarly, the sampled distances from spacecraft to Moon place the spacecraft anywhere on the surface of one of a family of spheres surrounding the Moon. The position of the spacecraft is constrained to the intersections of all these spheres, which place it somewhere on one of a family of rings surrounding the line connecting the Earth and Moon. The centers of these rings lie on the line connecting the center of the Earth to the center of the Moon, and they are radially perpendicular to that line. There is a ring associated with each possible moment in time. From any of these locations in space and time, the expected measurements for Earth width, Moon width, and Earth-Moon angle of separation will agree with those gathered.

Two batches of measurements remain for further reducing the search space of possible spacecraft locations. These include the batch of measured Earth-Sun angles and the batch of measured Moon-Sun angles. In order to incorporate this information, the cost associated with each point on each ring is computed using eqn. 5.8. For each ring, there exist two locations that agree with all measurements: one is above the plane formed by the Earth, Sun and Moon, and the other is symmetrically across that plane.

The search space is thus reduced to a collection of positions in space and time. For each moment in time, there is a pair of positions (one position above the plane formed by the Earth, Sun, and Moon, and one symmetrically below that plane). The number of pairs depends on the length of the ephemeris table. If the mission lifetime (and thus the length of the ephemeris table) is less than the orbital period of the moon, then the search space reduction yields a single pair of positions (for non-singular celestial configurations, see Section 5.9). If the ephemeris table includes multiple orbits of the Moon around the Earth, then one finds multiple pairs of positions. For each location, a particle filter is instantiated to recover the associated velocity and refine estimates for position and time.

5.7 Particle Filter

This subsection briefly discusses the general structure of a particle filter, and then discusses the specific instantiation for the trajectory recovery problem under consideration. At the end of this subsection, there is an abridged overview of the filter.

5.7.1 General Structure

The general structure of a particle filter is as follows. First, the search space is populated with a large number of potential solutions (particles). When the filter begins, each of these potential solutions has an equal probability of being the one that best represents the true state of the system. Each particle is propagated forward in time one time step by integrating its dynamic equations (eqn. 5.3-5.4). This transforms the initial collection of particles into a new collection of particles which may have had its shape distorted by the nonlinear equations that describe the time evolution of each particle.

Next, a set of expected measurements is generated for each particle in the collection using the measurement model. A true measurement is then gathered and compared to the hypothetical measurements that would be expected from each particle. The particles are re-weighted by comparing the true and hypothetical measurements, with those that do a better job predicting the actual measurements receiving more weight than those that do a worse job. An estimate for the state is calculated by taking a weighted average, and an estimate for the covariance is calculated numerically. The whole process is then repeated for the next time step. If one particle begins to dominate, then a new set of particles is generated based on that particle, all are given equal weight, and the process starts over.

5.7.2 Specific Instantiation

A particle filter is instantiated from each location/time pair in the reduced search space. For each pair, N_s particles are generated with Gaussian distributions in location, velocity, and time as shown in eqns. 5.3-5.4. The initial covariance is set to a value large enough to ensure that some of the particles populate the correct region of the search space. Both of these parameters (the number of particles and the initial covariance) are filter design elements with no optimal answer. Unlike in linear systems and filters, one cannot solve for an optimal particle filter design. Instead, numerical simulations are required in order to identify parameter values that work well enough for a particular application. In Section 5.8, I enumerate the values that were used for a case study trajectory recovery, but different values will work better for spacecraft on different trajectories. The filter design, however, remains constant.

Initially, each particle, which is characterized by a position, velocity, and a time, is instantiated with equal weight. Each particle is then propagated by numerically integrating its own dynamics model. After the particles have been propagated to the next timestep, their weights are re-calculated by comparing the measurements that would be expected from each particle (as calculated by the measurement model, eqns. 5.5-5.6) to the actual measurements. Because of the high gradients and size of the searchcspace being traversed, a unique weighting method is required for the particle filter to converge.

First, the innovation between each particle's expected measurement and the true measurement is calculated. Ordinarily, one calculates the weight of each particle by taking the exponential of the R-norm of the innovation. Instead, because the innovations in this problem have such a wide range of values, one can maintain better scaling on the weights by calculating the natural log of the exponential and adding the log of the previous weight.

With this weighting convention, particles that do a better job predicting the true measurement receive a greater weight. It is possible that the weights calculated in eqn. 5.17 will have a huge range of values. They are rescaled by the maximum weight. Finally, one can find the minimum weight and rescale once more so that the maximally-weighted particle has weight 1. The rescaled weights are given by the exponential of the difference between each weight and the minimum weight. The result is that better particles have greater weight, and worse particles have lesser weight. The updated state and covariance estimates are given by the weighted average of the propagated particles.

The effective number of particles can be calculated using eqn. 5.23. If the effective number of particles is greater than the resampling threshold, N_t, then the particles retain their weights and are propagated again to repeat the above cycle for the next timestep. If not, then the particles are resampled and their weights are reset according to eqn. 5.24. The nature of this problem calls for a somewhat unique method for resampling. In a conventional problem, one uses roulette selection to probabilistically resample particles, resulting in multiple copies of high-weight particles and fewer (or zero) copies of low-weight particles. The repeated high-weight particles are then spread out via process noise and dithering by sampling from an Epanechnikov kernel function. For this particular problem, that method of resampling does not work reliably. Because the measurements depend only on position and time (and not on velocity), information must accumulate for a number of timesteps in order to learn anything about the spacecraft's velocity. By instead resampling such that the spacecraft's position is driven almost entirely by the velocity, one can more quickly converge on the correct answer.

In the event that the number of effective particles below the threshold, the particles are replaced with a new set. The position and time states for each of these new particles are given by the current position and time estimates. The velocity for each particle is sampled from a multivariate Gaussian distribution with a mean of the latest velocity estimate and a covariance of the sum of the error covariance estimate and the process noise covariance. In order to maintain diversity, the process noise covariance for the velocity states is kept relatively large. This resampling procedure is summarized in eqn. 5.24-5.25. After resampling, the particles are again propagated according to eqn. 5.3-5.4, a new measurement is gathered, and the process is repeated.

Summary of filter steps

1. Generate particles and particle weights:

$$\chi_i^0 = \mathcal{N}\left(\mathbf{x}_i(0), P(0)\right) \tag{5.13}$$

$$w_i^0 = \frac{1}{N_s} \tag{5.14}$$

2. Propagate particles through dynamics equations:

$$\chi_i^1 = \mathbf{f}\left(\chi_i^0\right) \tag{5.15}$$

3. Calculate the innovation of each particle:

$$\nu_i^1 = \mathbf{z}^1 - \mathbf{h}\left(\chi_i^1\right) \tag{5.16}$$

4. Find log-based weights:

$$log(w_i^1) = -\frac{1}{2}\left((\nu_i^1)^T R \nu_i^1\right) + log(w_i^0) \tag{5.17}$$

5. Rescale log-based weights:

$$log(w_i^1) = \frac{log(w_i^1)}{max(log(w_i^1))} \tag{5.18}$$

6. Solve for weights, rescale such that maximum is of value 1:

$$w_i^1 = e^{min(log(w_i^1))-log(w_i^1)} \tag{5.19}$$

7. Update the state and covariance estimates:

$$\hat{\mathbf{x}}(1) = \sum_{k=0}^{N_s} w_i^1 \chi_i^1 \tag{5.20}$$

$$P(1) = \sum_{k=0}^{N_s} w_i^1 \left(\chi_i^1 - \hat{\mathbf{x}}(1)\right)\left(\chi_i^1 - \hat{\mathbf{x}}(1)\right)^T \tag{5.21}$$

$$\tag{5.22}$$

8. Calculate the effective number of particles:

$$N_{eff} = \frac{1}{\sum_{k=0}^{N_s}\left(w_i^1\right)^2} \tag{5.23}$$

9. If the effective number of particles is above the resample threshold, all particles retain their weights and are propagated to the next time step. If not, the particles are resampled:

$$\chi_i^1 = \begin{bmatrix} \hat{x}(1) & \hat{y}(1) & \hat{z}(1) & \mathcal{N}\left(\begin{bmatrix} \hat{\dot{x}}(1) \\ \hat{\dot{y}}(1) \\ \hat{\dot{z}}(1) \end{bmatrix}, P(1) + Q\right) & \hat{t}(1) \end{bmatrix}^T \tag{5.24}$$

$$w_i^1 = \frac{1}{N_s} \tag{5.25}$$

5.8 Case Study

The subsection considers a simulated trajectory recovery. The spacecraft in question is *en route* to the Moon, at a distance approximately halfway between Earth and Moon on December 15, 2017. It has lost knowledge of its location,

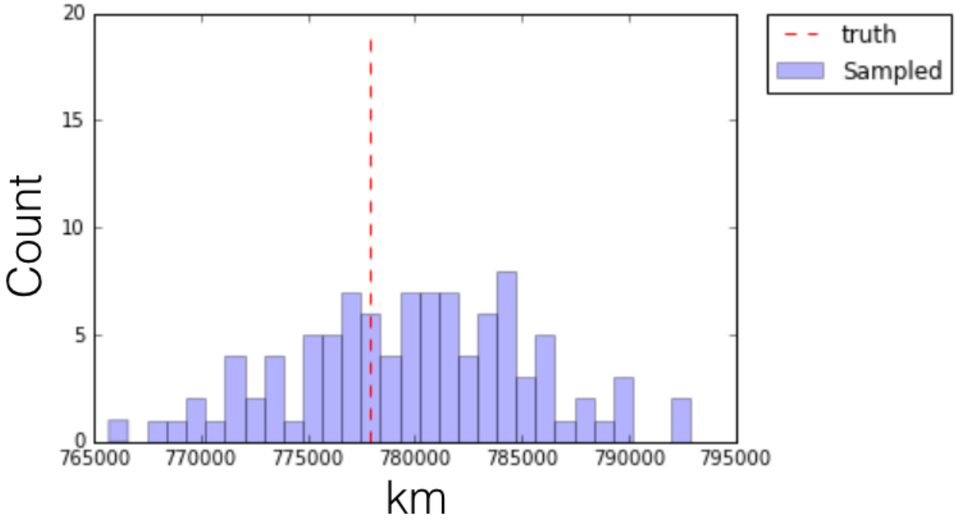


Figure 5.3: True and measured Earth distance

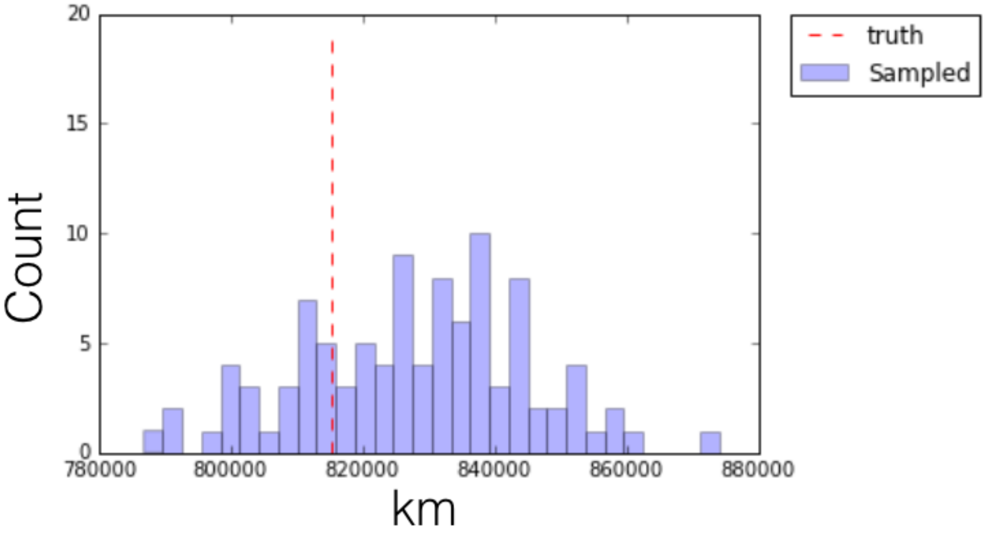


Figure 5.4: True and measured Moon distance

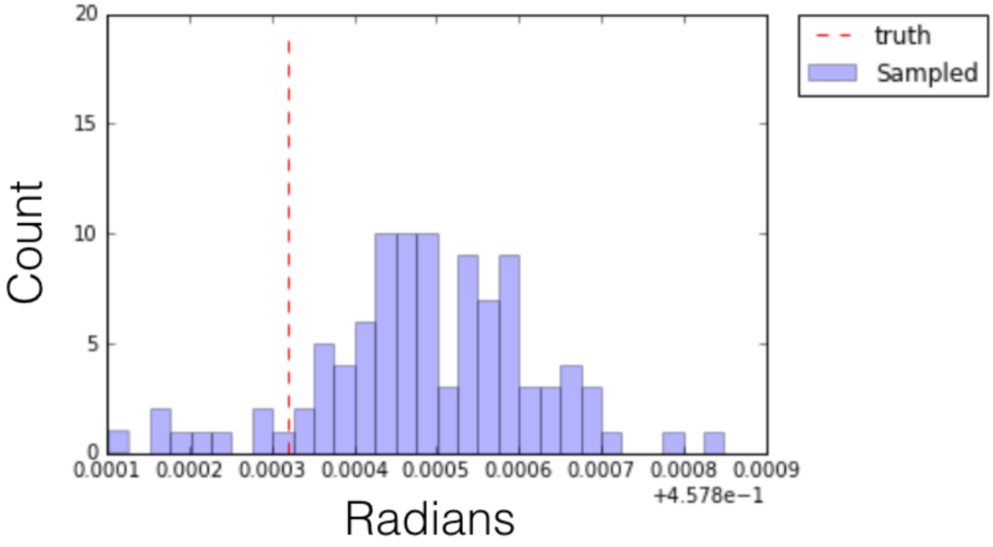


Figure 5.5: True and measured Earth-Moon angular separation

velocity, attitude, and the time. It must use its onboard cameras, ephemerides tables, and timer to recover all of its possible trajectories, per the method described in previous subsections. This case study assumes that the spacecraft is equipped with a camera system composed of 2500x2500 pixel cameras with fields of view tiled such that the field of regard includes all directions. Camera measurements of the angular widths and separations of and among celestial bodies are assumed to have a zero-mean Gaussian error with standard deviation of 0.1 pixels. Per the assumptions of Section 5.3, no celestial body occludes any other at the time of trajectory recovery.

Temporal reduction of search space

The spacecraft first reduces its searchspace in time, per section 5.6.1. It does so by rapidly gathering images of the Earth and Moon, and then uses eqns.

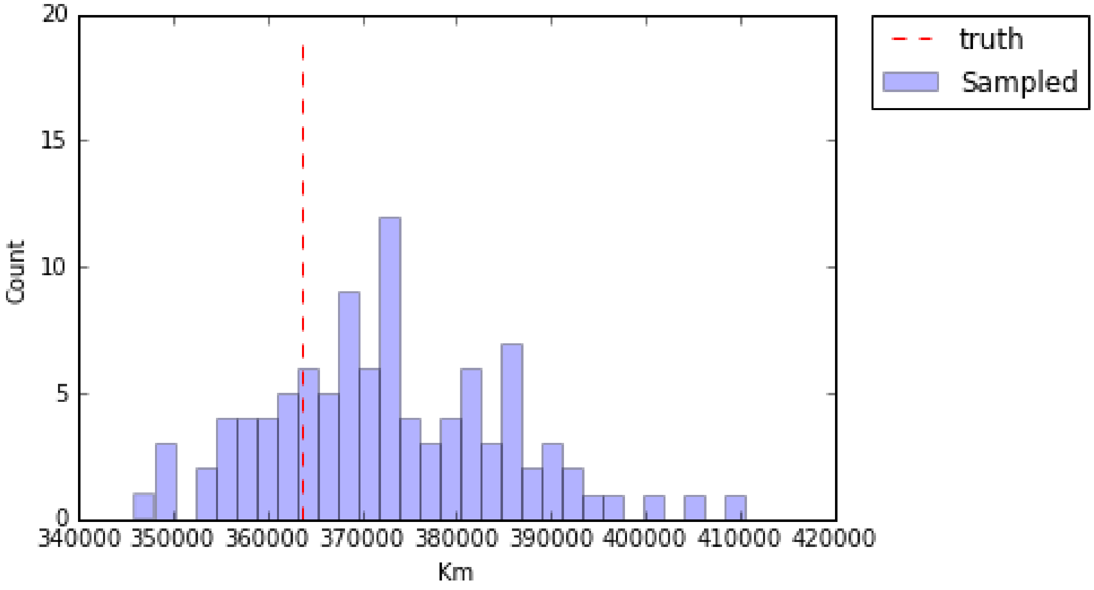


Figure 5.6: True and measured Earth-Moon separation distance

5.9-5.11 to convert those measurements to Earth-distance, Moon-distance, and Earth-Moon angle. The spacecraft calculates the mean and variance of each of these distributions, and uses that mean and variance to generate 100 samples for Earth distance, Moon distance, and separating angle. Figs. 5.3-5.5 show these distributions, along with the true quantities from the simulation (which are unknown to the simulated spacecraft).

For each set of Earth distance, Moon distance, and Earth-Moon angle, the spacecraft uses eqn. 5.12 to calculate the distance separating the Earth and Moon. The results, for this case study, are shown in Fig. 5.6. Each of these calculated distances is then compared to the true distances separating the Earth and Moon for the entire mission lifetime, as calculated from the onboard ephemerides tables. This generates the probabilistic distribution in time, shown in Fig. 5.7. The ephemerides tables used for this simulation are the same that

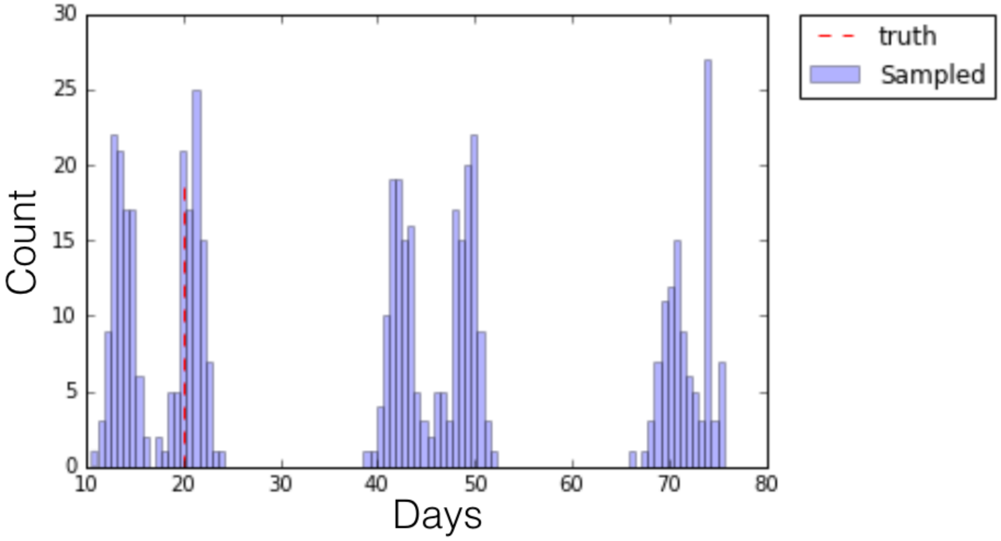


Figure 5.7: Probabilistic distribution for time from Earth-Moon separation, t = 20 days

were used to generate Fig. 5.2. The maximum likelihood estimates for time are ascertained by finding all relative maxima of this distribution.

Spatial reduction of search space

The distributions for distance to Earth and Moon (Figs. 5.3-5.4) and the distribution for separation distance between Earth and Moon (Fig. 5.6) can also be used to isolate the spacecraft to a collection of rings surrounding the line connecting the Earth and Moon. For each Earth-distance approximation, there is a sphere surrounding the Earth on which the spacecraft could reside. The same is true for the Moon distance approximations. The intersections of these spheres comprise a distribution of rings between the Earth and Moon that agree with the separation measurements. Figs. 5.8-5.9 show these distributions. The Earth/Moon

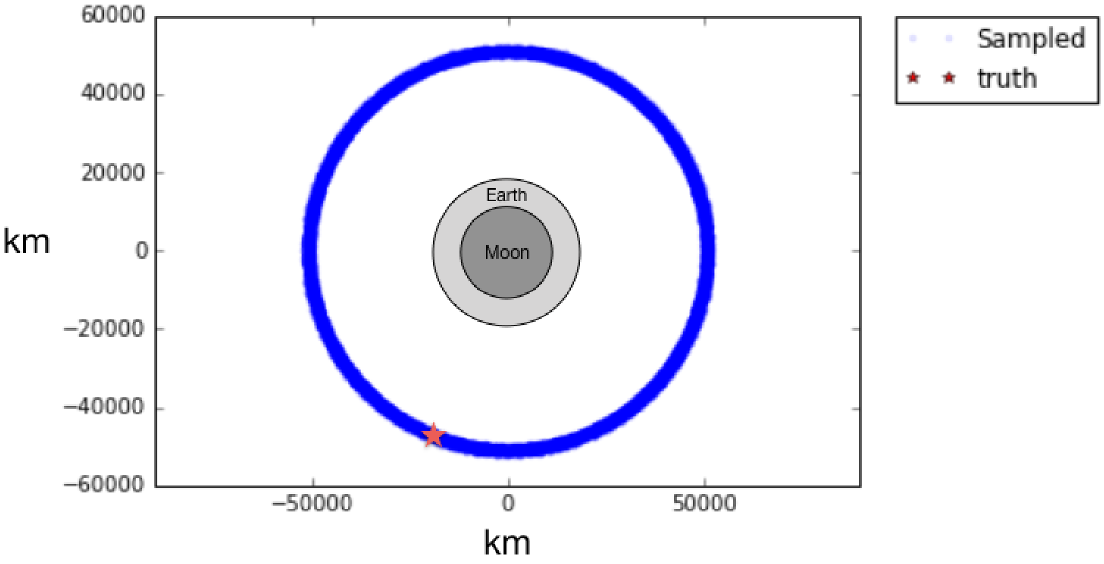


Figure 5.8: Distribution of rings of spacecraft positions (Earth/Moon not to scale, only to show orientation of ring)

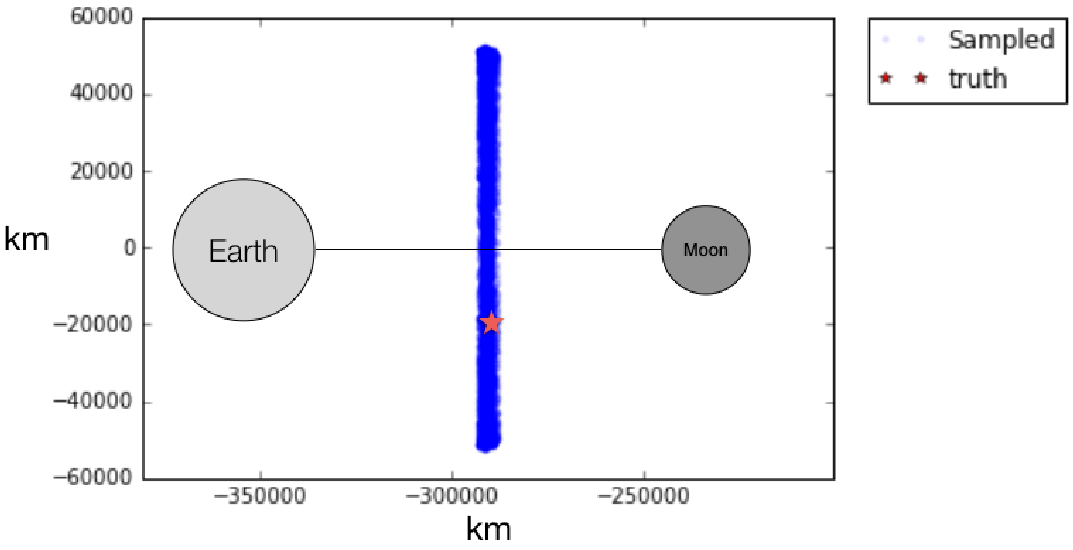


Figure 5.9: Distribution of rings of spacecraft positions (Earth/Moon sizes and distances not to scale, only to show orientation of ring)

sizes and distances are not to scale in these figures, they are for illustrating the orientation of the ring in space. These figures also show the true location of the spacecraft to illustrate that it falls within the reduced search space.

For each possible moment in time (Fig. 5.7), the spacecraft may reside anywhere on the rings shown in Figs 5.8-5.9. From any of these locations in space and time, the expected measurements for Earth width, Moon width, and Earth-Moon angle of separation will agree with those gathered. As described in 5.6.2, the spacecraft further reduces the search space by calculating the cost (eqn. 5.8) associated with each point in this ring. Doing so incorporates the final two measurements: Earth-Sun angle and Moon-Sun angle. Fig. 5.10 shows the true position of the spacecraft in the ring and the least costly particle in the ring. The rest of the particles are plotted with an opacity in proportion to their cost. For configurations of Earth, Sun, and Moon which span *R*2 (see section 5.9), there

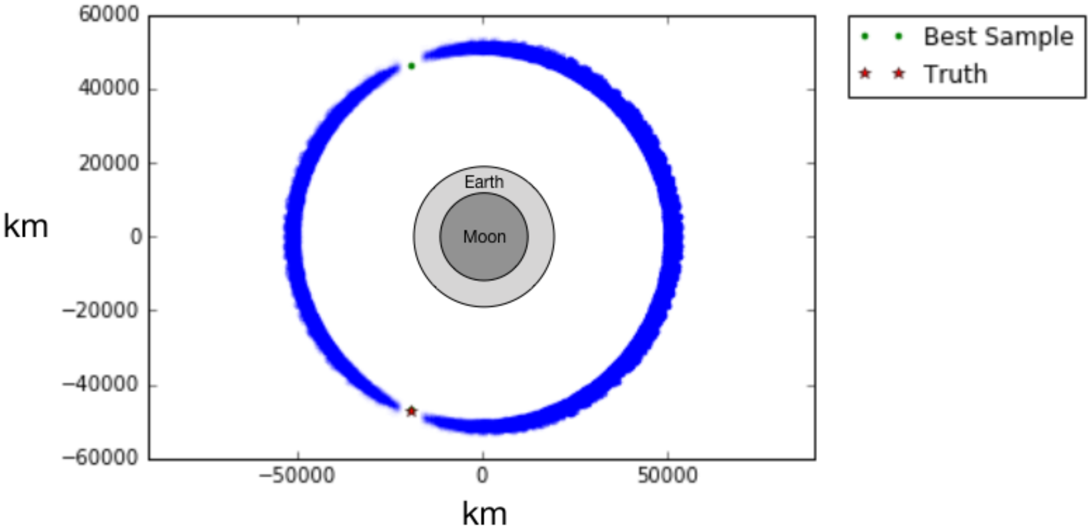


Figure 5.10: Distribution of rings of spacecraft positions (Earth/Moon not to scale, only to show orientation of ring)

are two locations on the ring that agree with all measurements. One is above the plane formed by the Earth, Sun and Moon, and the other is symmetrically across that plane.

A pair of such points exists for each possible moment in time. This reduces the search space to a series of clusters in space and time. For each moment in time, there is a pair of clusters (one cluster above the plane formed by the Earth, Sun, and Moon, and one symmetrically below that plane). If the mission lifetime were less than the orbital period of the moon, one would expect to find only two clusters. Since the mission under consideration in this analysis is long enough to include multiple orbits of the moon around the Earth, one finds multiple pairs of clusters. These are shown in Fig. 5.11. It can be seen that the true spacecraft position is within one of these clusters. From each cluster, I instantiate a particle filter.

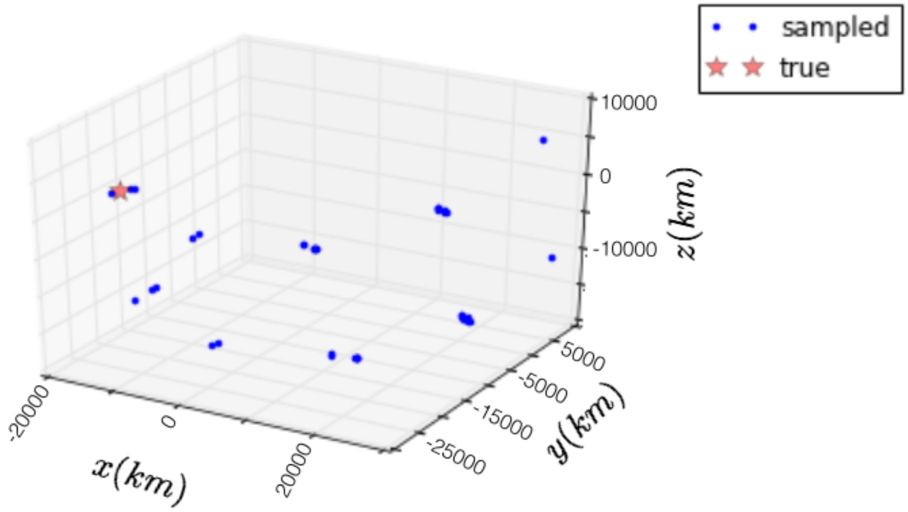


Figure 5.11: Reduced spacial searchspace, ECI frame

Table 5.2: Particle filter case-study parameters

Number of particles	N_s=1000
Resample threshold	N_t=200
Initial position estimate	cluster mean
Initial velocity estimate	0
Initial time estimate	cluster mean
Initial position error std. dev.	8000 km
Initial velocity error std. dev.	5 km/sec
Initial time error std. dev.	4 hours
Pixel measurements std. dev.	0.5 pixels
Elapsed time measurements std. dev.	0.0014 sec

Particle Filter

The particle filter is described in section 5.7.2, and summarized in eqns. 5.13-5.25. For this case study, I used the particular parameter values given in Table 5.2. Running this particle filter on each cluster yields the family of trajectories shown in Fig. 5.12. The true spacecraft trajectory is overlayed on top of this family of trajectories to illustrate that the spacecraft is indeed following one of the collection.

5.9 Practical Considerations

The trajectory recovery strategy considered in this chapter has a long list of assumptions, which implies a long list of practical considerations. Some of those

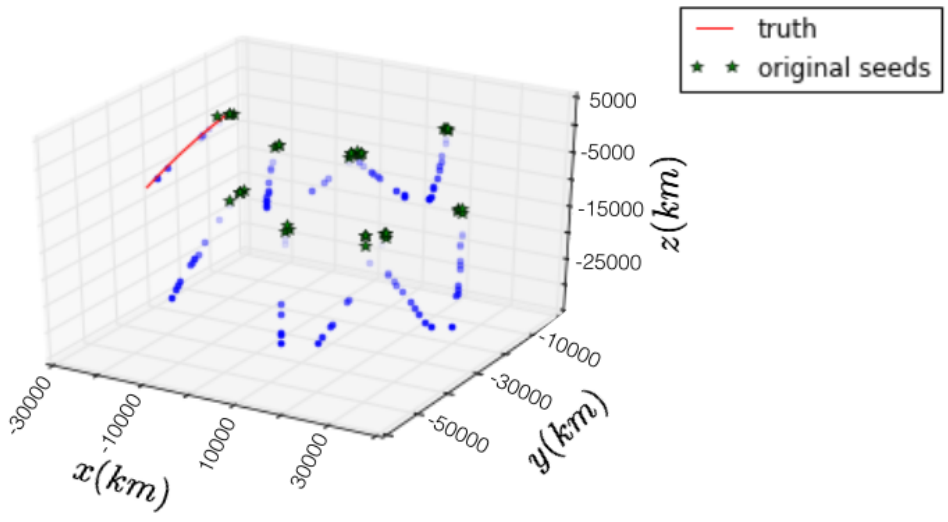


Figure 5.12: Family of trajectories which agree with gathered measurements

are considered in this section. In particular, I consider the case where the spacecraft is able to choose the correct cluster (as shown in Fig. 5.11) from the collection of clusters, I consider position ambiguities and singularities in the measurement model, and I consider how performance would be affected by the addition of attitude knowledge.

5.9.1 Selecting a cluster

Each of the clusters shown in Figs. 5.11-5.12 is separated from the others by tens of thousands of kilometers in distance and multiple days in time. In the situation that the spacecraft has sufficient *a priori* knowledge of its trajectory or of the time to choose one of these clusters, then it can use the particle filter to recover its trajectory. This is accomplished by simply running the particle filter

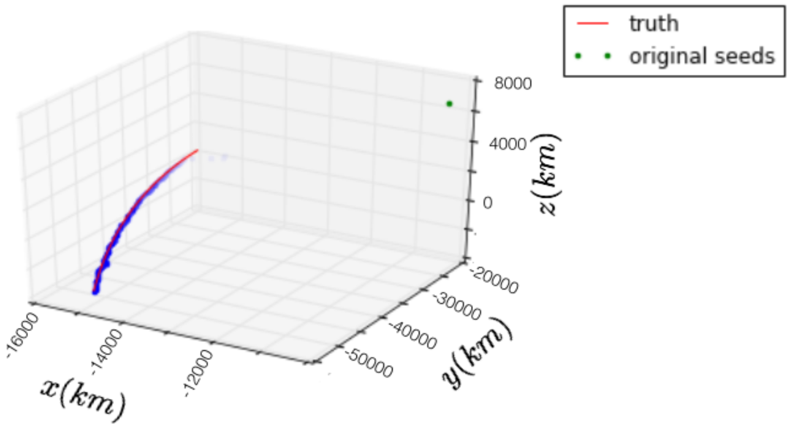


Figure 5.13: True and estimated spacecraft trajectory

on only the correct cluster.

For this case study, a point in the cluster nearest the spacecraft is used to seed a particle filter as described in section 5.7.2. The filter converges to within tens of kilometers of the true spacecraft position, tenths of kilometers per second of true spacecraft velocity, and tens of minutes of the true time in approximately 500 seconds, as shown in Figs. 5.13-5.15. In Fig. 5.13, more recent estimates have greater opacity than older estimates.

5.9.2 Position ambiguities and singularities

Assuming that the celestial bodies are in a non-singular configuration (they span R^2), then these measurements isolate the spacecraft to one of two positions. One is above the plane formed by the Earth, Sun, and Moon, and the other is symmetrically below the plane. This symmetry becomes apparent when one transforms the vectors of Eqns. eqns. 5.5-5.6 from the ECI coordinates to a coordinate system generated by performing a Gram-Schmidt orthogonalization of the vectors from Earth to Moon and Earth to Sun, defined by unit vectors $\hat{\mathbf{e}}_x$, $\hat{\mathbf{e}}_y$, and $\hat{\mathbf{e}}_z$. Doing so defines a coordinate system in which the $\hat{\mathbf{e}}_x$ axis points from the Earth to the Moon, the $\hat{\mathbf{e}}_y$ axis is orthogonal to the $\hat{\mathbf{e}}_x$ axis and points in the direction of the Sun, and the $\hat{\mathbf{e}}_z$ axis is orthogonal to the plane formed by the Earth, Sun, and Moon.

$$\hat{\mathbf{e}}_x = \frac{\mathbf{R}_{EM}}{||\mathbf{R}_{ES}||} \tag{5.26}$$

$$i = \hat{\mathbf{e}}_x \cdot \mathbf{R}_{ES} \tag{5.27}$$

$$\hat{\mathbf{e}}_y = \frac{\mathbf{R}_{ES} - i\hat{\mathbf{e}}_x}{||\mathbf{R}_{ES} - i\hat{\mathbf{e}}_x||} \tag{5.28}$$

$$\hat{\mathbf{e}}_z = \hat{\mathbf{e}}_x \times \hat{\mathbf{e}}_y \tag{5.29}$$

$\mathbf{R}_{EM}$ is the vector connecting Earth to Moon and $\mathbf{R}_{ES}$ is the vector connecting Earth to Sun. The transformation from the ECI to these time-varying coordinates defined by the ephemerides (the ”ephemeral” frame, denoted by superscript e or the tilde) is formed by stacking these unit vectors into a rotation ma-

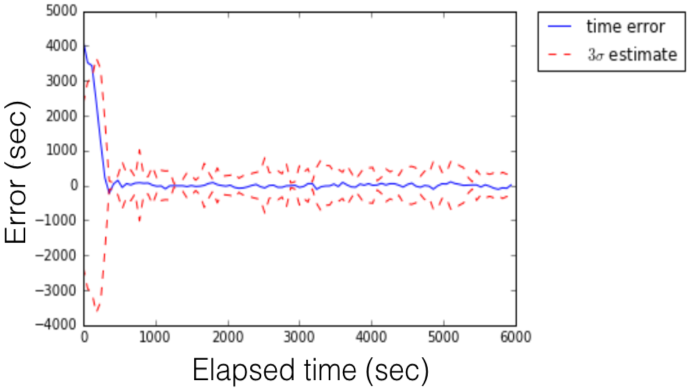


Figure 5.14: Error in estimated time

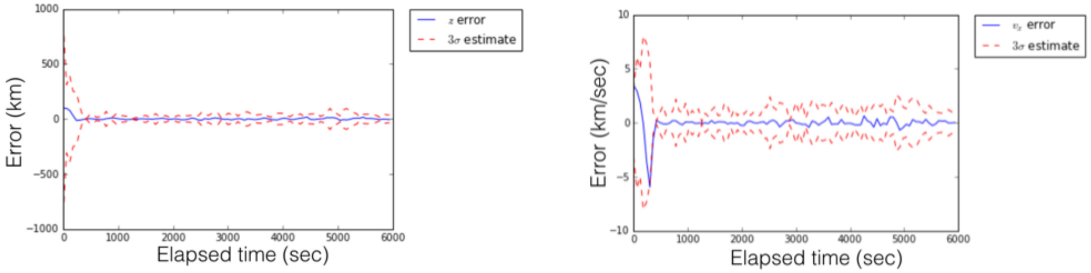


Figure 5.15: Error in estimated z-velocity and z-position

trix.

$$ {}^{e}Q^{E} = \begin{bmatrix} {}^{E}\hat{\mathbf{e}}_{\tilde{x}} & {}^{E}\hat{\mathbf{e}}_{\tilde{y}} & {}^{E}\hat{\mathbf{e}}_{\tilde{z}} \end{bmatrix} \tag{5.30} $$

$$ {}^{e}Q^{B} = \begin{bmatrix} {}^{B}\hat{\mathbf{e}}_{\tilde{x}} & {}^{B}\hat{\mathbf{e}}_{\tilde{y}} & {}^{B}\hat{\mathbf{e}}_{\tilde{z}} \end{bmatrix} \tag{5.31} $$

These rotation matrices and unit vectors can be used to define the new variables shown in eqns. 5.32-5.36.

$$ \begin{bmatrix} \tilde{x} & \tilde{y} & \tilde{z} \end{bmatrix}^{T} = {}^{e}Q^{E}\ {}^{E}\begin{bmatrix} x & y & z \end{bmatrix}^{T} \tag{5.32} $$

$$ = {}^{e}Q^{B}\ {}^{B}\begin{bmatrix} x & y & z \end{bmatrix}^{T} \tag{5.33} $$

$$ \tilde{d}_{mx} = \mathbf{R}_{EM} \cdot \hat{\mathbf{e}}_{\tilde{x}} = \mathbf{R}_{EM} \cdot \frac{\mathbf{R}_{EM}}{||\mathbf{R}_{EM}||} = ||\mathbf{R}_{EM}|| \tag{5.34} $$

$$ \tilde{d}_{sx} = \mathbf{R}_{ES} \cdot \frac{\mathbf{R}_{EM}}{||\mathbf{R}_{EM}||} \tag{5.35} $$

$$ \tilde{d}_{sy} = \mathbf{R}_{ES} \cdot \left[\mathbf{R}_{ES} - \left(\mathbf{R}_{ES} \cdot \frac{\mathbf{R}_{EM}}{||\mathbf{R}_{EM}||} \right) \frac{\mathbf{R}_{EM}}{||\mathbf{R}_{EM}||} \right] \tag{5.36} $$

By construction of the frame, the Earth, Sun, and Moon share the same $\tilde{z}$ coordinate. In these coordinates, the vectors from the spacecraft to each celestial body are given by eqns. 5.37. The transformed measurement model is given by eqns. 5.38-5.42.

$$
{}^{e}\hat{\mathbf{r}}_{ce} = \begin{bmatrix} \frac{\tilde{x}}{\left(\tilde{x}^2+\tilde{y}^2+\tilde{z}^2\right)^{\frac{1}{2}}} \\ \frac{\tilde{y}}{\left(\tilde{x}^2+\tilde{y}^2+\tilde{z}^2\right)^{\frac{1}{2}}} \\ \frac{\tilde{z}}{\left(\tilde{x}^2+\tilde{y}^2+\tilde{z}^2\right)^{\frac{1}{2}}} \end{bmatrix} \quad {}^{e}\hat{\mathbf{r}}_{cm} = \begin{bmatrix} \frac{\tilde{x}+\tilde{d}_{mx}}{\left((\tilde{x}+\tilde{d}_{mx})^2+\tilde{y}^2+\tilde{z}^2\right)^{\frac{1}{2}}} \\ \frac{\tilde{y}}{\left((\tilde{x}+\tilde{d}_{mx})^2+\tilde{y}^2+\tilde{z}^2\right)^{\frac{1}{2}}} \\ \frac{\tilde{z}}{\left((\tilde{x}+\tilde{d}_{mx})^2+\tilde{y}^2+\tilde{z}^2\right)^{\frac{1}{2}}} \end{bmatrix} \quad {}^{e}\hat{\mathbf{r}}_{cs} = \begin{bmatrix} \frac{\tilde{x}+\tilde{d}_{sx}}{\left((\tilde{x}+\tilde{d}_{sx})^2+(\tilde{y}+\tilde{d}_{sy})^2+\tilde{z}^2\right)^{\frac{1}{2}}} \\ \frac{\tilde{y}+d_{sy}}{\left((\tilde{x}+\tilde{d}_{sx})^2+(\tilde{y}+\tilde{d}_{sy})^2+\tilde{z}^2\right)^{\frac{1}{2}}} \\ \frac{\tilde{z}}{\left((\tilde{x}+\tilde{d}_{sx})^2+(\tilde{y}+\tilde{d}_{sy})^2+\tilde{z}^2\right)^{\frac{1}{2}}} \end{bmatrix} \tag{5.37}
$$

$$
\zeta_1 = \cos\theta_{ecm} = \left[\frac{\tilde{x}^2 - \tilde{x}\tilde{d}_{mx} + \tilde{y}^2 + \tilde{z}^2}{(\tilde{x}^2 + \tilde{y}^2 + \tilde{z}^2)^{\frac{1}{2}} \left((\tilde{x} - \tilde{d}_{mx})^2 + \tilde{y}^2 + \tilde{z}^2\right)^{\frac{1}{2}}} \right] = h_1(\mathbf{x}) \tag{5.38}
$$

$$
\zeta_2 = \cos\theta_{ecs} = \left[\frac{\tilde{x}^2 - \tilde{x}\tilde{d}_{sx} + \tilde{y}^2 - \tilde{y}\tilde{d}_{sy} + \tilde{z}^2}{(\tilde{x}^2 + \tilde{y}^2 + \tilde{z}^2)^{\frac{1}{2}} \left((\tilde{x} - \tilde{d}_{sx})^2 + (\tilde{y} - \tilde{d}_{sy})^2 + \tilde{z}^2\right)^{\frac{1}{2}}} \right] = h_2(\mathbf{x}) \tag{5.39}
$$

$$
\zeta_3 = \cos\theta_{mcs} = \left[\frac{\tilde{x}^2 - \tilde{x}\tilde{d}_{mx} - \tilde{x}\tilde{d}_{sx} + \tilde{d}_{mx}\tilde{d}_{sx} + \tilde{y}^2 - \tilde{y}\tilde{d}_{sy} + \tilde{z}^2}{\left((\tilde{x} - \tilde{d}_{mx})^2 + \tilde{y}^2 + \tilde{z}^2\right)^{\frac{1}{2}} \left((\tilde{x} - \tilde{d}_{sx})^2 + (\tilde{y} - \tilde{d}_{sy})^2 + \tilde{z}^2\right)^{\frac{1}{2}}} \right] = h_3(\mathbf{x}) \tag{5.40}
$$

$$
\zeta_4 = \theta_e = 2\tan^{-1}\left(\frac{\tilde{d}_e}{\sqrt{\tilde{x}^2 + \tilde{y}^2 + \tilde{z}^2}} \right) = h_4(\mathbf{x}) \tag{5.41}
$$

$$
\zeta_5 = \theta_m = 2\tan^{-1}\left(\frac{\tilde{d}_m}{\sqrt{(\tilde{x} - \tilde{d}_{mx})^2 + \tilde{y}^2 + \tilde{z}^2}} \right) = h_5(\mathbf{x}) \tag{5.42}
$$

In this transformed coordinate system, it becomes clear that the measurements cannot disambiguate the spacecraft position $(\tilde{x}, \tilde{y}, \tilde{z})$ from the position symmetrically across the plane $(\tilde{x}, \tilde{y}, -\tilde{z})$. This is made apparent by the fact that the variable $\tilde{z}$ does not appear in any form other than its square, $\tilde{z}^2$. Thus, if the planets span $R2$, there will be two points in space associated with each point in time. It is worth noting that one additional reference vector would eliminate this ambiguity. In the vicinity of the Galilean moons, for instance, position determination would be entirely unambiguous in non-singular configurations.

In the event that the Earth, Sun, and Moon align, ${}^{e}Q^{E}$ loses rank and the ephemeral frame becomes ill-defined. $\tilde{d}_{mx}$ and $\tilde{d}_{sx}$ of eqns. 5.34-5.36 remain defined, but $\tilde{d}_{sy}$ reduces to the zero vector. In this case, the measurement model reduces to eqns. 5.43-5.47.

$$\zeta_1 = \cos\theta_{ecm} = \left[\frac{\tilde{x}^2 - \tilde{x}\tilde{d}_{mx} + \left[\tilde{y}^2 + \tilde{z}^2\right]}{\left(\tilde{x}^2 + [\tilde{y}^2 + \tilde{z}^2]\right)^{\frac{1}{2}}\left((\tilde{x} - \tilde{d}_{mx})^2 + [\tilde{y}^2 + \tilde{z}^2]\right)^{\frac{1}{2}}}\right] = h_1(\mathbf{x}) \tag{5.43}$$

$$\zeta_2 = \cos\theta_{ecs} = \left[\frac{\tilde{x}^2 - \tilde{x}\tilde{d}_{sx} + \left[\tilde{y}^2 + \tilde{z}^2\right]}{\left(\tilde{x}^2 + [\tilde{y}^2 + \tilde{z}^2]\right)^{\frac{1}{2}}\left((\tilde{x} - \tilde{d}_{sx})^2 + [\tilde{y}^2 + \tilde{z}^2]\right)^{\frac{1}{2}}}\right] = h_2(\mathbf{x}) \tag{5.44}$$

$$\zeta_3 = \cos\theta_{mcs} = \left[\frac{\tilde{x}^2 - \tilde{x}\tilde{d}_{mx} - \tilde{x}\tilde{d}_{sx} + \tilde{d}_{mx}\tilde{d}_{sx} + \left[\tilde{y}^2 + \tilde{z}^2\right]}{\left((\tilde{x} - \tilde{d}_{mx})^2 + [\tilde{y}^2 + \tilde{z}^2]\right)^{\frac{1}{2}}\left((\tilde{x} - \tilde{d}_{sx})^2 + [\tilde{y}^2 + \tilde{z}^2]\right)^{\frac{1}{2}}}\right] = h_3(\mathbf{x}) \tag{5.45}$$

$$\zeta_4 = \theta_e = 2\tan^{-1}\left(\frac{\tilde{d}_e}{\sqrt{\tilde{x}^2 + [\tilde{y}^2 + \tilde{z}^2]}}\right) = h_4(\mathbf{x}) \tag{5.46}$$

$$\zeta_5 = \theta_m = 2\tan^{-1}\left(\frac{\tilde{d}_m}{\sqrt{(\tilde{x} - \tilde{d}_{mx})^2 + [\tilde{y}^2 + \tilde{z}^2]}}\right) = h_5(\mathbf{x}) \tag{5.47}$$

While $\tilde{x}$ remains defined, $\tilde{y}$ and $\tilde{z}$ now only appear in the measurement equations as the coupled quantity $\left[\tilde{y}^2 + \tilde{z}^2\right]$. In this case, the measurement model is no longer able to discern between any two points that are equidistant from the vector from the Earth to the Moon.

5.9.3 Introduction of attitude knowledge

Suppose that the spacecraft knew its attitude. In this case, one would be able to measure the angle of separation between two celestial bodies in the range $[0, 2\pi]$ rather than simply the magnitude of separation in the range $[0, \pi]$. One way of representing all of this information is to add the sine of the angles of separation to the measurement model. Doing so reduces the number of possible angular separations for celestial configurations that span $R2$ from two to one. The sine is obtained by taking the square root (ambiguously signed) of one minus the square of the cosine of the angle. This augmented measurement model is given by eqns. 5.48-5.55.

$$\zeta_1 = \cos\theta_{ecm} = \left[\frac{\tilde{x}^2 - \tilde{x}\tilde{d}_{mx} + \tilde{y}^2 + \tilde{z}^2}{(\tilde{x}^2 + \tilde{y}^2 + \tilde{z}^2)^{\frac{1}{2}}\left((\tilde{x} - \tilde{d}_{mx})^2 + \tilde{y}^2 + \tilde{z}^2\right)^{\frac{1}{2}}}\right] = h_1(\mathbf{x}) \tag{5.48}$$

$$\zeta_2 = \cos\theta_{ecs} = \left[\frac{\tilde{x}^2 - \tilde{x}\tilde{d}_{sx} + \tilde{y}^2 - \tilde{y}\tilde{d}_{sy} + \tilde{z}^2}{(\tilde{x}^2 + \tilde{y}^2 + \tilde{z}^2)^{\frac{1}{2}}\left((\tilde{x} - \tilde{d}_{sx})^2 + (\tilde{y} - \tilde{d}_{sy})^2 + \tilde{z}^2\right)^{\frac{1}{2}}}\right] = h_2(\mathbf{x}) \tag{5.49}$$

$$\zeta_3 = \cos\theta_{mcs} = \left[\frac{\tilde{x}^2 - \tilde{x}\tilde{d}_{mx} - \tilde{x}\tilde{d}_{sx} + \tilde{d}_{mx}\tilde{d}_{sx} + \tilde{y}^2 - \tilde{y}\tilde{d}_{sy} + \tilde{z}^2}{\left((\tilde{x} - \tilde{d}_{mx})^2 + \tilde{y}^2 + \tilde{z}^2\right)^{\frac{1}{2}}\left((\tilde{x} - \tilde{d}_{sx})^2 + (\tilde{y} - \tilde{d}_{sy})^2 + \tilde{z}^2\right)^{\frac{1}{2}}}\right] = h_3(\mathbf{x}) \tag{5.50}$$

$$\zeta_4 = \theta_e = 2\tan^{-1}\left(\frac{\tilde{d}_e}{\sqrt{\tilde{x}^2 + \tilde{y}^2 + \tilde{z}^2}}\right) = h_4(\mathbf{x}) \tag{5.51}$$

$$\zeta_5 = \theta_m = 2\tan^{-1}\left(\frac{\tilde{d}_m}{\sqrt{(\tilde{x} - \tilde{d}_{mx})^2 + \tilde{y}^2 + \tilde{z}^2}}\right) = h_5(\mathbf{x}) \tag{5.52}$$

$$\zeta_6 = \sin\theta_{ecm} = \pm\sqrt{1 - \left[\frac{\tilde{x}^2 - \tilde{x}\tilde{d}_{mx} + \tilde{y}^2 + \tilde{z}^2}{(\tilde{x}^2 + \tilde{y}^2 + \tilde{z}^2)^{\frac{1}{2}}\left((\tilde{x} - \tilde{d}_{mx})^2 + \tilde{y}^2 + \tilde{z}^2\right)^{\frac{1}{2}}}\right]^2} = h_6(\mathbf{x}) \tag{5.53}$$

$$\zeta_7 = \sin\theta_{ecs} = \pm\sqrt{1 - \left[\frac{\tilde{x}^2 - \tilde{x}\tilde{d}_{sx} + \tilde{y}^2 - \tilde{y}\tilde{d}_{sy} + \tilde{z}^2}{(\tilde{x}^2 + \tilde{y}^2 + \tilde{z}^2)^{\frac{1}{2}}\left((\tilde{x} - \tilde{d}_{sx})^2 + (\tilde{y} - \tilde{d}_{sy})^2 + \tilde{z}^2\right)^{\frac{1}{2}}}\right]^2} = h_7(\mathbf{x}) \tag{5.54}$$

$$\zeta_8 = \sin\theta_{mcs} = \pm\sqrt{1 - \left[\frac{\tilde{x}^2 - \tilde{x}\tilde{d}_{mx} - \tilde{x}\tilde{d}_{sx} + \tilde{d}_{mx}\tilde{d}_{sx} + \tilde{y}^2 - \tilde{y}\tilde{d}_{sy} + \tilde{z}^2}{\left((\tilde{x} - \tilde{d}_{mx})^2 + \tilde{y}^2 + \tilde{z}^2\right)^{\frac{1}{2}}\left((\tilde{x} - \tilde{d}_{sx})^2 + (\tilde{y} - \tilde{d}_{sy})^2 + \tilde{z}^2\right)^{\frac{1}{2}}}\right]^2} = h_8(\mathbf{x}) \tag{5.55}$$

The quantity in the square root of eqns. 5.54-5.55 can be obtained without solving the attitude problem. The only information that is unavailable is the sign on the square root. Solving the attitude problem introduces this information to the system. In other words, the attitude information introduces binary information to the system: positive or negative on each square root.

This binary information introduces information about the sign of the coordinates. The signs of the $\tilde{x}$ and $\tilde{y}$ position coordinates are already known unambiguously given the cosine information; so, the attitude information introduces information about the sign of the $\tilde{z}$ coordinate in the ephemeral coordinates. That is to say, it reduces the number of possible locations from two to one.

CHAPTER 6

METHODS FOR UNPRECEDENTED DATASETS

6.1 Background and motivation

There is a systematic method for conducting science in space. Long before a spacecraft is designed or built, scientists carefully articulate the hypothesis that they intend to confirm or deny. With that hypothesis articulated, they decide on the variety and quantity of data required to investigate it. Engineers design every aspect of the spacecraft and the spacecraft mission around these data requirements. Scientists choose payload sensors, and engineers choose attitude sensors and actuators so that the spacecraft can point and slew with sufficient accuracy and speed for the payload sensors. Scientists choose the mission destination for the data that lives there, and engineers size the power system (and other subsystems) based on that location. The spacecraft telemetry and command subsystem is built around the bandwidth requirements for communicating data. The goal of a scientific mission in space is a dataset, and that dataset is generally very specific to the question being answered.

There are rare examples of spacecraft that create a dataset which contains answers to scientific questions that we did not think to ask. Nearly all of these examples have to do with remote sensing. Consider the Mars Reconnaissance Orbiter (MRO), for example, which is equipped with a camera. The dataset from that camera is sufficiently rich for scientists to mine it for answers to questions that they did not have in mind when the spacecraft launched. The data contains discoveries beyond which scientists and engineers designed the mission to investigate. Or, alternatively, the scientists and engineers designed the mission such that it would generate a dataset which contains surprises. A swarm of Monarchs generates a dataset that shares this property.

A swarm of Monarchs will gather a spatially distributed dataset which includes information from many different in-situ sensors, of the sort described in Chapter 2. Of course one can design a chipsat mission such that it generates a dataset which will answer a specific scientific question. However, the dataset that chipsats generate has a richness similar to that from the MRO. The information from all those sensors spread over a large area for a large amount of time will likely contain answers to questions that we did not think to ask. Finding those answers in the Monarch dataset requires a different set of techniques than those employed for the MRO, however.

It is no coincidence that we find surprises in MRO imagery data, since our brains have evolved for extracting information from images. The human brain is phenomenally good at image processing. Our brains have not evolved for extracting information from distributed measurements from accelerometers and other sensors like those described in Chapter 2. If there are patterns and insights to be found within the dataset generated by a distributed swarm of Monarchs, we require different sorts of techniques for finding those patterns and insights. This chapter describes one such technique.

This chapter describes a method for performing data prognostics using symbolic regression. Symbolic regression searches a space of analytical mathematical expressions for the one which matches a particular dataset, from no particular starting point. It is a method for finding correlations and relationships among sensor values in a dataset. Data prognostics uses pattern recognition in datasets to make predictions. So, data prognostics using symbolic regression is a method for generating analytic mathematical expressions from a dataset which will make predictions based on patterns discovered within that dataset.

This technique is entirely application agnostic. Each source of data is treated strictly as a sequence of numbers, with no connection to the physical quantities that they represent. For this reason, I am free to demonstrate the utility of this technique for Monarch data using a dataset which was not generated by Monarchs. NASA maintains a Prognostics Data Repository for exactly this purpose. This repository contains datasets for training and testing prognostics strategies.

This chapter treats the characterization of turbofan engine degradation as a particular application for the general technique. The proposed genetic program (GP) characterizes engine degradation, and then uses that characterization to both detect engine faults and predict the remaining lifetimes of engines after a fault. The genetic program exploits the fact that engine degradation manifests itself as changing correlations between sensor outputs. The NASA Prognostics Data Repository provides a training set in which 100 simulated engines are run to failure, and a test set in which a separate set of 100 simulated engines are shut off before they fail. The GP uses the training fleet of engines to identify the sensor relationships that indicate engine fault and predict remaining lifetime, and then observes the learned sensor relationships in the test fleet. The genetic program successfully detects the moment that the fault occurs for every engine in the test fleet and accurately predicts the remaining lifetime of the engines after the fault.

6.2 Problem Definition

The input to the system is sensor data from 100 engines. Each engine has 26 sensors that, for the training data set, record information until the engine fails. The output of the system is a single number for each engine, which corresponds to that engine's remaining lifetime. The program evolves functions of the form shown in eqn. 6.1.

$$L_{remaining} = f(sensor_i, sensor_j, sensor_k, \dots) \tag{6.1}$$

Where $L_{remaining}$ is remaining engine lifetime, which may be a function of any combination of some or all of the sensor outputs. The program is entirely system agnostic. Each sensor is treated purely as a source of data without any connection to the physical world.

6.3 System Architecture

6.3.1 Data

The data used for both training and testing comes from NASA's Prognostic Data Repository. Each dataset consists of multivariate time series, with each time series corresponding to a different sensor or operational setting. The data was produced by C-MAPSS simulation software, the industry standard for simulating transient effects in turbofan engine degradation. For each engine in the training dataset, the engine starts under normal operation, develops a fault, and runs to failure [35] [31]. In the test dataset, the engines all start under normal operation, develop a fault, but are *not* run to failure. This analysis uses dataset FD001, which is composed of engines of all the same type.

6.3.2 Structure

The problem is framed in a machine learning context. The genetic program evolves tree data structures, where each node is an operator and each leaf is either a constant or an array of sensor output. The operators are arranged in a dictionary, and they include:

- Arithmetic: Left leaf and right leaf are combined according to the arithmetic operator {+, -, *, /}. These are four separate operators.
- Trigonometry: Left leaf is multiplied by the result of acting one of the three main trigonometric functions on the right leaf. These are three separate operators.
- Exponentiation: Left leaf is multiplied by the exponential of the right leaf.
- Logarithm: Left leaf is multiplied by the natural logarithm of the right leaf.
- Noise: Left leaf is multiplied by the right leaf, which has been modified by additive Gaussian noise.
- Standard Deviation: Left leaf is multiplied by standard deviation of right leaf.
- Gradient: Left leaf is multiplied by gradient of right leaf.
- Second Gradient: Left leaf is multiplied by second gradient of right leaf.

- Window Operators: Left leaf is multiplied by one of the above 3 operators (standard deviation, gradient, or second gradient) acting only on the most recent 10 data points from the right leaf. These are three separate operators.

Sensor output is arranged in a separate dictionary, with each key corresponding to a separate engine and the value of each key being a list of 26 lists, each list corresponding to a different sensor. The GP evolves functions with the variables represented as keys of these dictionaries. When a function is evaluated for fitness, it is evaluated on every engine in the training fleet.

At a high level, the program works with objects of two classes. The "person" class is a tree structure. An object of the person class has the ability to generate predictions for a particular engine, to evaluate its own depth, to perform crossover with another object of the person class, to mutate itself, to determine if it is dominant to another object of the person class, and to evaluate the accuracy of its predictions are on a particular engine.

The "population" class is, fundamentally, a list of objects of the person class. The population has the ability to perform operations on the population as a whole. It may add a person to the population, gather the traits of each member of the population (uniqueness, age, predictability, etc.), rank the population according to their traits, select a parent population from the entire population, and breed the members of the parent population to replenish its original size. These functions are combined into broad methods that attempt to maximize the fitness of the entire population.

As input, these broad optimization methods take mutation rate, population size, selection pressure, elitism pressure, and maximum allowable depth for the tree structures that compose the population.

6.4 Method

The genetic program operates on a population of solutions. For each generation, the population goes through five distinct steps. These steps include:

1. Ranking the population according to the established fitness criteria
2. Selecting a subset of the population that will survive/breed into the next generation.
3. Performing crossover among the surviving parent population to create a population of children.
4. Performing mutation on the child population.
5. Adding the child population to the parent population to replenish the population to its original size.
6. Returning to step 1, and repeating for many generations.

Each of the above steps is described in sections 6.4.1-6.4.5.

6.4.1 Rank the Population

Elitist multi objective optimization on a Pareto front is used to maintain genetic diversity in the population. Solutions are optimized along four dimensions:

- Age: Solutions that have been in the population for a shorter amount of time are more fit (in the age dimension) than solutions that have been evolving for a long time. This helps maintain diversity in the population by giving solutions with more potential to evolve an advantage over those that have become stagnant. During crossover, the child adopts the age of its oldest parent.
- Mean Prediction Error: Solutions that have a lower average prediction error (measured across all engines in the training fleet) are more fit than solutions that have higher average error.
- Uniqueness: Solutions that have better predictability on particular engines for which other members of the population are unable to predict are more fit in the uniqueness dimension. This helps maintain diversity.
- Worst Prediction: Solutions with lower error on their worst prediction are more fit than solutions with higher error on their worst prediction (even if they have a lower mean error over all engines). This prevents the algorithm from getting stuck at the mean remaining lifetime of all engines in the fleet.

The solutions that compose the population are placed on a series of Pareto fronts according to the NSGA non-dominated sort described by Seshadri in [10]. In brief, the population is sorted in the following way:

1. Initialize the number of individuals that dominate each member of the population to 0 and the members of the population that each member of the population dominates to an empty list.

2. For each member of the population p, loop through every other member of the population q. If p dominates q, then add q to the list of solutions that p dominates. If q dominates p, then increment the domination counter by 1.
3. If the domination counter equals 0 for a particular solution, then add that solution to the leading Pareto front.
4. Initialize a second Pareto front as an empty list.
5. Decrement the domination counter for every solution by 1. If the domination counter for any of the solutions becomes 0, add it to the second Pareto front (because this indicates that it was only dominated by one of the individuals in the first Pareto front).
6. Return to step 3 and continue until all solutions have been placed in a Pareto front

Domination is defined as being equally or more fit along each of the four dimensions of fitness (age, mean prediction error, uniqueness, and worst prediction), and more fit along any one of the dimensions. Once each solution is in a front, the algorithm moves on to the selection process [36]. See Fig. 6.1.

6.4.2 Selection

The algorithm uses elitism in that any solution that lives in the first Pareto front is guaranteed to survive into the next generation. All members of the population have some probability of surviving, but the solutions that occupy the more fit Pareto fronts have a higher probability of making it to the next generation.

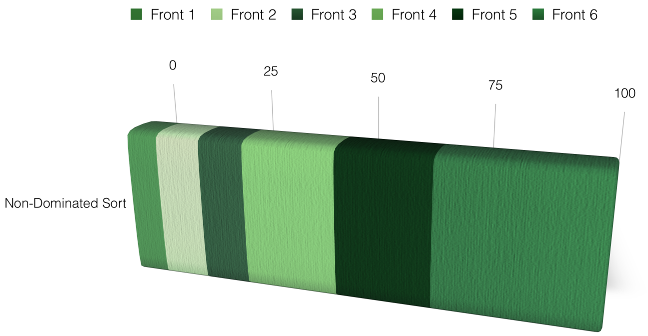


Figure 6.1: Visualization of NSGA non-dominated sort. The relative sizes of the Pareto fronts change from generation to generation.

The algorithm works with a selection pressure of 0.4. This is an empiracally determined parameter that can be tuned for different applications. Tighter selection pressures led to homogeneity in the population. After the members of the leading Pareto front are added to the surviving population, 80 percent of the remaining survivors are picked from the top 60 percent of the old population, and 20 percent of the remaining population are randomly generated *new* solutions. It is rare that any of these solutions have better predictive abilities than the older solutions that have been evolving for longer, but because they are younger than the rest of the solutions a few are able to survive to the next generation. This helps maintain diversity in the population. See Fig. 6.2.

6.4.3 Crossover

In order to replenish the population to its original size, crossover is performed to produce child solutions from the surviving parent solutions. The mother is preferentially chosen to be among the elite members of the population (ranking

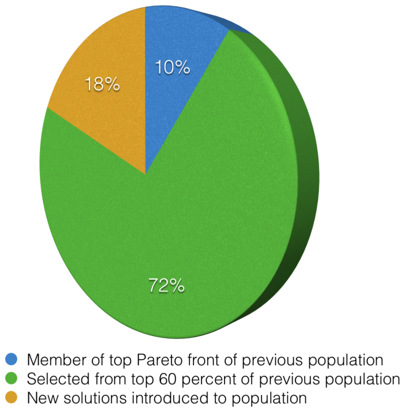


Figure 6.2: Constituent members of parent population after selection. The relative sizes change for each generation, depending on the size of the leading Pareto front.

somewhere in the top 20 solutions), but the father is randomly selected from the parent population. This practice leads to useful diversity in the children. When both parents are randomly selected from the surviving population, they produced very diverse children, none of whom are particularly fit. When both parents are selected from the elite members of the surviving population, the population loses diversity.

Because the members of the population are functions that are represented as a tree structure, crossover amounts to swapping branches between parents to produce a new child that has traits of both mother and father [39]. See Fig. 6.3. The depth of crossover is randomly determined for each parent every time crossover occurs.

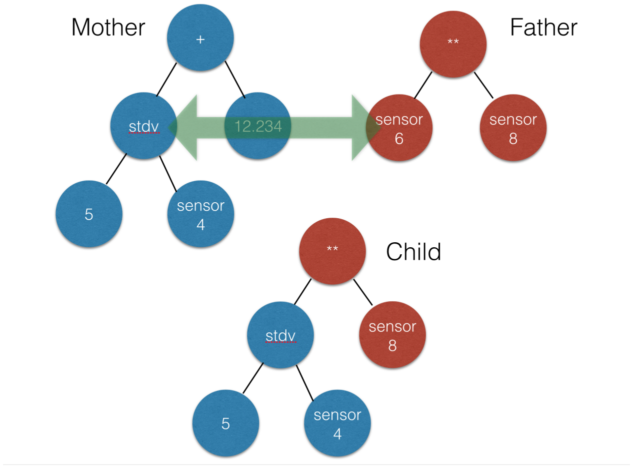


Figure 6.3: Two parent solutions producing a child solution by swapping branches.

6.4.4 Mutation

After crossover creates a new child, it is mutated before being placed into the population. While crossover allows the population to strategically explore new parts of the optimization landscape, mutations are small variations that allow solutions to climb to the nearest peak. The mutations are constructed such that they are not disruptive to the good genes in the genome.

The mutation rate cools as the algorithm runs, until it reaches the 100th generation. At this point the mutation rate heats back up before cooling off again over the course of the next 100 generations. The algorithm has the ability to mutate the value of a constant, change a constant to the output of sensor, or change a sensor variable to a constant. The algorithm may not mutate operators, because this sort of mutation is often extremely disruptive to the existing genes.

6.4.5 Replacement into Population

In order to maintain diversity, the children are replaced into the population using a form of deterministic crowding. After a child is created, the program checks whether it dominates either of its parents. If the child dominates a parent, then it replaces the parent.

Because the algorithm uses a cooling (and periodically heated) mutation rate, there are a few generations for every 100 for which mutations are extremely rare. If crossover occurs between two solutions that are not very deep (contain very few branches), it is possible for the child to be identical to one of its parents. In order to prevent duplicate solutions in the population, every child is compared with every member of the population before being injected into it. If a child does not replace a parent, and it is not identical to a solution that already lives in the population, then it enters the population.

6.5 Experimental Evaluation

6.5.1 Methodology

The fitness of each potential solution is judged in the training dataset alone, according to the criteria discussed in section 6.4.1. For sake of making accurate predictions, however, the important variable is prediction error. The other dimensions of the Pareto front exist to provide diversity and to help make improvements along this one dimension of actual concern. The constituent members of each population compete in the training set, and the performance of the

GP is baselined against that of the hill climber and random search in the training data set. Every variable that composes a dimension of the Pareto front is a dependent variable, and the sensor outputs form the independent variables. The performance data, however, is *mean prediction error in the training data set*. The solutions that better characterize the training dataset are considered better solutions than those that cannot make as accurate predictions. Comparisons are made between populations through performance curves that show predicability plotted against the number of evaluations.

Performance on the test dataset is evaluated by applying the function generated on the training dataset and comparing the predictions against the true values. These predictions can be judged on an engine-by-engine basis by plotting the true remaining lifetime of the engine (for every moment in time) along with the prediction generated by the function (for every moment in time). These are the sorts of plots that Moghaddass and Zuo used to make empirical judgements of the performance of their algorithm [31].

6.5.2 Results

The Proof of Concept

In the engine prognostics problem, the GP is asked to return a function of sensor output that returns the remaining lifetime of the engine. *A priori*, however, there is no guarantee that such a solution exists. For that reason, it is important to create a toy problem that verifies the GP will be able to find such a solution, if it does exist. This toy problem should be one where the solution exists and is known. In other words, there exists a combination of sensors in the toy dataset

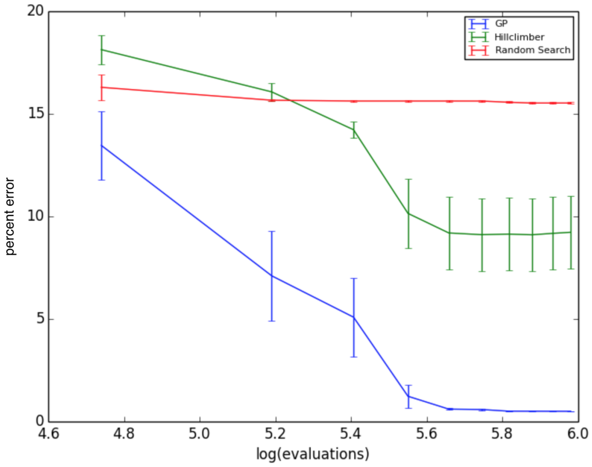


Figure 6.4: *Proof of Concept*: Percent error between estimated and true remaining engine lifetimes in the training data set. Shows GP finding the optimal solution constructed for the toy problem.

that will return remaining engine lifetime with absolute precision.

This can be accomplished by giving the GP, hill climber, and random search access to a clock on each engine. The clock is represented as another sensor, the output for which starts at 0 and linearly increases until the engine fails. The remaining lifetime of the engine, therefore, is just a scaling of this sensor's output. The optimal solution should include just the output of the clock and a scaling. Fig. 6.4 shows that the GP finds the optimal solution in the toy dataset, and it does so faster than either the random search or the hill climber. This suggests that, if such a solution exists in the real dataset, the GP will be able to find it.

Engine Prognostics

By removing the clock sensor from each engine, all optimization techniques are forced to search for a function that yields remaining engine lifetime based *solely* on sensor output. Fig. 6.5 shows that the GP generally finds a better absolute solution than the hill climber, though not always (as indicated by the overlapping error bars). The best solution that the GP found is shown in eqns. 6.2.

$$Lifetime = -119.95 \sin\left(sensor23 - sensor6\right) \tag{6.2}$$

Fig. 6.6 shows the estimated vs. actual remaining engine lifetime of the first 20 engines in the test data set. The GP not only captures the moment that the fault occurs in each engine (represented by the elbows in each curve), but also creates an empirically close estimate for each engines remaining lifetime after

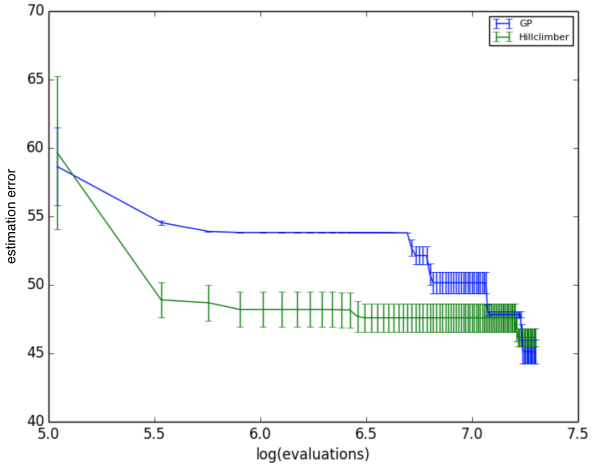


Figure 6.5: Difference between estimated and true remaining engine lifetimes in the training data set. Y axis is average difference between true and estimated number of remaining engine life cycles.

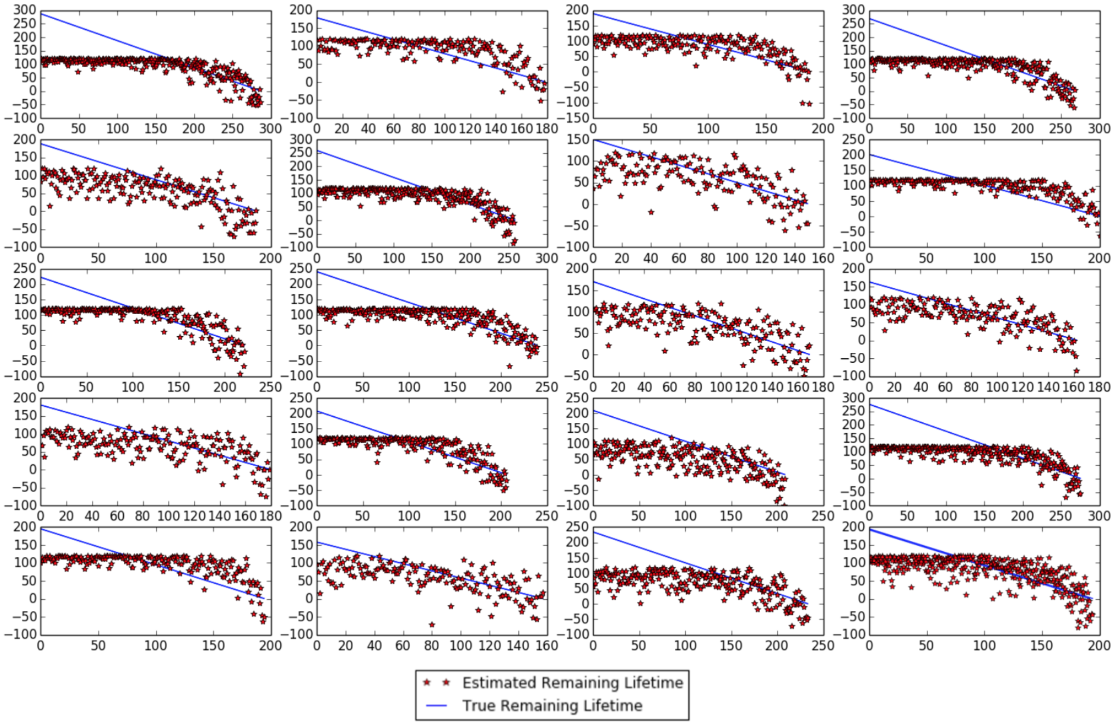


Figure 6.6: Estimated vs. actual remaining engine lifetime.

the fault.

The genetic program described in this chapter is able to determine the moment of failure for each engine tested. Although the GP's advantage over the hill climber lessens in the engine prognostics problem (vs. the toy problem), the GP generally does a better job characterizing engine failure in both situations. The correlation that the GP isolates as being indicative of engine failure is the difference between the outputs of sensors 23 and 6. It uses this information to determine when each engine fails, and to accurately predict remaining lifetime after failure.

CHAPTER 7
CONCLUSION

The Monarch applies biological principles for mission assurance to space exploration and consequently is the first spacecraft to trade high quantity for low mission risk. By taking a statistical approach to mission assurance and devaluing the importance of any particular spacecraft, Monarchs open the door to a new paradigm in space access and exploration. They are not small versions of large spacecraft, and they do not replace large spacecraft. Instead, Monarchs have an entirely new and unique set of use cases. They enable distributed, in-situ sensing, which will provide scientific datasets of an unprecedented variety. These datasets enable science of an unprecedented variety. As a consequence of their size and quantity, Monarchs can perform entry, descent, and landing missions that would be far too risky for conventional spacecraft to attempt. And, perhaps just as significantly, Monarchs reduce the cost of access to space by orders of magnitude. Because they can be carried to orbit by the hundreds or thousands, the launch costs may be divided among many hundreds or thousands of Monarchs. The result is that space is no longer only accessible to governments, large companies, and universities, but also to high school classrooms and hobbyists. The Monarch is the greatest force for the democratization of space that has ever existed.

CHAPTER 8

APPENDIX

8.1 Extensions for digital agriculture

Monarchs provide remote delivery of spatially distributed in-situ data over an extended period of time. This core capability has a number of applications in space, many of which have been discussed in this dissertation, but it also has applications on Earth. I sought out a few of these terrestrial applications for the Monarchs in my final year of graduate work. I wanted to test the Monarchs in uncontrolled environments to prove feasibility of the space missions discussed in earlier sections, but, as long as I was gathering datasets, I wanted for those datasets to also be valuable for other people. In-situ data exfiltration from vineyards and dairy calves met these criteria. Each provides a challenging environment for the Monarchs and utility for the data that the Monarchs produce.

8.1.1 Vineyards

The Problem

Fruit growers in cool-climate regions lose yield each season to diseases, pests, and frost/cold damage. In order to mitigate risk of fruit loss to diseases and pests, farm managers apply chemical sprays throughout the growing season. In the case of vineyards and orchards, managers apply chemical sprays 12-15 times each season. The materials and labor associated with each of these sprays is expensive. At $700/acre, these sprays represent one of the largest annual

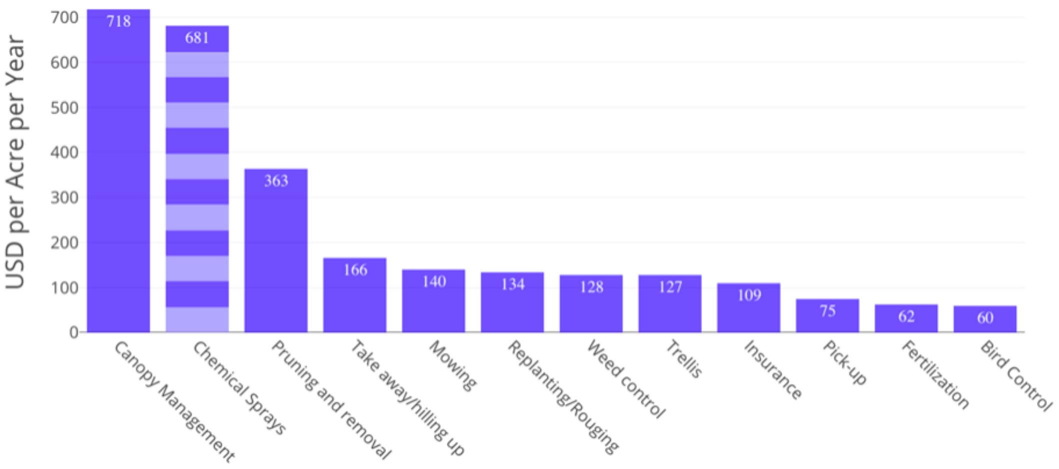


Figure 8.1: Annual variable costs for production of v. vinifera grapes in the Finger Lakes region.

variable costs for cool-climate vineyards and orchards, as shown in Fig. 8.1. [51]

With sufficient information, a farmer could safely eliminate several sprays each season. Disease forecasting models exist for most major fruit diseases, providing guidance on when to spray the crop. However, many growers do not use the disease forecasting models due to their lack of precision. The alternative to using the forecasting models is to spray on a calendar basis, which is commonly adopted in the production of wine grapes, where growers will spray every 7-10 days depending on weather and disease pressure. The high cost of each spray is considered lower than the potential cost of losing yield.

Expensive weather stations ($2,000) are used to inform the disease forecast models. Some growers will purchase their own but many will use a station that could be several miles away, providing inaccurate climate data for their site. The data from these weather stations is used to calculate disease and pest pressure.

This strategy is intended to allow farm managers to spray only when necessary, but in practice these models are often not used.

Regardless of whether a farm has its own weather station or is using one many miles away, these weather stations report environmental conditions in the air at the station height, but not the actual microclimatic conditions inside the vine canopies where the fruit are located. In cool-climate regions, this is not a good approximation. The leaves trap air and moisture near the fruit, creating optimal conditions for disease development that are not reflected in the weather station data. Because the farm managers have no means of knowing the conditions at the fruit, they must over-spray to mitigate risk of fruit loss. These microclimates can be significant enough for substantially different disease and pest forecasts inside the canopy vs. outside.

Furthermore, the conditions within the vineyard vary substantially with terrain. Cool air tends to flow like a viscous fluid, settling in the low points of the land. The assumption that environmental conditions across an entire farm are the same is also an inaccurate one and leads to increased risk of yield loss to both disease and frost/cold. Because fruit farmers do not know the conditions within vine canopies (next to the fruit), and because they do not know how conditions vary across their land, they must over-spray in order to be confident that the risk of fruit loss is mitigated. These sprays represent the biggest annual variable cost for vineyards. Lack of distributed environmental measurements also puts farmers at an increased risk of frost damage, which can be mitigated with turbines and heaters.

A Solution: Monarchs

With knowledge of the conditions within vine canopies, and with knowledge of how conditions vary across a farm, farm managers may apply chemical sprays and take preventative action against frost only when and where the conditions demand it. Such knowledge would prevent spending money on material and labor associated with unnecessary sprays, as well as preventing under-spraying at the places in the farm where conditions create exceptionally high risk of cold/frost, disease, or pest damage. In order to provide this knowledge, I developed a modified version of the Monarch spacecraft presented in earlier sections of this dissertation. The modified Monarch includes some extra, vineyard-relevant sensors (a relative humidity sensor and a temperature sensor), lower-cost solar cells, and a supercapacitor for energy storage. See Fig. 8.2.

Monarchs are deployed within leaf canopies across a farm, where they use their sensors to take measurements of the local environment. Each then uses its processor and radio to communicate those measurements both to other Monarchs and to an onsite receiver station. The onsite receiver station has an inter-

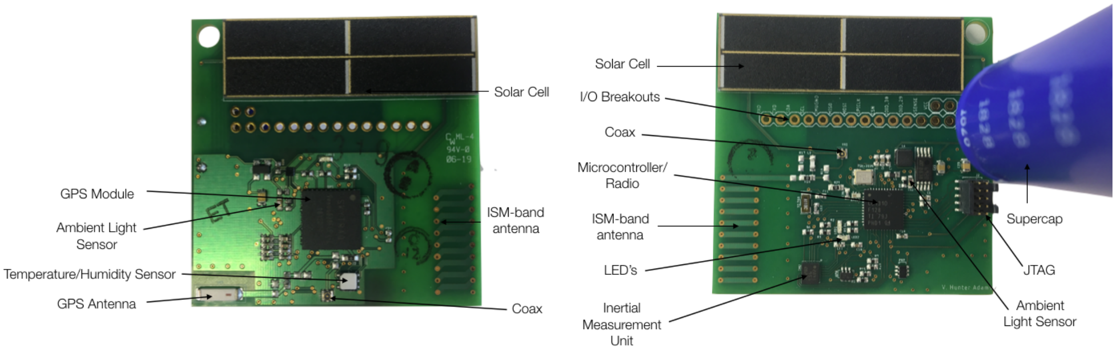


Figure 8.2: Agricultural version of the Monarch.

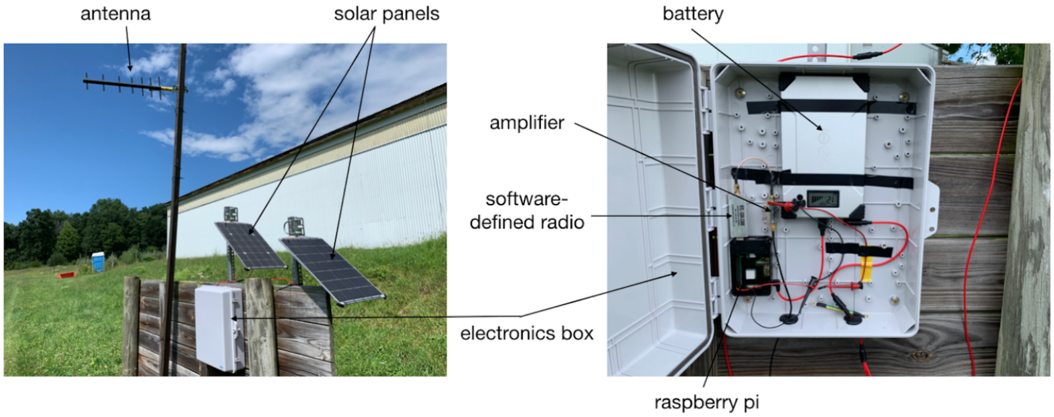


Figure 8.3: Receiver station for vineyard-deployed Monarchs, components labeled.

net connection. Data can be radioed a quarter-mile, line-of-sight, from each Monarch to the receiver station. The receiver station then communicates the data that it receives over the internet to a remote server. Once the communicated data is online, it may be viewed by the user via a website.

The receiver station is built from commercially available electronics. Radio signals from the Monarchs are gathered by an antenna. These signals are amplified by a low-noise amplifier, and then accumulated by an RTL-SDR software-defined radio. This software-defined radio is plugged in, via USB, to a Raspberry Pi computer with an internet connection. The Raspberry Pi performs the necessary realtime signal processing to convert the raw I/Q data from the software-defined radio to packets from the deployed Monarchs. It then sends these packets over the internet to a remote server. The receiver station can either be powered directly from a wall outlet, or via solar cells and a battery, as shown in Fig. 8.3.

Experiment and Data

I conducted two experiments. These experiments determined the extent to which Monarchs could measure microclimates within leaf canopies. The first experimental site was Anthony Road Winery in Penn Yan, NY. At this site,

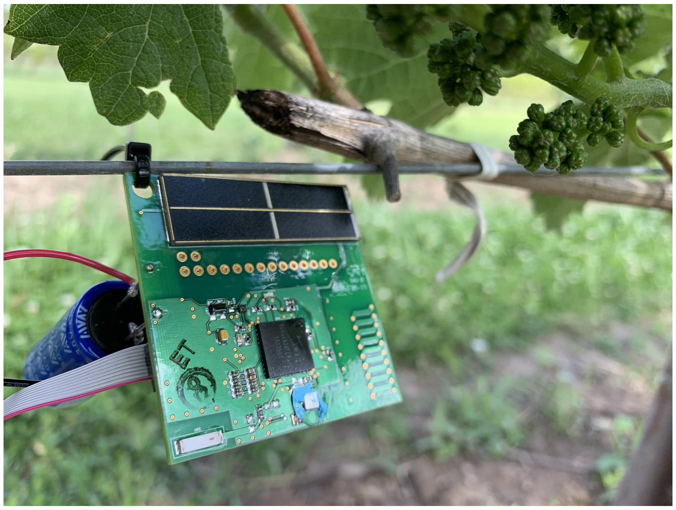


Figure 8.4: Monarch deployed without sunshade at Anthony Road Winery, Penn Yan, NY.

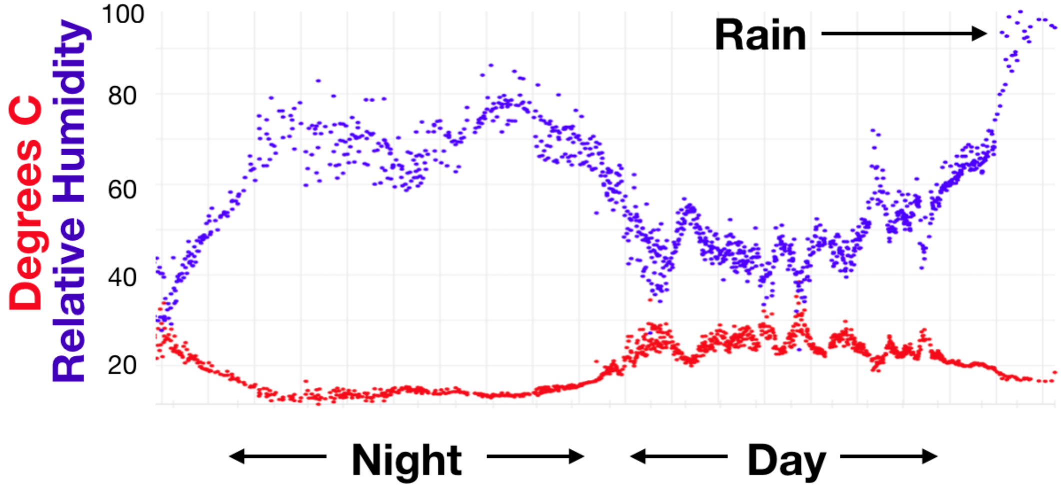


Figure 8.5: 24 hours of data from Anthony Road Winery. June 18-19, 2019.

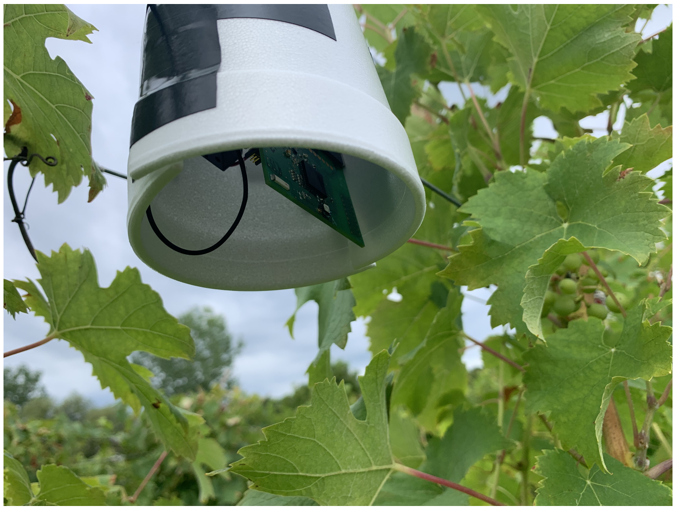


Figure 8.6: Monarch underneath a sunshade in Cornell's Lansing, NY vineyard.

20 Monarchs were deployed without sunshades within grape leaf canopies, as shown in Fig. 8.4. These Monarchs reported in-situ measurements of tempera-

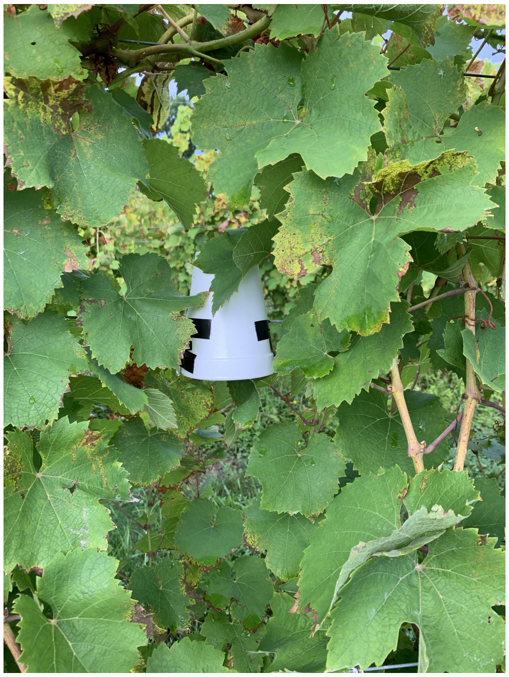


Figure 8.7: Monarch deployed in grapevine canopy in Cornell's Lansing, NY vineyard.

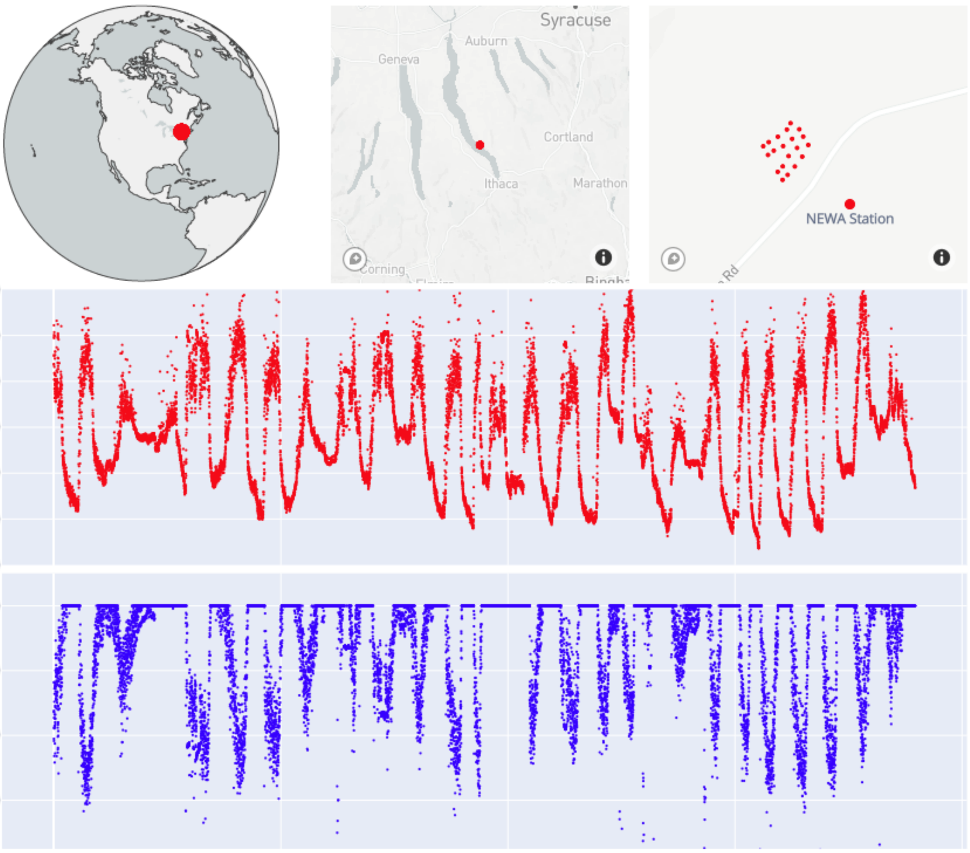


Figure 8.8: 4 weeks of temperature/humidity data from 20 Monarchs in Cornell's research vineyard in Lansing, NY.

ture and relative humidity (along with their GPS locations) every 5 minutes for 24 hours (June 18-19, 2019). Fig. 8.5 shows all of the data collected over this period.

The second site was the Cornell research vineyard in Lansing, NY, where 20 Monarchs were deployed with sunshades for a 4-week period of time (August 22, 2019 - September 24, 2019). Fig. 8.6 shows one of these Monarchs under a sunshade, and Fig. 8.7 shows the Monarch and sunshade within the leaf canopy. Each of these Monarchs transmitted in-situ measurements of temperature and relative humidity (along with their GPS coordinates) to the onsite receiver station. Fig. 8.8 shows all of the data collected over this period.

Analysis and Conclusions

Some aspects of these experiments worked well, and some aspects did not. The rainstorm indicated by the spike in humidity at the end of the Anthony Road dataset (Fig. 8.5) destroyed all of the Monarchs in that deployment. In the subsequent deployment, I added a sunshade which protected the Monarchs from

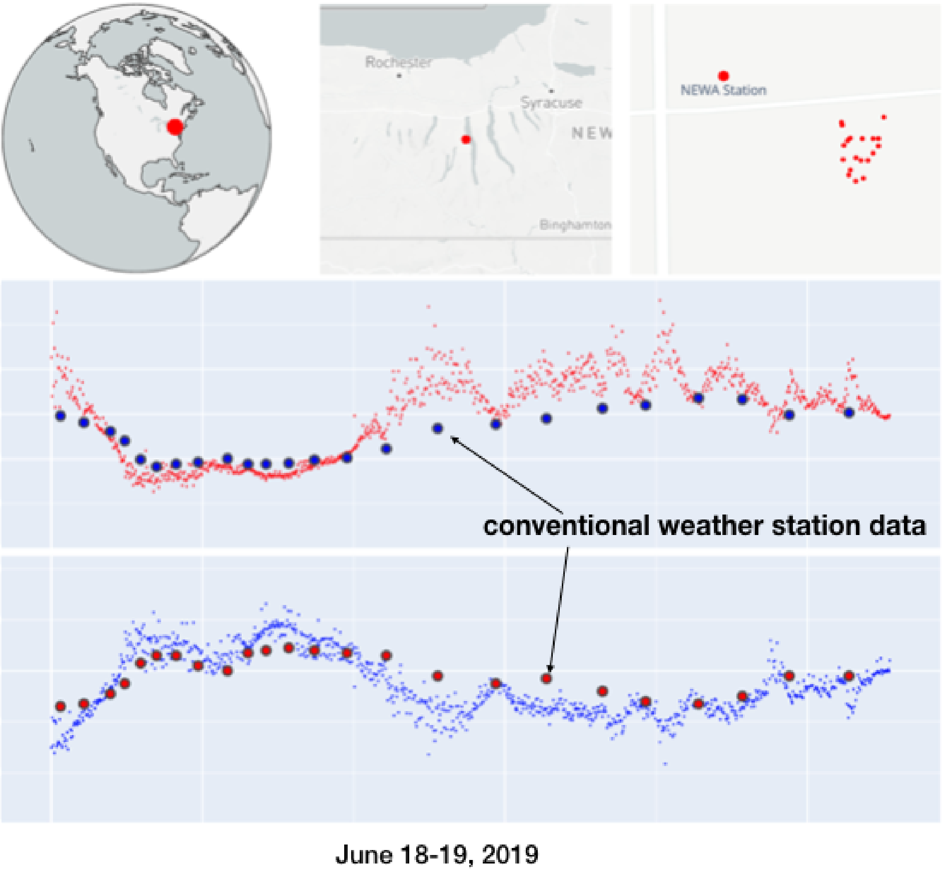


Figure 8.9: 20 hours of data from Anthony Road Winery, overlayed data from conventional weather station. June 18-19, 2019.

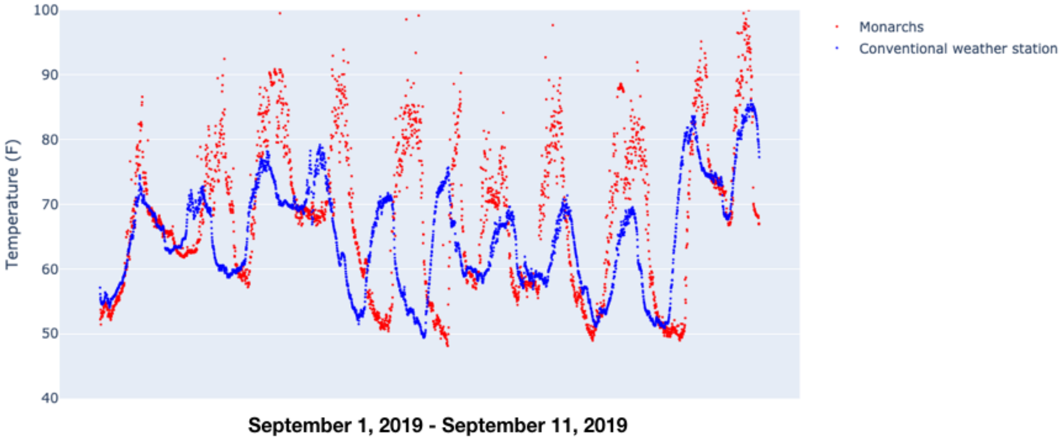


Figure 8.10: Data from Lansing vineyard overlayed with conventional weather station data, 9/1/2019 - 9/11/2019.

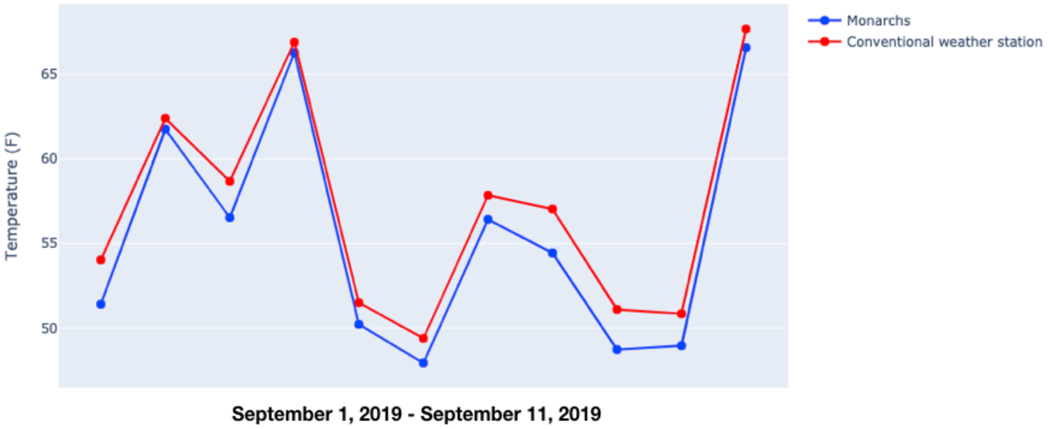


Figure 8.11: Comparison of overnight lows, Monarchs vs. conventional weather station, Lansing vineyard, 9/1/2019 - 9/11/2019.

running water, as shown in Figs. 8.6 and 8.7. With this addition, all Monarchs survived all four weeks in the Lansing vineyard, as evidenced by the dataset shown in Fig. 8.8. The second motivation for the addition of the sunshade is shown in Fig. 8.9, which shows the same dataset, with data from the onsite

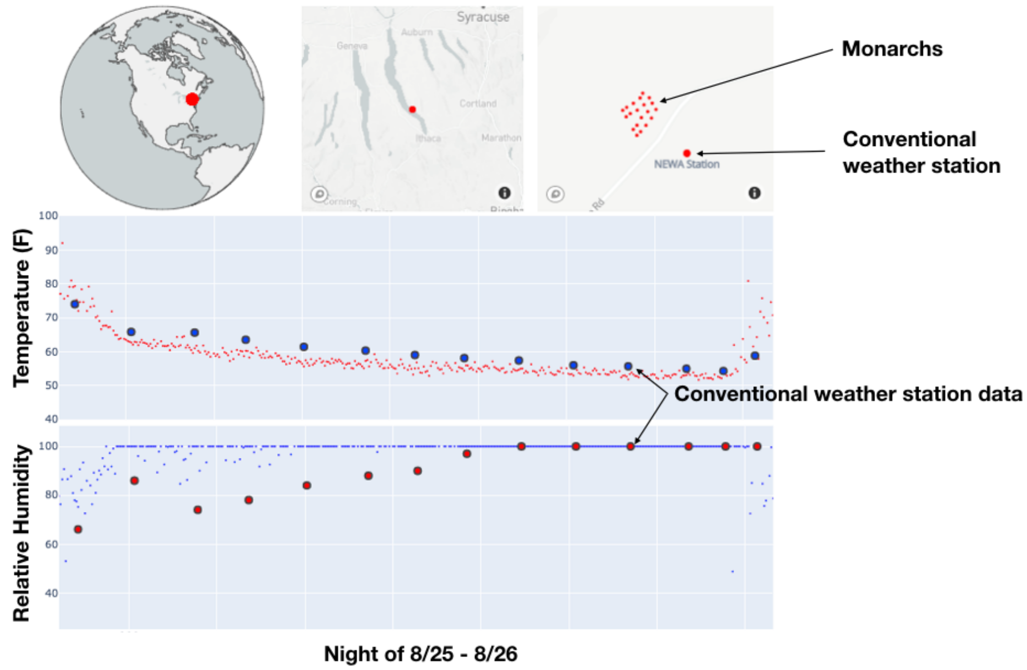


Figure 8.12: August 25-26, 2019 overnight measurements from Monarchs and conventional weather station.

conventional weather station overlayed. June 19, 2019 was a partly cloudy day. It can be seen that, when the sun was out, the temperatures as reported by the Monarchs were artificially high by as much as 10 degrees Farenheit. During the cloudy parts of the day, there is precise agreement between the Monarch data and the conventional weather station data. This indicates that the sun was warming the Monarchs, leading to artificially high temperatures. Overnight and during cloudy parts of the day, however, this is not a problem.

In the 4-week Lansing deployment, I added the sunshade to all Monarchs as shown in Fig. 8.6. Poor ventilation still led to temperature measurements that were too high during sunny afternoons. As with the Anthony Road deployment, the overnight temperature readings for which the sun was not an issue showed much better reliability. This is evidenced by Fig. 8.10. Because of artificial sun heating, the only fair comparisons that can be made between the Monarchs and the conventional weather stations are those which were gathered at night. Fig. 8.11 shows a comparison of overnight lows between September 1 and September 11, 2019 in the Lansing vineyard. The Monarchs are indeed measuring a microclimate within the leaf canopies. Fig. 8.12 compares overnight measurements on August 25-26, 2019 in the Lansing vineyard, showing significant climate variation in vs. out of the canopy. Future deployments will use a Stevenson Screen enclosure to prevent high daytime temperature measurements.

8.1.2 Dairy Calves

Dairy calves provided another challenging environment for stress-testing the Monarchs. In collaboration with Francisco Yeal-Lepes of the veterinary school at Cornell, I attached 10 Monarchs to collars which were placed on 1-5 day old dairy calves at Sunnyside Farms in Scipio Center, NY for 3 weeks. Fig 8.13 shows a calf wearing a Monarch.

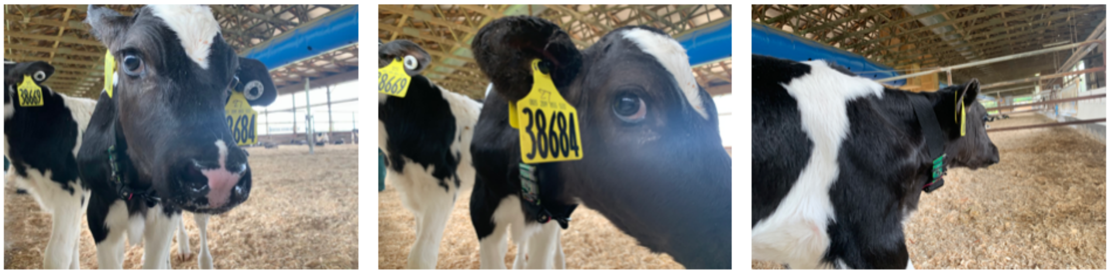


Figure 8.13: Dairy calf wearing a Monarch

Experimental Goals and Results

The experimental goals associated with the dairy calves were simpler than those associated with the vineyards. The suite of sensors with which the Monarchs were equipped was not ideal for use with the calves. To gather a more valuable dataset, a calf-specific Monarch will be constructed that includes some addi-

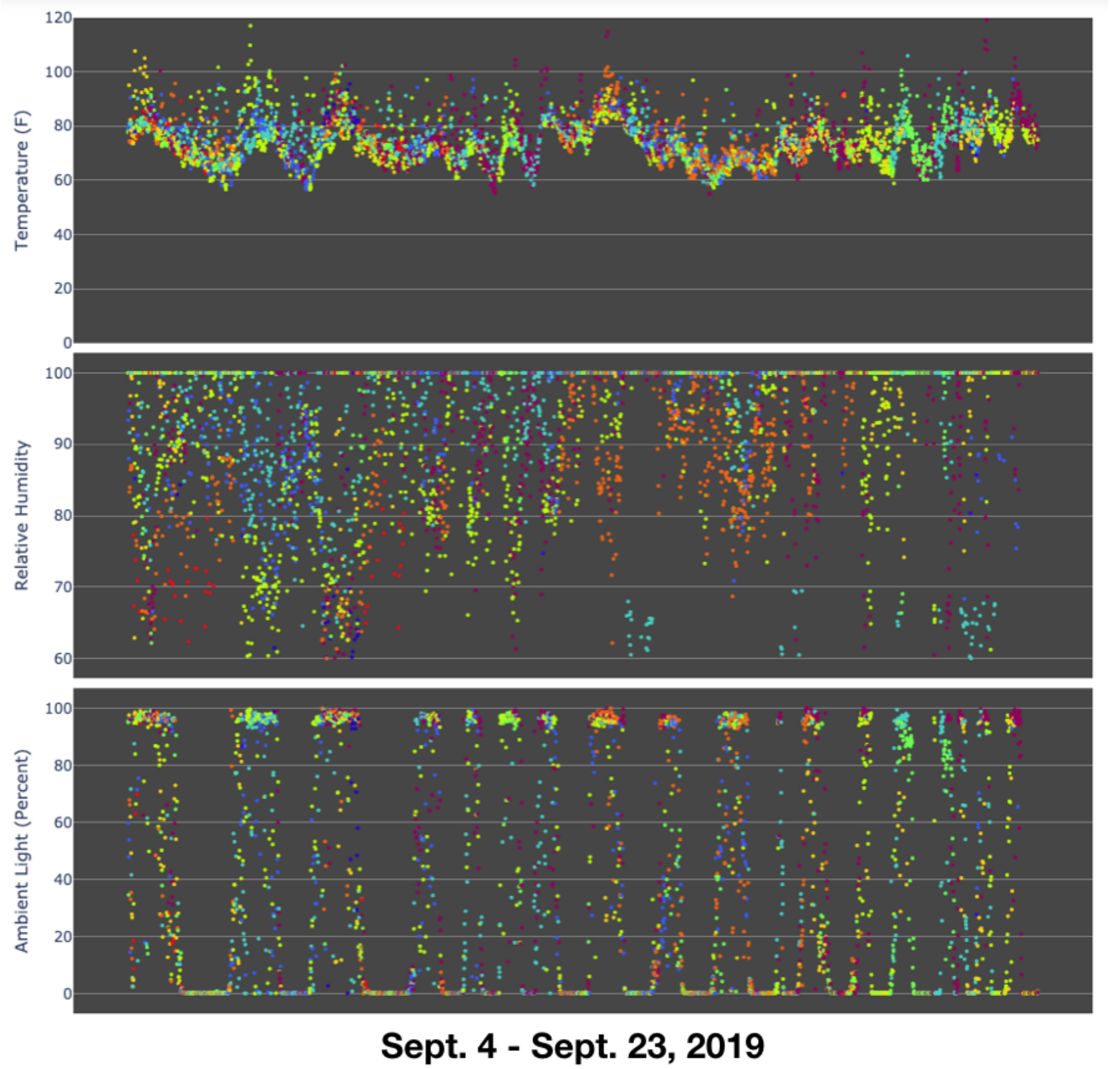


Figure 8.14: Temperature, humidity, and ambient light measurements from calves, individually colorized

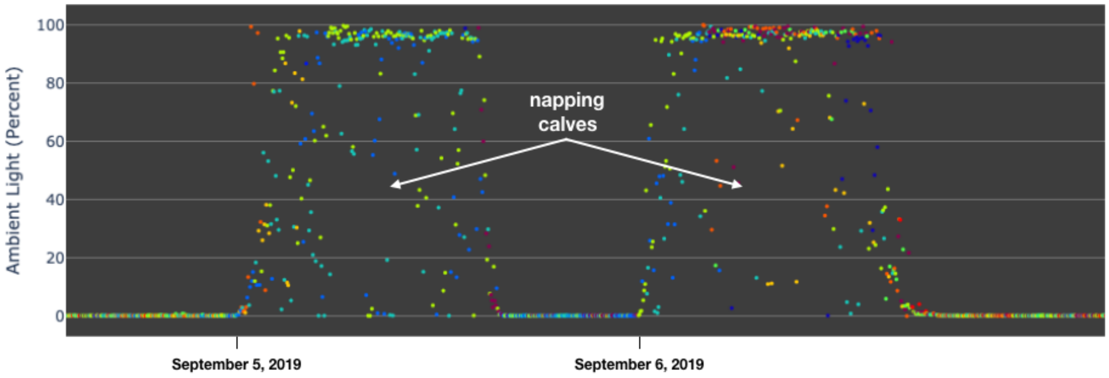


Figure 8.15: Two days of ambient light measurements, napping calves indicated.

tional sensors (like an ammonia sensor) and omits some irrelevant ones. The goal of this experiment was to build a case for a grant which would fund a calf-specific Monarch. I aimed to show that the Monarchs could continually transmit over a long enough period of time for an experiment to take place, that the calves would be unbothered by the collars and sensors, and to learn anything that that I could from the available sensors that might inform the calf-specific board. In this respect, the experiment was a success. Fig. 8.14 shows the temperature, humidity, and ambient light measurements. The measurements from each calf are individually colorized.

From these data I learned some important things for a future calf deployment. Because the Monarchs are mounted on the outside of the collars, as shown in Fig. 8.13, the temperature measurements are a difficult-to-understand mix of ambient and skin temperatures. In subsequent deployments, the temperature sensor will be mounted against the skin, which will give better measurements of skin temperature.

I also learned the utility of the ambient light sensor for this particular appli-

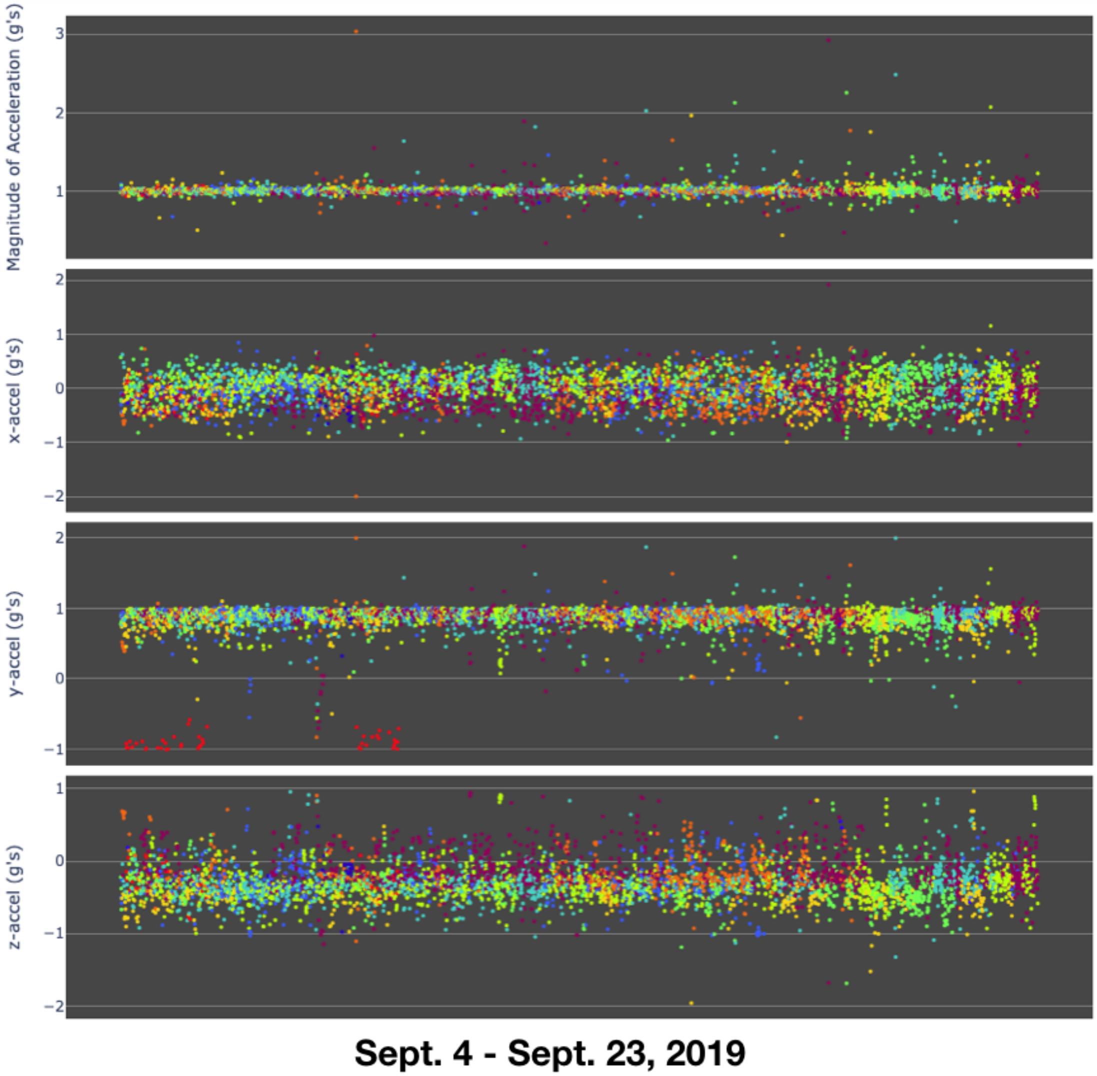


Figure 8.16: Accelerometer measurements from calves, each calf individually colorized.

cation. In general, the light sensor saturates during daylight hours, and drops to zero overnight. However, a number of measurements can be seen each day for which the ambient light detected is significanly lower than saturated. These measurements seem to be associated with napping calves, which tuck their chins into their forearms and prevent light from reaching the ambient light sensor. Fig. 8.15 shows these datapoints within a two-day subset of the full dataset.

The final lesson learned for subsequent calf experiments is the necessity for much more frequent measurements in order to detect activity levels. Each of the Monarchs in this deployment reported a set of measurements every five minutes. The accelerometer data, Fig. 8.16, shows that this is insufficiently frequent for capturing calf activity. In subsequent cattle deployments, Monarchs will report IMU data with a much greater frequency.

BIBLIOGRAPHY

[1] Spacex non-geostationary satellite system attachment a, technical information to supplement schedule s. 2016.

[2] Van Adams and Mason Peck. Chipsats: R-selected spacecraft. 2019.

[3] Van H Adams and Mason A Peck. A probabilistic network formulation for satellite swarm communications. In *2018 AIAA Information Systems-AIAA Infotech@ Aerospace*, page 1802. 2018.

[4] Elwood Agasid, R Burton, R Carlino, G Defouw, A Dono Perez, AG Karacahoglu, B Klamm, A Rademacher, J Schalkwyck, R Shimmin, et al. Small spacecraft technology state of the art. *NASA, Ames Research Center, Mission Design Division Rept. NASA/TP-2015-216648/REV1, Moffett Field, CA*, 2015.

[5] Justin A Atchison and Mason A Peck. Length scaling in spacecraft dynamics. *Journal of guidance, control, and dynamics*, 34(1):231–246, 2011.

[6] Claude Berrou, Alain Glavieux, and Punya Thitimajshima. Near shannon limit error-correcting coding and decoding: Turbo-codes. 1. In *Communications, 1993. ICC'93 Geneva. Technical Program, Conference Record, IEEE International Conference on*, volume 2, pages 1064–1070. IEEE, 1993.

[7] Dimitri P Bertsekas, Dimitri P Bertsekas, Dimitri P Bertsekas, and Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA, 1995.

[8] Shyam Bhaskaran, S Desai, P Dumont, B Kennedy, G Null, W Owen Jr, J Riedel, S Synnott, and R Werner. Orbit determination performance evaluation of the deep space 1 autonomous navigation system. 1998.

[9] J.A. Crisp, M. Adler, S. Squyres, R. Arvidon, and D.M. Kass. Mars exploration rover mission. *Journal of Geophysical Research: Planets*, (108), 2003.

[10] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and Tanaka Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *International conference on parallel problem solving from nature*, pages 849–858. Springer, 2000.

[11] Stuart Taylor Douglas. *High accelerations produced through secondary impact and its effect on reliability of printed wiring assemblies*. PhD thesis, 2010.

[12] Harald T Friis. A note on a simple transmission formula. *Proceedings of the IRE*, 34(5):254–256, 1946.

[13] S.B. Guarro, G.A. Johnson-Roth, and W.F. Tonsey. Mission assurance guide. Technical Report TOR-2007(8546)-6018, The Aerospace Corporation, El Segundo, California, 2012.

[14] Mark Handley. Delay is not an option: Low latency routing in space. In *Proceedings of the 17th ACM Workshop on Hot Topics in Networks*, pages 85–91. ACM, 2018.

[15] Henry Helvajian. *Microengineering aerospace systems*. Aiaa, 1999.

[16] Chung-Ming Huang, Kun-chan Lan, and Chang-Zhou Tsai. A survey of opportunistic networks. In *22nd International Conference on Advanced Information Networking and Applications-Workshops (aina workshops 2008)*, pages 1672–1677. IEEE, 2008.

[17] Texas Instruments. Cc1310 simplelink ultra-low power sub-1ghz wireless mcu. *CC1310 datasheet, Sept*, 2015.

[18] Harald Keller and Horst Salzwedel. Link strategy for the mobile satellite system iridium. In *Proceedings of Vehicular Technology Conference-VTC*, volume 2, pages 1220–1224. IEEE, 1996.

[19] Glenn E Lightsey and John A Christian. Onboard image-processing algorithm for a spacecraft optical navigation sensor system. *Journal of Spacecraft and Rockets*, 49(2):337–352, 2012.

[20] Peter W Likins. Effects of energy dissipation on the free body motions of spacecraft. Technical Report 32-860, NASA JPL, 1966.

[21] A Liounis, S Daniel, and J Christian. Autonomous navigation system performance in the earth-moon system. In *AIAA Space 2013 Conference and Exposition*, 2013.

[22] Marco Lovera and Alessandro Astolfi. Spacecraft attitude control using magnetic actuators. *Automatica*, 40(8):1405–1414, 2004.

[23] Kris Maine, Carrie Devieux, and Pete Swan. Overview of iridium satellite network. In *Proceedings of WESCON'95*, page 483. IEEE, 1995.

[24] Zachary Manchester, Mason Peck, and Andrew Filo. Kicksat: A crowd-funded mission to demonstrate the worlds smallest spacecraft. 2013.

[25] F Landis Markley. Attitude determination using two vector measurements. Technical report, NASA, 1998.

[26] F Landis Markley and John L Crassidis. *Fundamentals of spacecraft attitude determination and control*, volume 33. Springer, 2014.

[27] William Marshall and Chris Boshuizen. *Planet labs remote sensing satellite system*. Smallsat Conference, 2013.

[28] Thomas A McMahon and John Tyler Bonner. *On size and life*. Scientific American Library, 1983.

[29] Jingshi Meng. *Multi-Scale Dynamic Study of Secondary Impact During Drop Testing of Surface Mount Packages*. PhD thesis, 2016.

[30] ST Microelectronics. Lsm9ds1 inemo inertial module. *Product Datasheet*, 2013.

[31] Ramin Moghaddass and Ming J Zuo. An integrated framework for online diagnostic and prognostic health monitoring using a multistate deterioration process. *Reliability Engineering & System Safety*, 124:92–104, 2014.

[32] Timothy J Pedley. Scale effects in animal locomotion. In *International Symposium on Scale Effects in Animal Locomotion (1975: Cambridge University)*. Academic Press, 1977.

[33] Eric R Pianka. On r-and k-selection. *The American Naturalist*, 104(940):592–597, 1970.

[34] Stephen R Pratt, Richard A Raines, Carl E Fossa, and Michael A Temple. An operational and performance overview of the iridium low earth orbit satellite system. *IEEE Communications Surveys*, 2(2):2–10, 1999.

[35] NASA Prognostics Data Repository. Overview. Retrieved November 1, 2014.

[36] Je rey Horn, Nicholas Nafpliotis, and David E Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the first*

IEEE conference on evolutionary computation, IEEE world congress on computational intelligence, volume 1, pages 82–87. Citeseer, 1994.

[37] G Roesler. Robotic servicing of geosynchronous satellites (rsgs). *Defense Advanced Research Projects Agency, http://www. darpa. mil/program/robotic-servicing-of-geosynchronous-satellites (14 July 2017). Google Scholar*.

[38] Robert R Schaller. Moore's law: past, present and future. *IEEE spectrum*, 34(6):52–59, 1997.

[39] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *science*, 324(5923):81–85, 2009.

[40] Knut Schmidt-Nielsen. *Scaling: why is animal size so important?* Cambridge University Press, 1984.

[41] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001.

[42] R. Shimmin, E. Agasid, R. Burton, R. Carlino, G. Defouw, A.D. Perez, A.G. Karacaliglu, B. Klamm, A. Rademacher, J. Schalkwyck, J. tilles, and S. Weston. Small spacecraft technology state of the art. Technical Report TP-2015-216648, NASA, Moffett Field, California, 2017.

[43] Isao Shimoyama. Scaling in microrobots. In *IROS (2)*, pages 208–211, 1995.

[44] Malcolm David Shuster and S D_ Oh. Three-axis attitude determination from vector observations. *Journal of Guidance, Control, and Dynamics*, 4(1):70–77, 1981.

[45] Enrico Silani and Marco Lovera. Magnetic spacecraft attitude control: a survey and some new results. *Control Engineering Practice*, 13(3):357–371, 2005.

[46] Marcello Spagnulo, Rick Fleeter, Mauro Balduccini, and Federico Nasini. Examples of management applied to different space programs. In *Space Program Management*, pages 277–330. Springer, 2013.

[47] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM*

SIGCOMM workshop on Delay-tolerant networking, pages 252–259. ACM, 2005.

[48] Grace Wahba. A least squares estimate of satellite attitude. *SIAM review*, 7(3):409–409, 1965.

[49] James R Wertz. *Spacecraft attitude determination and control*, volume 73. Springer Science & Business Media, 2012.

[50] J.R. Wertz and W.J. Larson. *Space Mission Analysis and Design*. Microcosm, 3 edition, 1999.

[51] Adeline D Yeh, Miguel I Gómez, and Gerald B White. Cost of establishment and production of vinifera grapes in the finger lakes region of new york-2013. 2014.