Angles-Only Navigation Technique for Maneuver-Free Spacecraft Proximity Operations

by

Max William Yates, Maj, USAF
B.S., Aeronautical Engineering, Rensselaer Polytechnic Institute, 2006
M.S., Astronautical Engineering, Air Force Institute of Technology, 2011

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Aeronautics and Astronautics
at the
Massachusetts Institute of Technology

September 2017

Author
....
Department of Aeronautics and Astronautics
July 18, 2017

Certified by
....
David W. Miller, Sc.D.
Professor, Aeronautics and Astronautics
Thesis Committee Chairman

Certified by
....
Sheila E. Widnall, Sc.D.
Institute Professor & Professor,
Aeronautics and Astronautics

Certified by
....
Sumanth Kaushik, Ph.D.
Group Leader, MIT Lincoln Laboratory

Certified by
....
Kyle T. Alfriend, Ph.D.
University Distinguished Professor &
TEES Distinguished Research Professor,
Texas A&M University

Certified by
....
Alvar Saenz-Otero, Ph.D.
Principal Research Scientist,
Aeronautics and Astronautics

Certified by
....
Hamsa Balakrishnan, Ph.D.
Assoc. Professor, Aeronautics and Astronautics
Chairman, Graduate Program Committee

Disclaimer

This material is based upon work supported under Air Force Contract No. FA8721-05-C-0002 and/or FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the U.S. Air Force.

Angles-Only Navigation Technique for Maneuver-Free Spacecraft Proximity Operations

by

Max William Yates
Maj, United States Air Force

Submitted to the Department of Aeronautics and Astronautics
on July 18, 2017, in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Aeronautics and Astronautics

Abstract

The technique of angles-only navigation consists of a single surveyor making line-of-sight observations of a target to deduce a relative navigation state from a sequence of angle measurements. Historically, angles-only navigation has been impeded by a range ambiguity problem in its many applications, especially those involving linear dynamical models. A classical solution to the problem is for the surveyor to perform precise maneuvers to change the nominal angle profile between the surveyor and the target. In the space environment, the orbital dynamics are inherently nonlinear and natural orbit perturbations have the effect of continuous micro-maneuvers. These advantageous conditions present an opportunity to overcome the ambiguity problem and enable spacecraft to navigate passively with a lightweight, low-power camera without the associated fuel cost of maneuver-assisted angles-only navigation. This technology has military and civilian utility for a wide range of missions involving rendezvous and proximity operations, most notably with non-cooperative resident space objects (RSOs).

A novel procedure is developed that constrains the admissible region of the target's natural motion to a set of unit-less parameters. These parameters and an arbitrary scale factor combine to describe a single orbit hypothesis that translates into a set of classical orbital elements (COEs). A cluster of uniformly sampled hypotheses are propagated and rendered into angle vs. angle-rate curves. Although these curves exhibit very similar trends for all admissible hypotheses, the angles are slightly different at common angle-rate waypoints during certain parts of the orbit. The set of angle and range hypotheses at these waypoints form a linear map to transform the observed angle to a range approximation. Photometry can complement this procedure with a secondary mapping from the timing of virtual eclipse events if a sufficient time differential is manifested across the admissible hypotheses.

A nonlinear least squares (NLS) filter is designed to refine the accuracy of the initial orbit solution using a novel application of Kolmogorov-Arnold-Moser (KAM) theorem to model the Earth's geopotential to any degree and order in the filter dynamics. The KAM torus conveniently captures the full nonlinear effects that make angles-only navigation possible in space and is computationally superior to numerically integrated reference trajectories for exact temporal synchronization with angle observations.

Numerical results are presented that demonstrate the first angles-only navigation technique for natural motion circumnavigation trajectories without prior knowledge of the Target's state. An analytical proof is developed to compliment and verify the results.

Thesis Supervisor: David W. Miller
Title: Professor, Aeronautics and Astronautics

Thesis Supervisor: Sheila E. Widnall
Title: Institute Professor & Professor, Aeronautics and Astronautics

Thesis Supervisor: Sumanth Kaushik
Title: Group Leader, MIT Lincoln Laboratory

Thesis Supervisor: Kyle T. Alfriend
Title: University Distinguished Professor & TEES Distinguished Research Professor, Texas A&M University

Thesis Supervisor: Alvar Saenz-Otero
Title: Principal Research Scientist, Aeronautics and Astronautics

Acknowledgements

None of this is possible without a core cast of hundreds for whom I am immeasurably grateful. Isaac Newton said, “If I have seen further than others, it is by standing upon the shoulders of giants.” Surely, these include the shoulders of my family, friends, educators, mentors, colleagues, and even a few rivals that broadened the horizon.

To my committee: Thank you for your many contributions to this work, for your wonderful insights and feedback, and for stretching my potential to succeed in this endeavor and all that come after. Dr. Saenz-Otero helped me find my footing at MIT and gave me unparalleled experiences conducting vision-based navigation experiments with the SPHERES satellites on the International Space Station. Prof. Alfriend helped me discover the intricacies and the elegance of relative orbital dynamics that permeate this entire body of work. Dr. Kaushik expanded my analytical thinking skills by spending countless Friday nights with me at a white board making sure the fundamental physics agreed with our computational findings. Prof. Widnall stimulated my critical thinking by challenging assumptions that ultimately led to a more robust methodology. She also helped sharpen my communications skills by giving me an opportunity to guest lecture in her astrodynamics class. Thanks also to Dr. Gregory Ushomirsky, an early member of the committee, who supervised my first two years at Lincoln Laboratory and empowered me in this field of study. Finally, Prof. Miller—my advisor—is the keystone for all of this. Not only did he give me an extraordinary opportunity, he invested countless hours working with me and stimulating new ideas even while immersed with his responsibilities as NASA’s Chief Technologist. His sharp mind, profound space systems expertise and out-of-the-box thinking made him the perfect advisor for tackling this challenging problem. I am forever indebted.

To my readers & mentors: Thank you for your incredible generosity, guidance and insight. I am particularly humbled and grateful for Maj Gen (ret) William N. McCasland’s mentorship over the past three years as I navigated his alma mater. The trust he placed in me at AFRL to help enhance our nation’s space situational awareness capabilities played a significant role in preparing me for this research and developing me as a future materiel leader for our Air Force. I owe Prof. William Wiesel credit for almost everything I know about astrodynamics. His innovative contributions in dynamical systems theory for Earth-orbiting satellites has yet to be fully recognized, but it inspired and enabled much of this work. Dr. Paul Cefola (a fellow RPI alumnus) has also been a terrific sounding board and mentor. His selfless service to a generation of MIT’s military graduate students since 1979 includes one astronaut so far… I’m also appreciative of Dr. Aaron

Schutte of The Aerospace Corporation who reviewed this work as an independent reader. I'm hopeful that this leads to further shared pursuits supporting our nation's aerospace industry and defense needs. Finally, Mr. Lloyd Magnuson, a former middle-school teacher, introduced me to the path of an explorer and the wonders of science. Our homemade 8" Dobsonian telescope was a proud platform for exploring new worlds for the first time. Homemade rockets blasted off from my dad's team roping arena as the horses reared in the background. The diamonds nestled within Montana's magnificient Big Sky were beyond our reach. The rockets would have to be bigger! The journey toward this end would pass through Edwards, Houston, Cambridge and places yet to be explored.

To my friends and family: Thank you for building me up, sustaining me and sharing this precious journey on our Pale Blue Dot. My mom and dad deserve so much credit. They worked tirelessly to make sure their three children had every opportunity that this world has to offer. The work ethic and values they instilled in us made all the difference. My sister and brother are the best that one could ask for. I am so proud of them and their beautiful young families. Finally, my partner, Jason, has shown tremendous resilience as a military spouse (and dare I say MIT spouse!). In moving homes and jobs, he has sacrificed his stability for my own. He is the real trooper. Thank you for your love and support! I could not have done it without you.

To my sponsors: Thank you to Lincoln Laboratory who generously funded my research through the Military Fellowship Program. Special thanks goes to Group 91 who adopted me as one of their own and to Col (ret) John Kuconis for overseeing the program. Thank you also to the Air Force for giving me the time and resources to pursue greater knowledge. Col (ret) Barbara Hunter has looked out for me since I wore gold bars and I am forever grateful for her faith in me as a technology leader in our Air Force.

M. Yates

Contents

Appendices

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Under a waning crescent moon[1] on the morning of May 23, 2014, a converted Soviet-era UR-100N intercontinental ballistic missile (ICBM) roared to life at Plesetsk Cosmodrome—a northern launch site in Russia that once threatened the West with nuclear-tipped missiles. The repurposed Rokot booster was purportedly carrying three military satellites for the Rodnik[2] (Strela 3M) constellation internationally designated Kosmos-2496, -2497 and -2498. Sensors in the U.S. Space Surveillance Network (SSN) obtained observations of the launch and satellites. As is customary for new launches, the U.S. Strategic Command's (USSTRATCOM) Joint Space Operations Center (JSpOC) assigned them entries in the U.S. space catalog for routine tracking: 39761, 39762 and 39763, respectively. The Briz-KM upper stage was also designated 39764, and a 30 cm object (internationally designated 2014-028E) that appeared to be debris was given 39765 [1][2][3][4].

The problem with 39765 is that it did things that drifting debris cannot do. 39765 performed impressive rendezvous and proximity operations (RPOs) with Briz-KM and perhaps even an interception or docking. In October 2014, JSpOC changed the designator

[1] Search for Plesetsk, Russia at https://www.wunderground.com/history/

[2] For more information on Rodnik, see http://russianspaceweb.com/rodnik.html

to Kosmos-2499 and categorized it as an experimental spacecraft without confirmation from the Russians. By the end of the year, Colonel General Oleg Ostapenko, the director of Roskosmos, acknowledged the satellite as a peaceful joint effort between Roskosmos and the Russian Academy of Sciences [4].

There are several justifications for the capabilities demonstrated by 39765 and other spacecraft like it: to deorbit or deflect dangerous debris; to inspect and/or repair defunct satellites; or to make scientific observations of natural objects—all noble endeavors. Several U.S. organizations have proposed autonomous inspector satellites for the accomplishment of these tasks, including NASA's Satellite Servicing Capabilities Office (SSCO) and the Defense Advanced Research Projects Agency (DARPA), but there are also strategic military reasons to mature these capabilities. From a defensive posture, there is a desire to covertly observe characteristics, capabilities, and intentions of other state actors. From an offensive posture, denying an adversary command and control, intelligence, surveillance and reconnaissance (C2ISR) capabilities and position, navigation and timing (PNT) networks with anti-satellite (ASAT) technology is the modern form of a blockade—an anti-access/area-denial (A2/AD) strategy that is global in nature. An attack in space would be a blinding event that could allow the perpetrator to seize the initiative and obtain superiority across all domains: air, sea, space, and cyber.

Top military officials within the People's Liberation Army (PLA) believe that Western technology combined with Eastern wisdom is the key to overcoming a confrontation with the U.S.—a concept known as *shashoujian*—and that a robust A2/AD strategy could achieve this [5][6]. According to PLA General Huang Bin, "We can fight a war with them [the United States], they will not be able to continue the war after a while. Moreover, we also have our shashoujian" [7]. In March 2013, Frank Kendall, the former Undersecretary of Defense for Acquisition, Technology and Logistics [USD(AT&L)], indicated that such an A2/AD strategy is possible with the rapid advancement of technology worldwide. From sensors and jamming equipment to precision guided munitions, unmanned aerial vehicles (UAV) and satellite technology, the U.S. no longer has the luxury of a monopoly on technological superiority. To overcome future A2/AD threats, we have to be strategically smarter.

The National Security Space Strategy (NSSS) [8] specifically calls for improved space situational awareness (SSA) in response to emerging threats. The United States Air Force (USAF) Doctrine on Space Operations divides SSA into four functional capabilities: characterization; detect/track/identify; threat warning and assessment; and data integration and exploitation [9]. A modest assessment of the detect/track/identify capability is to tally the publically available U.S. space catalog, which reveals that the SSN tracks approximately 23,000 objects on a daily basis. While this number may seem large, it is not large enough. An estimated 500,000 objects larger than 1 cm are not presently tracked [10]. While the vast majority of these objects are natural and artificial debris, there are operational satellites hidden within this population. NASA describes the current threshold for the size of SSN cataloged objects as ~10 cm in low earth orbit (LEO) and >1 m in geosynchronous earth orbit (GEO) [11]. Statistical models—developed in part by data collected from MIT Lincoln Laboratory's (MIT/LL) Long-Range Imaging Radar (Haystack radar), MIT/LL Haystack Auxiliary Radar (HAX), and the Jet Propulsion Laboratory's (JPL) Goldstone Radar—indicate that there are as many as 2,200 uncatalogued objects between the size of 1 cm and 1 m at GEO [11].

Some of these unknown space objects may already be wreaking havoc. U.S. Department of Defense (DoD) officials have expressed concern over anomalies at GEO that cannot be explained or resolved [10][12][13][14]. Are these nefarious acts or natural phenomena such as space weather events? An MIT study [15] on cross-correlation between Inmarsat anomalies and space weather events shows how natural phenomena can be probabilistically eliminated as a cause, but anomaly resolution is hardly the solution. This would be akin to preventing a murder by looking for bullet holes at the scene of a crime.

Space platforms with missions dedicated to SSA such as the Space-Based Space Surveillance (SBSS) and Geosynchronous Space Situational Awareness Program (GSSAP) are highly prized for the data and deterrence they provide, but there are presently just five of these assets that cannot be in all places at once. Therefore, the body of work presented in this dissertation enables a new concept that we call *deputization*. The idea is to supplement space-based SSN coverage with any willing and able satellites that can use their navigation sensors to perform SSA functions as a secondary mission (background tasks). Existing satellites can be deputized for local coverage without any new hardware since almost all operational spacecraft have some form of a camera such as star trackers for attitude determination. Star tracker algorithms typically ignore objects in the field of view that do not match the star field database, but with a software update, the angle observations of these objects can be recorded and processed to potentially estimate their orbits using angles-only techniques.

Camera-type sensors have an advantage over other relative navigation sensors in terms of availability. Far fewer satellites are equipped with heavier, bulkier, more complicated and ultimately costlier radar and LiDAR systems. Communications or global navigation receivers could also be repurposed for relative navigation by collecting third-party electromagnetic (EM) signals of opportunity reflected off nearby objects from omnipresent transmitters such as the Global Positioning System (GPS) or high-powered communications satellites. This is a passive ranging technique known as bistatic radar. The signals are typically too weak for far field operations as discussed in Section 2.1.1, and for that reason, camera-type sensors are superior for relative navigation.

Navigating with a camera is not without difficulties. At close distances, generally within a few hundred meters (the nearfield), object size and distance can be inferred from pixel disparities with stereo cameras or motion parallax, but at larger distances (the far field) the pixel disparities approach zero leading to scale ambiguity. Is the object small and near or is the object large and distant? The answers to these questions lie at the heart of this research and hold the key to enabling a deputization strategy for local space situational awareness (LSSA).

The classical problem of angles-only navigation attempts to disambiguate scale by using pure angle measurements to infer distance to the target. Linear systems are impossible to solve with angles-only navigation since an infinite number of trajectories can generate an identical angle profile. For these systems, range is unobservable. Even nonlinear systems with weak observability pose a challenge for state estimation with angles-only techniques. A brute force solution to angle-only navigation in its many applications has been to maneuver the observer, thereby altering the nominal angle profile. Consumables are a limited resource during space operations, so thrusting is not an optimal solution for spacecraft.

Fortunately, the nonlinear nature of space dynamics provides an opportunity to observe range without maneuvering, but after nearly six decades of research, the application of angles-only navigation for space operations remains unsolved without assistance from *a* *priori* data.[3] The difficulty in solving the problem is partially a human-made dilemma from estimation techniques that use linearized dynamical and measurement models that obscure weakly observable conditions from nonlinear and naturally perturbed dynamics. No common star tracker, let alone the most powerful eletro-optical payloads, can solve the problem with existing methods. If the method is the problem, then new methods are the solution. And with the right innovation, angles-only navigation will be a game changer for military and civilian spaceflight applications.

1.2 Problem Statement

An autonomous own-satellite, hereafter referred to as the *Surveyor*, is operating in a proximal orbit with an unknown, non-cooperative object, hereafter referred to as the *Target*. The Surveyor must passively estimate the relative position and velocity of the Target using angles and light intensity measurements from a monocular electro-optical camera. No *a priori* information about the Target is available or assumed. The Surveyor's inertial position, velocity and orientation are estimated onboard in real-time using GPS, a star tracker and possible ground updates. An ideal solution will render an estimate of the Target distance with an error of <1% for generating follow-on plans such as optimal collision avoidance maneuvers or rendezvous for close-up inspection and/or grappling/docking. The scenarios of interest are limited to natural motion circumnavigation (NMC) and distances in which the Target and Surveyor are separated by greater than 200 m, but less than 100 km—boundaries that will be justified in Chapter 2.

[3] Some researchers have claimed to solve the problem while still relying on ground-based observations of the target to initialize their estimator (e.g. [152]).

1.3 Research Objectives

This work will develop an angles-only navigation solution for passively estimating the relative kinematic state of a non-cooperative space object engaged in persistent, maneuver-free proximity operations with a Surveyor spacecraft. The specific research objectives are to:

1. Establish a framework for generating admissible orbit hypotheses from angle observations in realistic conditions (nonlinear orbital dynamics in the full geopotential).

2. Determine if, how and to what extent angle and angle-rate curves of admissible hypotheses can be used to disambiguate range from actual observations for an initial orbit determination (IOD).

3. Determine if, how and to what extent time differentials from an Earth eclipse can be used to approximate range as a supplemental technique for scaling an angles-only navigation solution.

4. Develop a batch filter process that adapts a nonlinear dynamics model of the IOD reference trajectory in the full geopotential.

1.4 List of Contributions

A literature review conducted in Chapter 2 reveals a valiant effort to solve angles-only navigation for space flight, but the cumulative body of work spanning more than six decades includes several less-than-ideal solutions: maneuver-assisted or spin-assisted techniques, pseudo-ranging techniques that require prior knowledge of the Target's shape and size, and other methods of augmentation from coarse range knowledge such as two-line element sets (TLEs) or radar campaigns (supplants initial orbit determination). This author is not aware of any pure angle-*only* navigation solutions for space rendezvous and proximity operations in the open literature. Major contributions developed and claimed in this thesis toward the field of angles-only navigation in space include:

1. Establishing the first-of-its-kind, *non-augmented, non-maneuvering* angles-only navigation technique for initial orbit determination of natural motion circumnavigation trajectories.

    a. Parameterizing an admissible region for generating orbit hypotheses in a multiple-hypothesis framework.

    b. Discovering and verifying an explicit range relationship with the angle and angle-rate curves from a set of admissible hypotheses.

    c. Formulating a concept for using an Earth eclipse to estimate or verify range in a multiple hypothesis framework.

2. Designing, developing and verifying an angles-only batch filter that uses Kolmogorov-Arnold-Moser (KAM) theorem to model nonlinear dynamics in the Earth's full geopotential.

3. Conceiving a qualitative metric for assessing range resolvability and observability in a multiple hypothesis framework.

1.5 Outline of Thesis

Chapter 1 motivates the need for passive relative navigation technology to conduct RPOs with non-cooperative space objects. A special emphasis is given to SSA capability gaps that threaten U.S. national security. The concept of deputization for local SSA is articulated as a possible stopgap measure enabled by camera-type sensors and angles-only navigation. The problem statement emerges from these needs. The research objectives and contributions of this research are also introduced.

Chapter 2 explores the state of the art for relative navigation technology used in space and reviews the literature associated with some of these technologies, particularly angles-only navigation.

Chapter 3 contains the fundamentals of angles-only navigation such as descriptions of reference frames, measurement models, and transformations that map observations in the camera frame to line-of-sight vectors in the local vertical local horizontal frame. This material is mostly standard across all space-based angles-only navigation research and is isolated from the other Chapters since it does not contain novel contributions to the field.

Chapter 4 introduces a new perspective on how one may observe range from nonlinear orbital dynamics. The derivation of Clohessy-Wiltshire equations is performed, but not with an intent to use the equations; we reveal quantitatively how simplifications in their derivation lead to a complete lack of range observability. By doing so, we can appreciate what we have to gain from the pristine nonlinear dynamics. Chapter 4 introduces a novel method of mapping range from the angle and angle-rate profiles with nonlinear dynamics and presents an analytical proof for demonstrating range observability. A metric for range resolvability and observability in different orbital regimes is a byproduct of the proof.

Chapter 5 describes a methodology for performing initial orbit determination from angles-only observations during natural motion circumnavigation (no maneuvering or prior knowledge of the Target's orbit are used). A set of dimensionless parameters are defined that constrain the hypotheses to an admissible region from which range may be approximated with a novel range mapping technique. A batch filter is developed that employs a KAM torus of the IOD reference orbit to model the global and local motion dynamics for an enhanced Target orbit estimate. Verification tests are conducted on the hypothesis generation algorithm within the admissible region and on the dynamics models in the filter.

Chapter 6 presents the results of our angles-only navigation pipeline from IOD through differential correction. It demonstrates the first *pure* angles-only navigation solution for NMC trajectories. Quantitative results show the accuracy of the solution.

Chapter 7 culminates the thesis with conclusions and a list of recommendations for future work.

Chapter 2

Background and Literature Review

We begin our investigation into the problem of angles-only navigation with a historical perspective as a backdrop. Throughout most of history, humans have navigated the Earth by predominantly passive techniques. The Phoenicians (1500-300 BC) are among the first civilizations known to have used the positions of celestial bodies for navigation. Until the creation of the chronometer in 1761, pathfinding with stars was limited to latitudinal measurements since longitudinal measurements require an accurate timepiece. Navigators had to supplement the stars with something else for east-west navigation. Early Polynesians—perhaps most famously the Māori people who migrated by canoe from eastern Polynesia to New Zealand around 1280 AD—are thought to have navigated the South Pacific by observing other natural phenomena such as cloud formations, ocean swells and light glistening off waves on the horizon[4] [16]. At higher latitudes, Nordic explorers could not rely on the stars during summer months, so they took cues from the direction that seabirds traveled: birds with a beak-full of food were generally headed toward land; hence the term "as the crow flies" [16]. More complicated lunar-distance methods appeared in the 18th century to overcome the ambiguity in east-west navigation. None of these techniques was very accurate, but the feats of the Phoenicians, Polynesians,

[4] Light is known to reflect differently from waves crashing against a shoreline in the distance.

and others throughout history is a worthy place to start. Their clever tricks to navigate our world with so few resources provides a valuable lesson about our modern challenges; there is always a way!

The steady progression of human achievement in position, navigation and timing (PNT) from compasses (~300 BC) to sextants (1730) to GPS (1978) have led to unprecedented absolute navigational capabilities on Earth. Similarly, the Deep Space Network (1963) and other radiometric ground stations around the globe have enabled space probes to navigate every planet and several other bodies in our solar system throughout the past half-century including America's Apollo Program that landed six manned vessels on the surface of the Moon between 1969 and 1972. Despite these remarkable advancements, alternative PNT technologies are still intensely pursued for absolute and relative navigation.

Given the nature of the problem posed in Section 1.2, this chapter unfolds with a cursory look at a wide range of relative navigation technologies and algorithms to contextualize the state-of-the-art and the current capability gap for passive relative navigation. Given the allure of bistatic radar for passive relative navigation with non-cooperative targets, a special section will briefly explore its limitations as an alternative to camera-type sensors. Finally, an in-depth review of literature is performed on angles-only navigation across all physical domains—land, sea, air and space—with particular emphasis given to the latter.

2.1 Relative Navigation Technology for Space Flight

In the space domain, a perennial interest in missions to natural small body satellites like asteroids and comets, a growing prevalence of resident space objects (RSOs) such as orbital debris, and increasing threats from adversarial satellites demand an array of PNT capabilities to perform rendezvous and proximity operations (RPOs) with non-cooperative targets. Rendezvousing with an RSO typically requires a range estimate having less than 1% error [17]. Fehse [17] and Uhlig et al. [18] surveyed existing relative navigation technologies to assess which sensors could meet this accuracy requirement at different ranges. Their data is replicated in Figure 1.

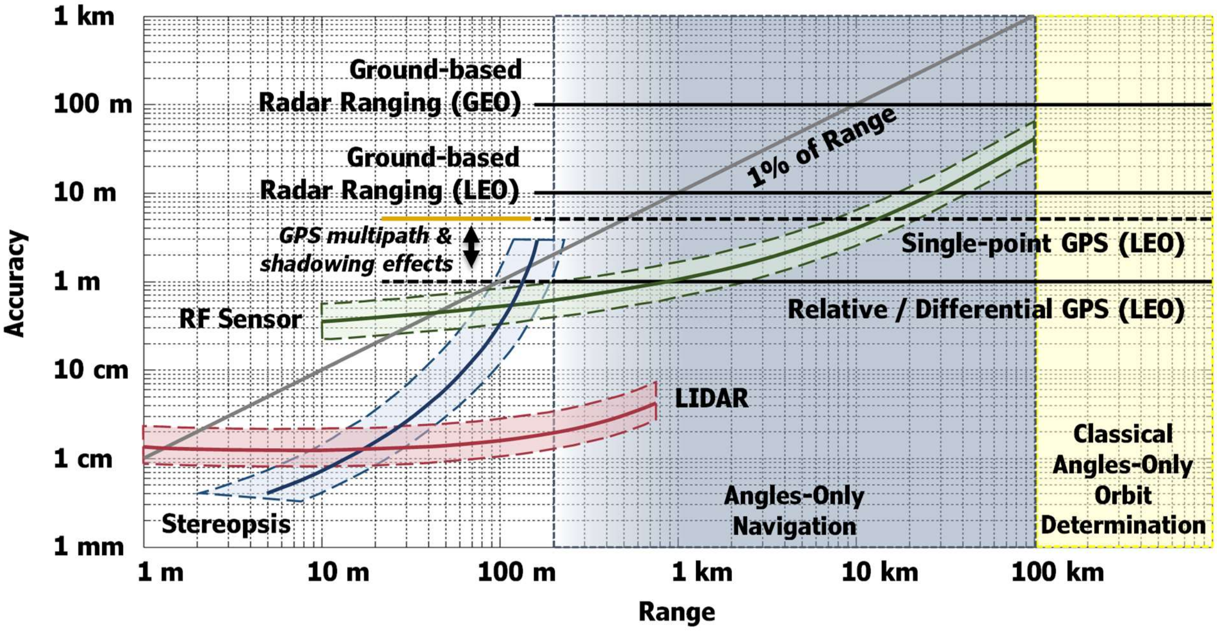


Figure 1: Typical Sensor Capabilities for Absolute & Relative Navigation. Sensors such as RF (green line) and LIDAR (red line) perform very well for relative navigation, but they are not passive. Camera-based sensors that can use stereopsis (blue line) to estimate depth perform well at close range. Beyond ~200 m,[5] stereopsis becomes impractical due to loss of accuracy. Classical angles-only orbit determination techniques with a camera sensor

[5] Fehse does not specify the parameters used to determine the stereopsis curve, but the threshold is not constant since it depends on the satellite design. If we consider 12 megapixel cameras with a focal length of 25 mm and a pixel pitch of 3.1 μm, the baseline length would be approximately 2.5 m for a downrange error of 1% at 200 m. Clearly, small spacecraft will have a smaller threshold than that which is depicted.

generally produce useful results at distances greater than 100 km[6] (shaded yellow). If ground-based radar cannot observe the non-cooperative target, the gap between 200 m and 100 km (shaded blue-gray) must be filled with alternative navigation techniques such as angles-only navigation.

Radar and LIDAR are suitable for ranging to a non-cooperative Target for RPOs, but these are active techniques that may be prohibitive to the size, weight, and power (SWaP) budget of the Surveyor. Most satellites are not equipped with radar and LIDAR anyway, but those that are equipped, may have a need for backup relative navigation techniques or may have mission constraints on electromagnetic emissions. Passive RF tracking—also known as electronic support measures (ESM)—is not an option if the Target is silent or incapable of transmitting signals. Relative GPS is also eliminated from consideration when navigating with a non-cooperative Target. Assuming the Target is unobservable by ground-based radar ranging due to a small radar cross-section and weak returns from GEO, this technique is eliminated as well. If the Surveyor must navigate autonomously, it cannot count on a ground uplink for navigation in any case—not even for a TLE. Moreover, a TLE will not exist if the Target cannot be observed by ground-based radar and electro-optical systems. The only remaining options in this plot are GPS multipath and stereopsis, or more generally, relative optical navigation using camera sensors. These sensors are explored in the next two subsections.

[6] Note that the upper threshold is dependent on the orbit altitude and eccentricity. Generally, at lower altitudes, a short arc of the orbit exhibits more curvature (more nonlinear) than higher altitudes, so we treat this upper threshold as merely a loose rule of thumb.

For a historical review of orbital rendezvous missions spanning the period from 1960 to 2007 and insight into the relative navigation technologies that were used, see Woffinden* and Geller[7] [19].

2.1.1 Spaceborne GPS for Relative Navigation

In 1984, the Landsat-4 remote sensing satellite became the first to carry a spaceborne Global Positioning System (SGPS) receiver. At the time, there were only a handful of Block-1 GPS satellites, so position accuracy was limited to about 50 m [20]. Since that time, the accuracy of SGPS has improved by as much as four orders of magnitude with carrier phase differential GPS (see [21][22][23][24]) thereby opening up new opportunities for satellites to exploit GPS for spacecraft operations and Earth monitoring missions (receivers can double as distributed sensors). One particular boon for operations is precision orbit determination, which enables new tracts for formation flying spacecraft and angles-only navigation by providing the Surveyor its absolute position and velocity.

A major underlying assumption for the objectives of this thesis is that the Surveyor's inertial position and velocity are known. SGPS represents one approach for meeting this PNT requirement. Additionally, it may be possible to use GPS as a bistatic radar to estimate the relative position of a nearby non-cooperative Target as shown in Figure 2. In this section, we investigate the state of SGPS technology to potentially satisfy these prospects.

[7] David Geller has been a preeminent voice in the field of angles-only navigation for the past two decades. His work began at Draper Laboratory where he advised students at MIT before becoming part of the faculty at Utah State University. Geller's students will be indicated with an asterisk (*) throughout this thesis out of respect for their many contributions.

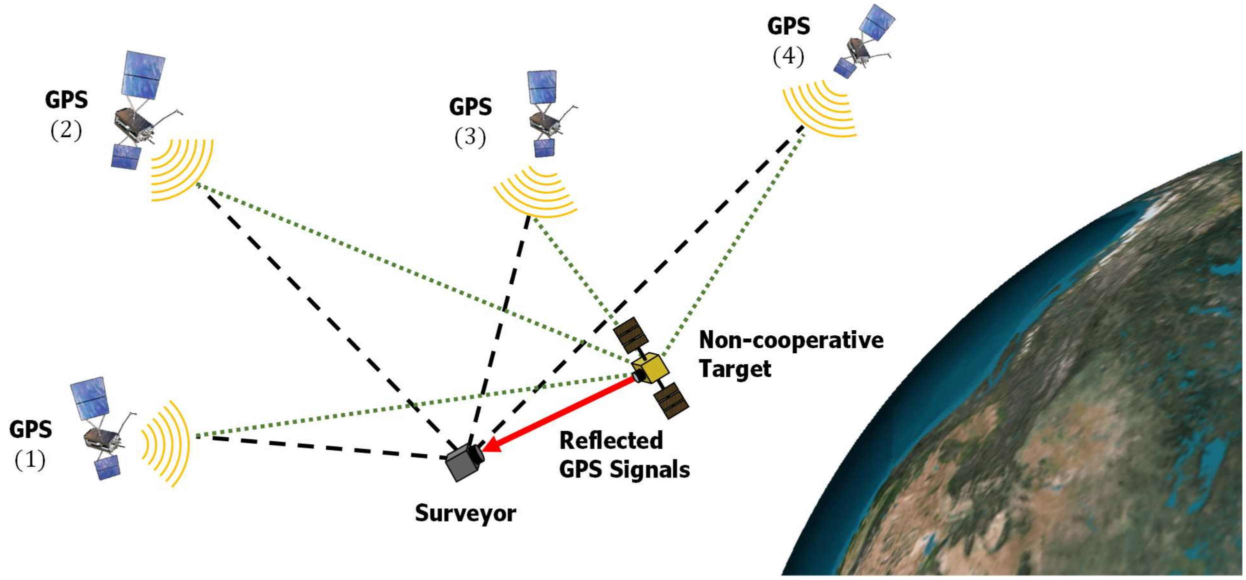


Figure 2: Multiple GPS Bistatic Radar. The concept of target localization with GPS multipath reflections is depicted. As with direct GPS localization, at least four GPS satellites must be in view.

GPS Challenges in Space

The concept of using GPS for the navigation of space platforms is not entirely unlike terrestrial and air-based platforms, but there are some limitations (signal availability), restrictions (export control laws) and special considerations (hardware and software). Since GPS satellite antennas are nadir-facing from an altitude of ~20,200 km, low earth orbits (LEO)[8] are fully encompassed by the volume of the main beam (23-26 deg to cover the Earth) [25]. SGPS receivers that exceed the GPS orbit altitude, such as highly elliptical orbits or geosynchronous orbits (see Figure 3), will experience limited performance in accuracy due to GPS signal visibility and strength. The spillover from the main beam and the weaker side lobes may still be used for orbit position and timing, but a navigation filter is needed for sparse observations [25]. The Block III satellites have been

[8] LEO is defined as less than 2,000 km altitude.

developed to provide improved signals past the Earth's limb to support spacecraft operations at GEO [26].

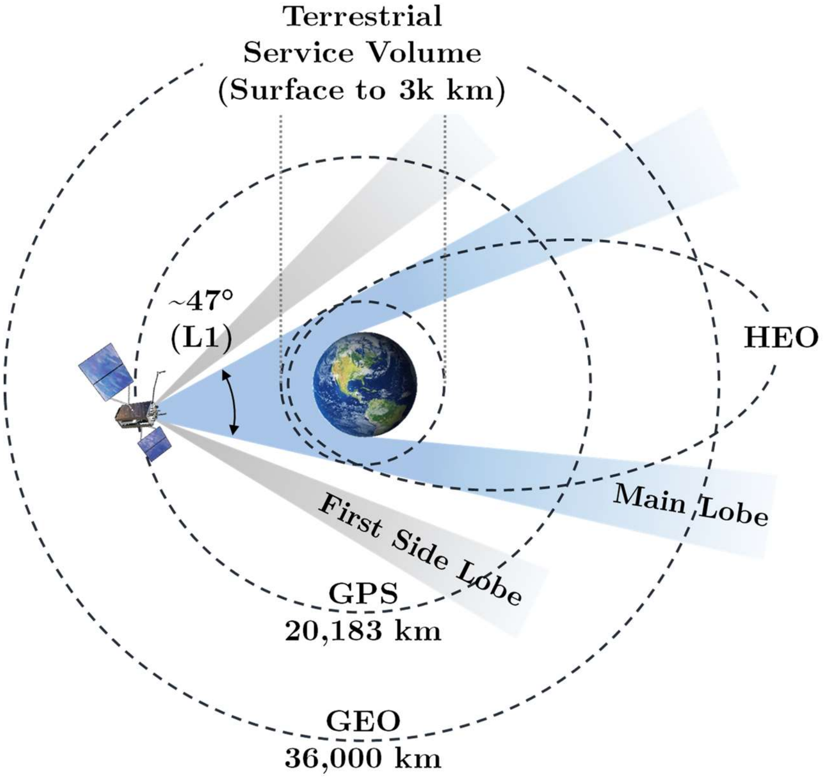


Figure 3: GPS Space Service Volume. GPS satellites orbit at an altitude of ~20,200 km. The main lobe of the L1 signal has a beamwidth of ~47 deg that extends beyond the Earth's limb to cover the entire terrestrial service volume up to an altitude of 3,000 km. Satellites that use GPS for navigation above LEO require an estimator to handle sparse signals.

One major distinction between SGPS technology and its terrestrial equivalent is commercial availability due to U.S. export control laws. SGPS receivers are regulated by the U.S. Commerce Department on the Commerce Control List (CCL) and the State Department-controlled U.S. Munitions List (USML) to prevent bad actors from using them for military or missile-specific applications [27]. While many U.S. allies are able to obtain approval to equip their satellites with SGPS receivers, the hardware options have been scarce until recently and quite costly.

The scarcity of commercial-off-the-shelf (COTS) SGPS receivers stems from the added engineering challenges associated with spaceflight operations to include vibration and shock loads from launch and satellite deployment, thermal-vacuum conditions (fluctuation of temperature from -30 to 60°C), ionizing radiation and single-event upsets. Several terrestrial COTS receivers have demonstrated a tolerance of 5-15 krad total ionization dose (TID) [21], which is typically sufficient for short duration LEO, missions. Only a handful of COTS receivers has gone through board-level single event latch-up (SEL) tests which tend to be a minimum requirement to avoid total failure from the overcurrent effects of radiation [21]. Given all of the extra engineering and testing to qualify space-grade receivers and the relatively small customer base, the price per unit is dramatically higher than terrestrial receiver are priced.

The extra cost and engineering challenges are no barriers for the U.S. defense and civil space programs. SGPS was recently used on the U.S. Air Force Research Laboratory's Automated Navigation and Guidance Experiment for Local Space (ANGELS) satellite for conducting angles-only navigation experiments with the upper stage of its Delta-IV launch vehicle. Launched in July 2014, ANGELS used "advanced algorithms from NASA to receive GPS side lobe signals and generate near-continuous navigation solutions" [28]. The results of the experiment are classified; nevertheless, the mission is proof that GPS can be used for absolute navigation of the Surveyor despite GPS signal challenges at GEO.

GPS Reflectometry

The omnipresence of GPS signals and the inescapable opportunity (or misery) of multipath propagation has led to its dual use as a bistatic radar for several innovative applications in remote sensing, relative navigation, and object detection. The field of GPS reflectometry (GPS-R) seeks to exploit GPS multipath as a passive radar system. Farmers and ranchers can now use GPS to measure soil moisture content in their fields for precision irrigation and water management [29]. Geophysical surface characteristics, snow pack and ocean roughness (correlates with wind speed) can be garnered from Earth monitoring missions at LEO [30][31]. Airborne experiments have demonstrated GPS bistatic radar for aircraft altimetry [29], ship detection [32], and the detection of land structures in urban and rural environments [33].

Long before GPS, the idea of "sanctuary" bistatic spaceborne radar (SBR) was envisioned to provide global coverage for ground-based or airborne receivers. In the late 1970s, a communication satellite was demonstrated as a bistatic illuminator but suffered from limited range capabilities against ground and airborne targets (2 nautical miles) [34].

The idea of using GPS as a bistatic illuminator for relative navigation or surveillance in space is a relatively new one in the literature despite the old idea. The application of SGPS for RPOs only emerged in the last two decades, but the original idea applied only to cooperative scenarios where both spacecraft were equipped with SGPS receivers and multipath was treated as an annoying source of error from complex geometric reflections. In 2003, Gaylor and Lightsey studied GPS blockage and multipath in the close vicinity of the ISS [35] and modelled the errors in the design of a GPS/INS extended Kalman filter (EKF) [36]. Shah et al. also studied multipath signals from GPS receiver data collected on the last Hubble servicing mission flown in May 2009 (STS-125) [37].

Cohen [38] performed the first known published work that sought to utilize GPS multipath signals for rendezvous in 2007. He designed an EKF for a simulated Space Shuttle servicing mission to the Hubble Space Telescope. A very simple cylinder radar cross-section is used for modelling GPS signal reflections off Hubble. Cohen's simulated scenario initializes the Shuttle and Hubble 574.5 m apart with exact knowledge of their initial state in the filter. Clohessy-Wiltshire dynamics are used for the rendezvous, which appear to be sufficient even when GPS reflections are sparse. A higher fidelity nonlinear dynamics model (including J_2 and air drag) was also attempted, but Cohen admits there are problems with his implementation. In any case, the results indicate that GPS bistatic radar is possible for relative navigation at *close range*. Based upon these results, GPS bistatic radar does not appear to fill the gap identified in Section 2.1.

Another study in 2009 by Pogemiller [39][40] explored the feasibility of GPS bistatic radar for small satellite applications using a commercial-off-the-shelf (COTS) GPS receiver. There are no novel contributions from Pogemiller's work other than to verify the underlying link budget analysis for ground-reflected and satellite-reflected GPS signals. The latter is marginally interesting because it can be used to show a practical range limit for GPS bistatic radar at different orbital regimes using COTS hardware as demonstrated next.

Limitations of GPS Bistatic Radar for Target Localization

Target localization with GPS bistatic radar is possible with at least four GPS satellite signals just like traditional GPS navigation. Within the terrestrial service volume ($<$ 3000 km altitude) which encompasses LEO, it is anticipated that four or more GPS satellites will be in view at all times. Beyond the terrestrial service volume, the number of visible signals becomes sparse. At altitudes higher than 20,183 km (GPS orbit), there may be GPS satellites in view, but their direct broadcast signals are not observable since the transmit antenna is nadir-pointing. Instead, multipath signals from these signals will be detected after reflecting off the surface of the Earth, but are expected to be too weak for bistatic radar. Other weak signals will be present from satellites on the opposite side of the Earth. The Earth occludes most of the main lobe, but the outer edge of the main lobe and even weaker side-lobes are present. The availability of four sufficiently powerful signals at GEO and HEO for traditional GPS navigation is non-continuous, but enabling.

The signal-to-noise ratio (SNR) for bistatic radar limits the operating range for relative navigation. Unlike the monostatic radar equation that scales with $1/R^4$ along the same uplink and downlink path, the bistatic radar equation must account for separate uplink and downlink paths as shown in Equation (2.1)

$$P_R = \frac{P_T G_T G_R L \lambda^2 \sigma}{(4\pi)^3 R_T^2 R_R^2} \tag{2.1}$$

where P_R is the received power, P_T is the transmitted power which is known for GPS ($P_T = 27$ W, 14.25 dBW), G_T is the gain of the transmitter antenna which is known for GPS ($G_T = 12.9$ dBi), G_R is the gain of the receiver antenna which is a design parameter for improving SNR, L accounts for all system losses, λ is the wavelength of the signal which is known for GPS ($\lambda = 0.1903$ m), σ is the radar cross section (RCS) of the Target, R_T is the distance between the transmitter (GPS) and Target, and R_R is the distance between the receiver (Surveyor) and Target.

The broadcast power of GPS is designed to provide a minimum of -160 dBW on the Earth's surface for coarse acquisition (C/A) codes in the L1 band [41]. This threshold roughly describes the minimum capability of standard COTS GPS receivers. High sensitivity GPS receivers can use correlation processing over an extended period to extract and calculate position from “a weak signal that is 1/1000 the strength of typical outdoor signal” [42]. Therefore, a reasonable approximation for the minimum P_R for high sensitivity GPS receivers that would be preferred for bistatic radar is -190 dBW.

A quick, back-of-the-envelope range feasibility study can be performed by rewriting the bistatic radar equation to solve for range

$$R_R = \sqrt{\frac{P_T G_T G_R L \lambda^2 \sigma}{(4\pi)^3 R_T^2 P_R}} \tag{2.2}$$

Consider a simple LEO scenario ($R_T = 19{,}800$ km) where the Target is a perfect sphere ($\sigma = \pi r^2$). Assume the very best case scenario of a lossless system ($L = 1$) and a high gain receiver ($G_R = 14$ dBi). The range limit as a function of the Target radius is plotted for the cases corresponding to a standard and high-sensitivity receiver: $P_R = -160$ dBW and $P_R = -190$ dBW, respectively.

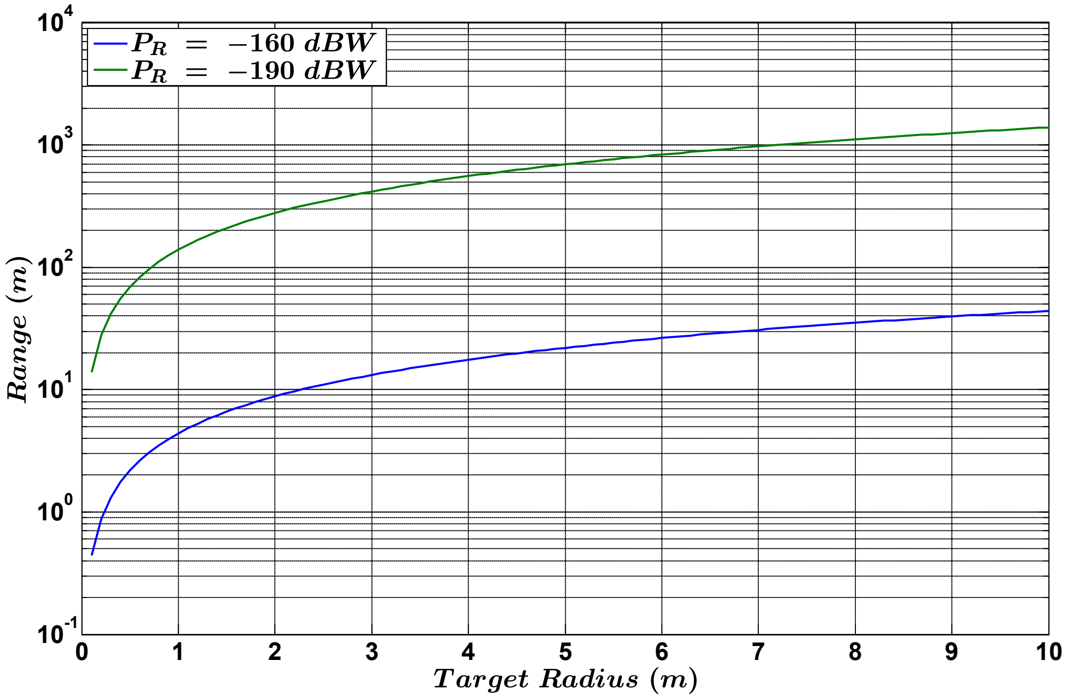


Figure 4: Range Limitation for GPS Bistatic Radar. The bistatic radar equation is used with a lossless system to bound the feasible envelope for range estimation. A standard GPS receiver with a received power of $P_R = -160$ dBW is represented by the blue line and a high-sensitivity GPS receiver with a received power of $P_R = -190$ dBW is represented by the green line.

These results clearly demonstrate that even the best-case scenario of a lossless system and a high-sensitivity receiver barely exceeds 1 km for a 10 m radius Target. Since the most likely Target-class of interest is smaller than 1 m, GPS bistatic radar is not even feasible for a small fraction of the 0.2-100 km range capability gap. Relative optical navigation is, in fact, the only possible solution given the current cognitive and technological capacities of the human race.

2.1.2 Relative Optical Navigation

Several methods of relative optical navigation are possible depending on knowledge and cooperativeness of the Target, number of cameras on the Surveyor, as well as the distance separating the Target and the Surveyor. Among these approaches is angles-only navigation, also known as line-of-sight (LOS) navigation, which uses a single camera to generate a unit vector to the Target centroid with the same image processing techniques used for star-trackers [43]. This approach works for unresolved and resolved imagery as long as the extent of the Target does not exceed the camera frame. Section 2.2 will cover the literature on angles-only navigation. The other approaches require resolved imagery and are most accurate when the Target extent is *much* greater than a detector element or pixel. These approaches include pseudo-ranging (use knowledge of Target dimensions), feature-based triangulation (use knowledge of sparse fiducials or features/landmarks on the Target), motion parallax or temporal stereo from a monocular camera system (use pixel disparity between dense random features observed at different times), and binocular stereopsis from two synchronized cameras (use pixel disparity between dense random features observed at the same time from different perspectives).

Pseudo-ranging

By knowing the Target's shape and dimensions, the pixels in the resolved imagery can be counted to estimate the Target's range since the pixel pitch and focal length are known. The range is not actually measured, hence the term pseudo-range. Woffinden* [44] used knowledge of a spherical spacecraft's diameter to estimate a pseudo-range at distances less than 500 m in simulations, but this approach is almost never an option since one cannot presume knowledge of the Target size. Terrain-based navigation uses a similar strategy in which crater maps of celestial bodies are used for pseudo-ranging at even greater distances [45][46][47]. In 2019, NASA's Exploration Mission 1 (EM-1) will use the diameter of the Moon for ranging in Orion's distant retrograde orbit (DRO) [48].

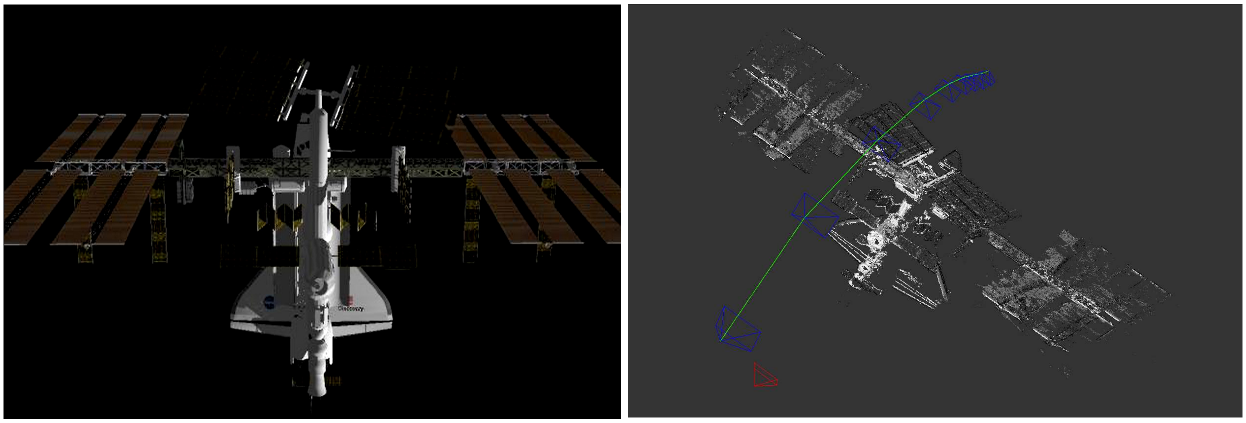
Triangulation

For a known or mapped Target, feature points or fiducial markers can be used to triangulate the Surveyor's location. Calhoun and Dabney [49] present algorithms developed at Marshall Space Flight Center (MSFC) for estimating the 6 DOF state using vector measurements to either a 3-spot or 5-spot reflective target that are illuminated by an array of laser diodes on the Surveyor. A similar capability called the Autonomous Navigation (AutoNav) System has been developed by the Jet Propulsion Laboratory (JPL) for navigating on interplanetary missions with known and unknown targets [50]. AutoNav relies entirely on optical images, so it can navigate using natural landmarks on the target body as opposed to artificial ones. To learn those landmarks, AutoNav uses stereophotoclinometry capabilities such as those of Gaskell [51] to produce landmark maps (L-maps). At an astronomical scale, AutoNav can triangulate from known objects ("beacons") in the solar system to estimate absolute position as well. Several interplanetary missions have successfully used AutoNav such as Deep Space 1 (flyby of Borrelly comet) [52], Stardust (flybys of Wild 2 and Tempel 1 comets and Annefrank asteroid), and Deep Impact (flyby of Hartley 2 comet).

The triangulation approach for navigating interplanetary probes has an interesting heritage in ballistic missile technology. MIT's Instrumentation Laboratory (IL)—the precursor to Draper Laboratory—designed the inertial navigation systems for most of the United States' strategic weapons systems (see [53]). A military requirement then and now for strategic systems is that the guidance and navigation system shall be self-contained. In other words, once launched, the missile shall be able to navigate autonomously without external guidance by way of ground or satellite communications. This mentality influenced an Air Force funded IL study in the late 1950s on a Mars probe that would navigate by "automatically measuring four angles, between the sun, stars, and planets, to determine the spacecraft's position" [54]. This mode of navigating is reflected in the writing of the late Richard Battin in his book *Astronautical Guidance* [55] and played a major role in the development of the Apollo Guidance System (see Chapter 5 in [54]).

Motion Parallax

A sequence of images (an image set) from a monocular camera system can estimate 3-dimensional (3D) structure and motion fields; however, without knowledge of the camera location from frame-to-frame (odometry) or a depth sensor, the scale of the scene is unknown [56]. Since the mid-1980s, so-called "structure from motion" (SFM) algorithms (e.g., [57][58]) and scene flow algorithms (e.g., [59][60][61][62][63]) have saturated the computer vision field in support of autonomous robot navigation. SFM tends to be an off-line, batch process for digitally reconstructing a 3D scene and/or the camera trajectory, but with the evolution of computer processing over the past several decades, scenes and trajectories can now be generated in real-time with simultaneous localization and mapping (SLAM) algorithms (e.g. EKF SLAM [64], MonoSLAM [65], FastSLAM [66], PTAM [67], RTAB-Map [68], LSD-SLAM [69]) and Bayesian inference or recursive estimation. This approach is useful for generating scalable maps that can be used to characterize a non-cooperative Target, but it does not provide a range, which is essential for navigating. This author has experience implementing Large-Scale Direct SLAM (LSD-SLAM) in parallel with a space-based optical navigation simulator to demonstrate the state of the art in the presence of solar illumination effects. An example of this work is shown in Figure 5.



(a) Simulated view from the Surveyor of the ISS docked with the Space Shuttle

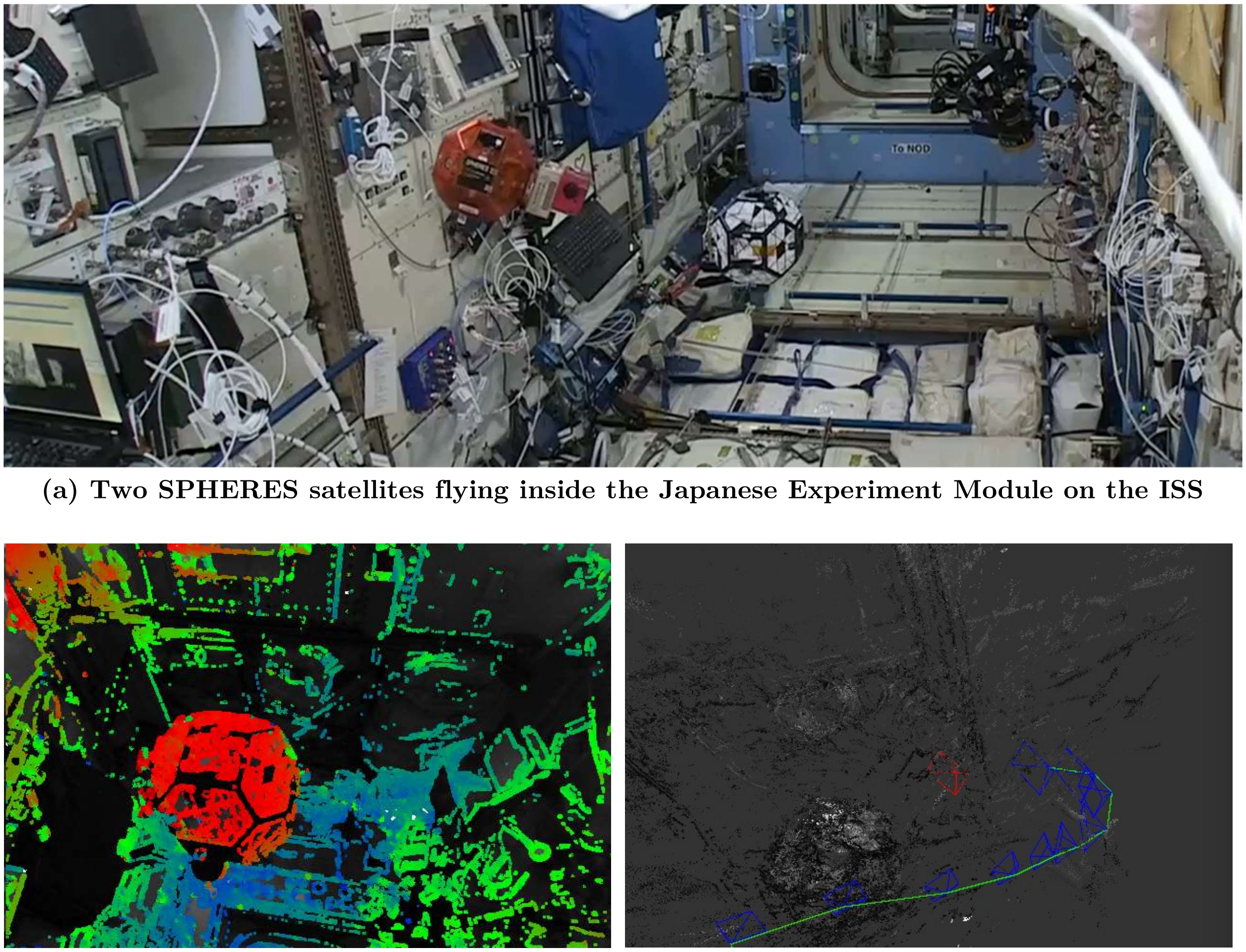
(b) Point cloud map of ISS & Space Shuttle as generated from LSD-SLAM

Figure 5: Demonstration of LSD-SLAM in Yates' Optical Navigation Simulator. A space-based optical navigation simulator generates a sequence of images from a realistic relative orbit such as in (a) that are processed in real-time with LSD-SLAM to create a scalable point cloud map and an estimate of the relative trajectory as in (b).

To overcome range ambiguity with monocular SLAM algorithms, visual data can be fused with other data from any number of sensors: IMU, GPS, LIDAR and/or sonar [70]. Multi-sensor SLAM algorithms have become prevalent and widely utilized on uninhabited aerial vehicles (UAVs) (e.g., [71][72]), self-driving automobiles (e.g., [73]) and autonomous underwater vehicles (AUVs) (e.g., [74]).

Binocular Stereopsis

The perception and estimation of depth/scale are possible with two synchronized cameras having a known baseline distance. The ambiguity in scale from SFM and SLAM is eliminated allowing a pure optical navigation system to estimate its actual trajectory (e.g., [75]). As the baseline increases, so too does the parallax or binocular disparity between features in the images allowing greater depths to be measured. JPL's exploration rovers on Mars are famous for using binocular stereopsis [76]. MIT's Space System's Laboratory also developed a stereovision appendage for the Synchronized Position Hold Engage and Reorient Experimental Satellites (SPHERES) for conducting optical navigation experiments onboard the International Space Station (ISS) [77][78]. The SPHERES are shown in Figure 6 during an optical navigation experiment designed by this author and conducted by astronauts Scott Kelly and Kimiya Yui on Expedition 45 that demonstrated trajectory reconstruction of scaled relative orbital dynamics for satellite inspection missions.



(a) Two SPHERES satellites flying inside the Japanese Experiment Module on the ISS

(b) Depth map of stationary target satellite. Red indicates closer objects. Blue indicates more distant objects.

(c) Point cloud map of target satellite and trajectory estimate from LSD-SLAM

Figure 6: SPHERES Satellite Experiment. In this experiment, a stationary SPHERE is imaged by an inspector SPHERE's binocular camera on a forced

trajectory that emulates natural motion circumnavigation. The sequence of images were post-processed in real-time with LSD-SLAM to create a point cloud map and an estimate of the relative trajectory as in (c).

The biggest limitation of binocular systems is that the range error grows quadratically with range and inversely with the camera baseline. Spacecraft hardware dimensions typically prohibit a large camera baseline, so there is a practical limit on range. Fehse [17] sets this range limit at $100-200$ m where range accuracy degrades beyond the 1% range error objective for rendezvous. Fehse does not specify the parameters used to determine the threshold, but if we consider 12 megapixel cameras with a focal length of 25 mm and a pixel pitch of 3.1 μm, the baseline length would be approximately 2.5 m for a downrange error of 1% at 200 m. Clearly, small spacecraft will have a smaller threshold, so the Fehse boundary is merely illustrative of a particular binocular system in the mini to large class of satellites that can accommodate meter-level baselines.

In summary, for distances less than approximately 200 m where parallax may be reasonably observed, stereo cameras are ideal for passive relative navigation. Beyond about 200 m, the range error of stereo cameras exceeds 1%. Angles-only navigation with a monocular camera is the only remaining approach for performing relative optical navigation at these greater distances.

2.2 Angles-only Navigation

The terminology of angles-only navigation (AON) has been used synonymously in literature with line-of-sight (LOS) navigation and bearing-only tracking (BOT) in which the name tends to be a matter of preference. The problem is fundamentally the same in all three cases: how to determine the location of an observed Target using direction data alone when no range information is explicitly available. This author distinctly delineates the terminology to treat BOT and LOS navigation as subclasses of AON based upon methodological and application heritage described below. This delineation helps to categorize the literature cleanly.

2.2.1 Bearing-only Tracking

BOT terminology tends to be associated with naval, air and terrestrial applications that use passive sonar, electronic support measures (ESM) or passive "listening" with seismo-acoustic sensors, respectively, to track an emitting Target. A sequence of bearing measurements can then be used for target motion analysis (TMA) or more specifically to determine the Target's kinematic state. Kolb [79] and Murphy [80] describe the range observability challenges that plague this field of study for a single Surveyor; without calibrated maneuvers by the Surveyor, the Target's position and velocity cannot be determined from a sequence of bearing-only measurements. Lindgren [81] shows that range is unobservable even for a Surveyor moving with a constant velocity in a horizontal plane as would be the case for naval surface applications. Several known velocity variations or "legs" are required of the Surveyor (one leg is insufficient) to converge on a solution. Not all maneuvers assure a unique tracking solution [82]; a maneuver is necessary but not sufficient for range observability. Nardone [83] and Fogel [84] establish rigorous mathematical requirements for range observability in the constant depth case (horizontal plane), whereas Jauffret [85] and Hammel [86] establish range observability requirements in 3-dimensions with broader applications to air and space. The literature is expansive on optimal observer trajectories that meet the aforementioned requirements (e.g., [87][88][89][90][91]).

Aidala [92] theoretically analyses the performance of an extended Kalman filter (EKF) for BOT in the ocean environment using sonar bearings from a radiating Target. Aidala naturally constrains the problem to two dimensions for naval surface operations and elects to use Cartesian coordinates for the state dynamics and measurement models. Without an explicit range measurement, the Surveyor ship must maneuver, and the Target must remain fixed or have a constant velocity. Even with this modus, the EKF exhibits erratic and divergent behavior in the estimate due to premature collapsing of the covariance matrix. Aidala believes that the covariance matrix is ill-conditioned because of the choice of Cartesian coordinates that couple the observable (bearing) and unobservable (range) quantities in the state vector. He reformulates the problem in [93] using modified polar (MP) coordinates which decouple the observable and unobservable components. The MP state vector includes bearing, bearing rate, range rate divided by range, and inverted range—all constrained in a horizontal plane. Aidala's state vector selection is notable because the first three quantities can be obtained without maneuvering the Surveyor and the reciprocal of range minimizes system nonlinearities.[9] The covariance matrix shows good stability in the EKF using MP coordinates, but Aidala is careful to note that range observability still depends on the Surveyor "tactics [maneuvers], environmental disturbances, measurement errors, etc."

One apparent weakness in Aidala's work is the assumption of constant Target motion parameters (either position or velocity). Kirubarajan [96] devised a batch-recursive method that tracks a *maneuvering Target* in clutter with a signal-to-noise ratio (SNR) as low as 8 dB. Le Cadre [97] applies a hidden Markov model (HMM) methodology that maximizes state probabilities conditioned on a sequence of bearing-only observations of a maneuvering Target. Since the Target state is only partially observable from bearing

[9] Similarly, Montiel [94] and Civera [95] use an inverse depth parameter to perform monocular SLAM in an EKF framework. This parameterization enables feature points to be tracked at any range from nearby to infinity. Targets at "infinity" are held in abeyance until sufficient parallax is observed.

measurements, a partially observable Markov decision process (POMDP) was used. The POMDP was solved with dynamical programming or specifically the Viterbi algorithm, but Le Cadre encountered problems with the number of states and decisions that had to be solved. If range and bearing to the Target are known (complete information case), a Markov decision process (MDP) with dynamical programming is said to be “feasible and efficient” for observer trajectory optimization, but this is generally not the case, nor is it the problem.

The amount of work in the field of BOT seems endless and has become a classical problem in nonlinear estimation. Further work can be found pertaining to AUVs (e.g., [98][99][100]), UAVs (e.g., [101][102][103][104]) and other applications (e.g., [105][106]), but the common thread through all of it is this: the kinematic state of the Target under observation by a single Surveyor is unobservable without sufficient calibrated maneuvers or range measurements by the Surveyor.

2.2.2 Line-of-sight Navigation

LOS navigation—a subclass of AON—shall be associated with applications that use visual cues from a sextant, passive electro-optical (EO) camera or infrared (IR) camera to determine the direction to the centroid of a Target. The body of research on LOS navigation mirrors BOT TMA with the same frustrations over range observability. In the space domain, if LOS navigation is used for RPOs under the assumption of linearized relative orbital dynamics, then the Surveyor must maneuver to observe range.[10,11,12] One significant advantage for LOS navigation in space is that the natural relative orbital dynamics for non-maneuvering RSOs are not linear. This is why the orbits of planets and satellites can be determined with angles-only observations from Earth.

Orbit Determination Comparison

LOS navigation shares a methodological heritage with classical and modern celestial mechanics techniques such as Laplace's method [110], Gauss' method [111], Escobal (double-r) [112] and Gooding [113] that use a series of declination and right ascension angles from telescope observations to determine the orbit of a celestial body. Modern recursive estimation techniques that are needed for LOS navigation have also been applied to these orbit determination (OD) methods. Sabol et al. [114] demonstrated angles-only OD with modern sequential filters using high-accuracy angle measurements from a single site.

So why not just apply angles-only orbit determination techniques for navigating during RPOs? There appears to be a range limit at which angles-only OD techniques are viable. Bingham* and Geller found that Gauss' method breaks down at ranges less than about 100 km [115][116][117] due to observations of a short arc length. The arc length appears in the denominator of the solution to Gauss' method, so significant curvature in the orbit is needed for preliminary orbit determination. As range decreases, the ability to observe

[10] "As well known from the literature, the angles-only navigation problem is not fully observable in the absence of maneuvers." [107]

[11] "It is well known that the linearized relative orbital dynamics are not observable with angles-only measurements." [108]

[12] "The relative position/velocity state is not observable during orbital proximity operations when angles-only measurements and Clohessy-Wiltshire dynamics are employed." [109]

curvature in the orbit gets more challenging. This author has also attempted—without success—the methods of Laplace, Gauss and double-r iteration using Vallado's code [118] for coplanar natural motion circumnavigation (NMC) relative orbits within the "Bingham threshold" for nearly circular, low-earth orbits. Code tested at MIT Lincoln Laboratory has encountered a similar range limit [119]. Therefore, 100 km roughly marks the upper range limit for the capability gap that LOS navigation must fill. This author also believes the limit must be a function of the orbital regime since LEO has more curvature at a specified range than at GEO. Further work is merited in this area. Researchers at the Air Force Research Laboratory's Space Vehicles Directorate (AFRL/RV) have recently started pursing this with initial results on the methods of Gauss and Laplace [120]. Unfortunately, their current work does not test trajectories within the Bingham threshold. Lovell has also expressed interest in studying the viability of Gooding for LOS navigation [121].

Optical Navigation Capability Gap for Space Operations

The limitations of angles-only orbit determination techniques within a distance of roughly 100 km marks the upper boundary for the optical navigation capability gap that LOS navigation seeks to fill. We previously found the lower boundary of the gap at approximately 200 m based upon the limitations of stereovision systems on SWaP constrained space systems. When confronted with a non-cooperative Target and a mission that intentionally limits or precludes first party and second party EM emissions, LOS navigation represents the best option to fill this need. Subtle nonlinear dynamical effects persist at these ranges as well as environmental perturbations that are believed to contribute to weakly observable conditions, so not all hope is lost.

The Birth of LOS Navigation in Space

The first application of LOS navigation in space arose during the Gemini program in preparation for lunar orbit rendezvous (LOR) on the Apollo missions. The statement of work for the Apollo guidance and navigation system stipulated that the tasks of guidance, navigation and control had to be performed onboard the spacecraft.[13] The Lunar Module (LM) was equipped with a rendezvous radar system for tracking a transponder on the Command and Service Module (CSM), and the digital data[14] would be processed by the LM guidance computer. If the radar system failed during the LOR,[15] the CSM was equipped with a space sextant (primarily used for calibrating the gyroscopes in the inertial measurement unit) that could be used for LOS navigation to a sunlit target or high intensity flashing lights[16] [122][123].

Two studies in 1959 and 1960 by Levin et al. [124] and Brissenden et al. [125], respectively, showed that astronauts could manually perform rendezvous and docking if provided LOS cues, range and closure rate data from onboard instrumentation. Edwin "Buzz" Aldrin also famously wrote his MIT dissertation on the topic [126].

[13] Additional external data such as a state vector update from Earth could be used "to increase reliability, accuracy, and performance," but the spacecraft had to be able to operate independently with astronauts in the loop [54]. The Jet Propulsion Laboratory had been developing command and control capabilities for interplanetary probes using ground-based RF systems (e.g., the Deep Space Network), but relying on external guidance alone was deemed too risky. The U.S. government was concerned that the Soviet Union might attempt to jam ground-based communication to the Apollo Command Module, so the philosophy of self-dependence shared by strategic weapons guidance systems was embraced for Apollo [54].

[14] The radar provided angle, range, range-rate, and LOS angle-rate data (pp. A-20, [122]).

[15] A radar anomaly occurred on both Gemini XI and XII. On Gemini XII, Lovell and Aldrin had to use manual sextant navigation techniques (pp. 5-121, [130]).

[16] The LM tracking light or beacon light failed on Apollo IX which prevented the Command Module pilot from calculating the backup maneuvers while in darkness (pp. 5-16, [122]).

Wolowicz et al. [127] showed that at ranges less than about 800 meters, astronauts could simply use the apparent size of the rendezvous target to judge range on the terminal phase of the approach. At much greater ranges, where the target appeared as a point of light, Lineberry et al. [128] developed the first techniques for determining the range and range-rate from only LOS measurements with the sextant during a rendezvous *maneuver*. Emphasis is placed on maneuver here since the range was deemed unobservable without thrusting. Lineberry made the assumption of a coplanar orbit for the rendezvousing vehicles and expressed the equations of relative motion in polar coordinates, which conveniently decouple the observable and unobservable coordinates. Analytical expressions were developed from the equations of motion to determine the range and range rate from three different techniques that utilize a combination of measured angles, thrusting and precise timing:

> *One technique required measuring two angular increments traversed by the moving line of sight while coasting, and a third angular increment while a known level of acceleration is applied to arrest the line-of-sight motion. A second technique required measuring the two angles while coasting but not the third angle while thrusting. A third technique required the same measurements as the first technique to be used as inputs to a matrix solution and produced several progressive solutions, but dictated the use of additional equipment for rapid, continuous computing* [128].

The methods of Lineberry and his colleagues were part of the training provided to astronauts in the Rendezvous Docking Simulator at NASA Langley Research Center [123][129].

As fate would have it, Command Pilot James Lovell and Pilot Buzz Aldrin—a student of Battin who had a strong background in LOS navigation from his research at MIT—were the only Gemini and Apollo astronauts that would put the theory of LOS navigation into practice. During their Gemini XII mission in 1966, a rendezvous test with the Agena target vehicle encountered problems when the radar malfunctioned at a range of 120 km following a successful coelliptic maneuver. Lovell and Aldrin reported that the radar lock-on indicator and range and closure rate numbers were "erratic," so they reverted to onboard backup methods where Aldrin took angular measurements with the sextant and Lovell performed terminal phase maneuvers with guidance from pre-populated LOS rendezvous charts to complete three of four docking attempts [130].

The methodologies and experience gained from Gemini paved the way for Apollo. Following Gemini XII, the next rendezvous conducted by the United States occurred on Apollo IX—the first flight of the CSM and LM—to demonstrate a manned lunar-module-active rendezvous in low Earth orbit. The LM's primary guidance system was used for this demonstration with an active and functioning rendezvous radar, but the manual backup techniques for performing the rendezvous (with "mirror image" impulses) from sextant sightings onboard the Command Module were calculated simultaneously and exhibited "excellent agreement" with the LM rendezvous solution (pp. 10-20 [122]). It is unclear if this practice of manually calculating the rendezvous solution became the standard for the remaining Apollo missions, but it is clear that the backup LOS navigation techniques were never needed.

Meanwhile, the U.S. Air Force was pursuing research on LOS navigation for its own purposes. A memo released by the DoD in December 1963 revealed its pursuit of a manned spaceflight program called the Manned Orbital Laboratory (MOL) that would repurpose the Gemini capsule for launch on the Titan III [131]. Relative navigation requirements for MOL and Apollo were a significant driver for the Air Force Avionics Laboratory's sponsored work of Schneider and Prussing [132][133][134] who appear to be the first to have expanded LOS navigation techniques from the planar assumption of Lineberry et al. to include out-of-plane relative motion. Structural differences between Lineberry et al. and Schneider and Prussing are that the latter used Cartesian coordinates and the linearized Clohessy-Wiltshire dynamics whereas the former used polar coordinates and simplified analytical equations from the exact expressions of in-plane radial and normal relative acceleration.

For each set of angular measurements, Schneider and Prussing's technique leads to an in-plane (IP)—the xz-plane (or "sr-plane")—state vector that is normalized by the radial element, z, in the SWR[17] Cartesian set: $[x/z \quad 1 \quad \dot{x}/z \quad \dot{z}/z]^T$. Once the astronauts calculate the normalized state vector with a slide rule (this is how computers worked back then!), the normalized velocities are compared to pre-populated Universal Trajectory Charts (UTC) to uniquely identify the shape parameter, r, of the scaled relative trajectory and a dimensionless time interval, $\bar{\tau}$, that specifies when $z(\tau)$ is a maximum. The shape parameter is the ratio of the extreme values of z, i.e. $r \equiv z_a/z_b$. Using Clohessy-Wiltshire dynamics, the out-of-plane (OOP) motion is a sinusoidal oscillation, so the scaled OOP position and velocity is represented on a phase plane plot. In order to determine the scale of the full six-element state vector, a calibrated maneuver is required.

Schneider and Prussing offer three different procedures for collecting the angle measurements and thrusting:

[17] S represents the along-track direction, W represents the orbit-normal direction, and R represents the radial direction

> *1) the 3-instant OOP method uses three (σ, δ)[18] pairs and a thrust in the y direction performed between the first and third measurement instants; 2) the 4-instant IP method uses four (σ, δ) pairs and a thrust parallel to the $x - z$ plane performed between either the third and fourth or the first and second measurement instants, and 3) the 4-instant OOP method differs only in that the thrust is in the y direction and is performed at a time at which $y = 0$.* [132]

The significance of Schneider and Prussing's technique for its time cannot be overstated. These innovators went to great lengths to simplify the computational burden on the astronauts such that a mere 2×2 matrix inversion was the worst that would be encountered. In 1970, just two years after completing his report for the Avionics Lab [132], Schneider adapted his LOS navigation technique into a set of closed-form algorithms that could be performed by a digital computer [135]. The digital approach requires inversion of a 6×6 matrix—a task over which he still fretted. These seem like trivial matters 47 years later, but it is an illuminating example of how far we have come.

In 1971, Schneider (an MIT graduate and alumnus of the Instrumentation Lab) realized that if he made a slight modification to his closed-form solution to use curvilinear coordinates, a solution was converged upon with an accuracy of 0.001 nautical miles (1.85 m) for downrange separations from 50-400 nautical miles (92.6-740.8 km) [136]. He contends that range need not be initially known at these distances since an iterative process can be used to find it (range is needed in a nonlinear geometry block that transforms LOS angles in Cartesian coordinates to "curvi-angles" in curvilinear coordinates, but it can be initialized with 0). Interestingly, the 92.6 km mark is very near

[18] δ is the in-plane angle and σ is the out-of-plane angle

the 100 km Bingham threshold for classical orbit determination techniques. He makes no statement on the viability of this method to converge at shorter distances—the gap that LOS navigation seeks to fill. This is a worthy topic to revisit based on new clues from Geller and Lovell that will be discussed shortly.

Post-Apollo Progress

Throughout most of the Space Shuttle era, published literature on LOS navigation in space is scant, perhaps due to funding opportunities from the DoD[19] and NASA. The dearth of literature certainly reflects a broader statement about the need or priority at that time for passive navigational capabilities with non-cooperative targets. By the turn of the century, LOS navigation started picking up interest again, but this time motivated by a need for autonomous spacecraft to inspect and/or repair defunct satellites, to deorbit or deflect dangerous debris, to make scientific observations of natural objects, or to carry out sensitive missions for the DoD and intelligence community. The Acting Director of the Defense Advanced Research Projects Agency (DARPA) illuminates these needs at a hearing of the Subcommittee on Emerging Threats and Capabilities of the Senate Armed Services Committee in June 2001 [137].

The trade-space for solving the LOS navigation problem grew considerably in the decades following Apollo. Apollo era solutions had to be simple enough for limited computing and resource constrained human operators, so maneuvering to observe range was completely acceptable. Rendezvous maneuvers had to be performed regardless, but the modern resurgence of the problem is considerably less constrained. Computer processing has

[19] The author is unable to find any relevant reports on the Defense Technical Information Center (DTIC) website in the period after the Apollo Program and prior to 2001.

drastically improved as have digital cameras and space technologies. Solutions that are far more complicated are now possible, so *the modern goal of LOS navigation research is to solve the problem without maneuvering and without augmentation* (no *a priori* knowledge).

Augmented[20] Natural Motion Trajectories with Linear Covariance Analysis

DARPA's Orbital Express mission emerged as a driver for autonomous LOS navigation before its launch in March 2007. Chari* [138] and Geller designed and analyzed potential RPO trajectories that could be used with LOS navigation on the Orbital Express mission, but more generally, showed the effect of natural motion trajectories and maneuver-assisted trajectories on range observability for nearly circular, low-earth orbits. Chari's study, performed at MIT, used a linear covariance (LINCOV) analysis program modeled after Geller's original work for the NASA/JPL Mars Sample Return project. The LINCOV program uses a linearized Kalman filter to propagate the state covariance as opposed to the actual state. The state transition matrix that is used for propagating the covariance is generated by linearizing about a two-body propagated reference trajectory. The output is an estimate of the state uncertainty that, if set up correctly, should match the statistics generated from a Monte Carlo simulation. Chari's instantiation of LINCOV is initialized with *a priori* navigational uncertainties in Target and Surveyor relative position (1% of range) and relative velocity (1/10,000 sec^{-1} of position uncertainty, on the order of 1 cm/sec). *This decision sets it apart from an unaugmented angles-only solution in which nothing is known of the Target.* While studying natural motion trajectories, he limits the other system uncertainties to zero-mean noise in the angle measurements ($3\sigma = 1°$). For

[20] The word "augmented" indicates the use of prior knowledge of the Target's state.

maneuver-assisted trajectories, he also includes uncertainties for un-modelled accelerations, camera alignment (angular measurement biases), attitude determination, and thrust vector.

Chari's analysis of natural motion station-keeping orbits considers co-circular and "football" orbits. Chari claims that range uncertainty with LOS navigation decreases during co-circular station keeping with enough natural cross-track motion. He demonstrates graduated improvements at 87 m (5°), 268 m (15°) and 1 km (45°) cross-track motion of the Surveyor at 1 km downrange from the Target. Due to the oscillatory nature of OOP motion, the relative trajectory is similar to a ship zigzagging with multiple legs as described by Lindgren [81] and van Huyssteen [82] for naval surface operations. Chari shows similar results for station-keeping "football" orbits where the Surveyor performs relative elliptical motion at various distances downrange from the Target. He states that the variation in the radial motion of the Surveyor imitates the previous cross-track motion. As the size of the vertical motion in the ellipse grows, the uncertainty in range decreases more rapidly. Chari claims, "The best results from the station-keeping football tend to occur when the target is enclosed by the football [an NMC trajectory]" (pp. 74).

Chari's analysis of co-elliptic, "traveling football" (differential semi-major axis and eccentricity) and v-bar hops for closure orbits also shows a reduction in range uncertainty in all cases with the best results occurring at closest approach to the Target. He finds that motion perpendicular to the LOS has no meaningful effect during closure unless the degree of perpendicular motion exceeds the already induced angle variants from closure motion.

The output of Chari's simulations only includes the covariance matrix as a function of time (not actual position estimates), but the qualitative merits of his findings were taken as evidence in the present work that an IOD solution and trajectory estimate are possible from natural motion alone.

Augmented, Maneuver-Assisted Trajectories with Linear Covariance Analysis

Chari incorporates maneuvers into the same categories of relative trajectories he explored with purely natural motion in addition to the r-bar hop. He initializes the covariance matrix with a 10% error in range to highlight any improvements from maneuvers. As expected from [128], he finds "using maneuvers to augment natural orbital motion significantly helps navigation filter performance." It is a bit perplexing that he considers cases as close as 10 m where the extent of the Target would be sufficient for SFM and SLAM, but his results are consistent at short and long distances. The most useful lessons from Chari's maneuver-assisted trajectory studies are: 1) thrusting perpendicular to the line-of-sight is better for reducing downrange uncertainty, 2) maneuvering for range observability has diminishing returns, and 3) un-modelled accelerations (process noise) increase downrange uncertainty until counteracted by a maneuver.

As it relates to process noise, Chari and Geller identified IMU accuracy as an important consideration for maneuver-assisted trajectories since some maneuvers use a Δv on the order of mm/sec. For Chari's study, he assumed the Δv for maneuvers were perfectly known, but Jason Schmidt* later studied the effect of accelerometer errors which will be discussed momentarily.

Finally, Chari found that expending more fuel to obtain range observability does not necessarily give a better result over smaller maneuvers (pp. 130). This finding instigated

Chari to identify three "hybrid trajectories" that string together a series of previously studied maneuvers to improve downrange observability during RPOs. The hybrid trajectories are merely qualitative to show that the combination of maneuvers reduces range uncertainty; no optimization work was attempted to determine if these trajectories are fuel optimal or observability optimal. That work is left to fellow MIT alumnus, David Woffinden*[21], whose work will be introduced next.

Augmented, Closed-Loop GN&C Simulations with EKF and Linearized J_2 Dynamics

Subsequent to Chari's findings, Geller produced a LINCOV program that modeled the closed-loop performance of a complete guidance, navigation and control (GN&C) system for satellite inspection and orbital rendezvous [139]. The program showed the effects of different quality sensors, actuators, and algorithms for relative pose and position estimation. An extended Kalman filter is used with a linearized J_2 gravity dynamics model for state propagation. Geller's published simulation results were limited to a v-bar approach from 25 m behind the Target, but the utility of the LINCOV program for longer-range LOS navigation scenarios was demonstrated and described by Woffinden [44]. Woffinden shows the results of a v-bar approach from 1 km behind the Target, but the downrange uncertainty is not reduced until reaching 500 m where the GN&C mode switches to allow pseudorange measurements from the apparent size of the Target just as Wolowicz [127] during the Apollo era. Without the pseudorange measurements, the range is unobservable during v-bar approaches and other rare rendezvous trajectories in which

[21] Advised by Geller for his Master's thesis at MIT and then followed Geller to Utah State in 2004 to complete his PhD. He now works for Draper Laboratory in Houston and consults NASA on angles-only navigation research.

the maneuver sequence does not alter the true LOS measurement profile from the "nominal" LOS measurement profile[22].

Woffinden created a Monte Carlo counterpart to Geller's LINCOV program: a high fidelity, 6 degrees-of-freedom (DOF), closed-loop, GN&C simulator in MATLAB and Simulink for assessing system-level performance during LOS navigation. The sensor, actuator, trajectory and disturbance models mirrored the LINCOV program and produced comparable results. Woffinden's GN&C simulator and EKF successfully demonstrated that precisely known maneuvers, pseudo-range measurements from apparent size of the Target, and triangulation measurements of Target features could be used to determine the Target's relative state [140].

Schmidt*[23] [117][141] built upon Woffinden's work with a similar Simulink GN&C simulation[24] by filling three gaps pertaining to filter stability, thrust accuracy and camera errors in the simulation. The instability of the Kalman filter has been known since the Mariner II mission (see Wiesel [142] pp. 111 for a discussion). LOS navigation without maneuvering will lead to a highly elongated and potentially unbounded covariance ellipsoid in the direction of the LOS vector. The covariance matrix will have extremely large and extremely small eigenvalues, which mean the Kalman covariance can become singular in the presence of highly accurate LOS observations. This crisis is averted with maneuvers that prevent the covariance matrix from becoming unstable, but it has also been addressed in the past with "square-root" filters. Schmidt implemented five different extended Kalman filters to compare their numerical performance for LOS navigation:

[22] The notion of a nominal trajectory is manifested from the dynamics model used with the filter.

[23] Sponsored, in part, by AFRL.

[24] Includes sensor, actuator, and dynamic models with uncertainty parameters that include noise and bias. The simulation can be run open or closed-loop.

conventional, Joseph form, two variants of square-root Kalman filters (Potter and Carlson) and UD factorization. He does not declare a clear winner but does demonstrate the stability of square-root filters over the conventional EKF at ranges less than 500 m.

An alternative method that Schmidt studied for improving filter performance involved thrust accuracy and camera accuracy in the GN&C simulation. By varying error parameters associated with the accelerometer and camera sensor package, he could perform a system-level trade study to determine range errors related to each potential sensor. The accelerometer accuracy was previously ignored by Chari and Woffinden in favor of more process noise or no noise to study the structural problem.

Optimal Observability Maneuvers

One contribution by Woffinden was his research on optimal observability maneuvers [143]. The cost function Woffinden chose for optimal Surveyor maneuvers is a function of the relative position deviation, $\delta\boldsymbol{r}$, from the unperturbed, linearized trajectory or nominal trajectory of the Surveyor. Woffinden shows geometrically and analytically how this deviation relates to the uncertainty in range that he calls the detectability range error metric, $\delta\rho$. The cost function for an optimal maneuver minimizes $\delta\rho$ which is proportional to range and inversely proportional to the sine of an effective observability angle (the angular difference between the true LOS and the unperturbed, linearized LOS or nominal LOS). An effective observability angle of $\pm 90°$ is optimal. In essence, the optimal maneuver balances the two competing interests of closing on the Target while thrusting perpendicular to the nominal LOS as Chari observed.

Woffinden verifies the optimal maneuver cost function analytically, numerically (with his 6DOF simulator) and with LINCOV. He selects a case in which the Surveyor is station-keeping on the v-bar 100 m in front of the Target and shows with all three methods that an impulsive maneuver directed upward in the radial direction is optimal;[25] however, he also clearly demonstrates with LINCOV that any maneuver perpendicular to the Surveyor's velocity vector (no downrange component) reduces downrange uncertainty from 10 m to less than 2.5 cm in this scenario. It should be noted that the LINCOV program only accounts for angle measurement errors ($\sigma = 0.3$ mrad) and the relative position (10% of range) and velocity (orbital rate times initial position uncertainty) errors of both the Target and Surveyor. The acceleration magnitude and direction from thrusting is assumed to be perfectly known.

Observability Criteria

The detectability range error metric identified by Woffinden for optimizing maneuvers is a component of analytical observability criteria that Nardone [83] first published for BOT TMA. Woffinden generalizes Nardone's criteria for a 3-dimensional state transition matrix so that any linearized dynamics model (e.g. Clohessy-Wiltshire) can be used for orbital rendezvous [144]. The closed-form observability criteria are used by Woffinden to demonstrate its utility on five relative orbit scenarios[26] using Clohessy-Wiltshire dynamics and impulsive or continuous thrust maneuvers. Woffinden claims that the criteria may be extended to include nonlinear or disturbance terms in the dynamics.

Relative Orbital Elements with Secular J_2 Dynamics (Augmented or Maneuver-Assisted)

[25] The solution space was constrained by 1) a single impulsive maneuver, 2) restricting changes to the Surveyor's orbital energy or semi-major axis, and 3) the Surveyor cannot intercept the Target or bypass it in the downrange direction.

[26] These include r-bar station keeping, v-bar approach, co-planar circumnavigating "football" trajectory with an impulsive maneuver, arbitrary in-plane trajectories with OOP impulsive maneuver, and arbitrary relative trajectory with an arbitrary impulsive maneuver.

Gaias et al. [145] reconfigured the problem using a different set of coordinates that they call relative orbital elements (ROEs) given by

$$\delta\boldsymbol{\alpha} = \begin{bmatrix} \delta a \\ \delta e_x \\ \delta e_y \\ \delta i_x \\ \delta i_y \\ \delta u \end{bmatrix} = \begin{bmatrix} \delta a \\ \delta e \cos\varphi \\ \delta e \sin\varphi \\ \delta i \cos\theta \\ \delta i \sin\theta \\ \delta u \end{bmatrix} = \begin{bmatrix} (a - a^o)/a^o \\ e\cos\omega - e^o\cos\omega^o \\ e\sin\omega - e^o\sin\omega^o \\ i - i^o \\ (\Omega - \Omega^o)\sin i^o \\ u - u^o \end{bmatrix} \tag{2.3}$$

where the classical orbital elements are represented as usual (a, e, i, ω, Ω and M) and the mean argument of latitude is $u = M + \omega$. The phase angles φ and θ indicate the orientation of the relative trajectory and the magnitude of the $\delta\boldsymbol{e}$ and $\delta\boldsymbol{i}$ indicate the magnitude of the in-plane and out-of-plane (OOP) orbital motion, respectively. The trajectory offsets from the center of the LVLH frame are easily parameterized in the along-track direction by the relative mean longitude $\delta\lambda = \delta u + \delta i_y \cot i$ and in the radial direction by δa. The superscript o designates the Surveyor spacecraft at the origin of the LVLH frame.

In this formulation, the first five elements are claimed to be “mostly” observable since they describe the shape and orientation of the relative orbit whereas the last element is described as the scaling term which is akin to the unobservable range in the traditional sense (“range is well approximated by only one term of the state” [145]). Gaias et al. use the case of equal orbital energy ($\delta a = 0$) and no OOP motion ($\delta i = 0$ and $\delta\Omega = 0$) to support their claim with a linear mapping that shows $a\delta u$ gives the average range during one orbit (pp. 442). It is unclear from their publication if the unobservable range acts through $a\delta u$ for all relative trajectories, particularly those that circumnavigate the Surveyor, or if this is only true for leader-follower configurations (this author believes that may be the case).

The ROEs are very convenient for generating a simple constant coefficient linear system to describe relative motion for circular Keplerian orbits. First-order, secular J_2 effects were added to the A matrix to produce a linear time-varying (LTV) system. The state transition matrix is formed in the usual way and implemented in a nonlinear least squares (NLS) navigation filter. D'Amico[27] claims the original STM and filter in his dissertation [146].

In 2012, Gaias and D'Amico used their LOS navigation filter to support the Advanced Rendezvous demonstration using GPS and Optical Navigation (ARGON) phase of the Swedish Prototype Research Instruments and Space Mission technology Advancement (PRISMA) formation-flying mission. PRISMA consisted of two 3-axis stabilized spacecraft—Mango and Tango—to mature guidance, navigation and control technology for autonomous RPOs. ARGON is one of only two experiments to have *openly* demonstrated LOS navigation in the post-Apollo era, but it was not performed autonomously, and it was not performed without maneuvers. Mango was equipped with a propulsion system so that it could maneuver near Tango [147] and Gaias et al. developed a metric for determining the Δv needed to obtain range observability. The images collected by Mango were downlinked to the ground-station for processing and maneuver commands were sent during up-links to guide Mango from a distance of >30 km to within 3 km of Tango. The navigation solution during the ARGON phase was generated solely from the NLS LOS navigation filter with an accuracy of 1-10% using the GPS solution as the truth.

[27] D'Amico has since moved to Stanford University where he has built a space rendezvous laboratory with the experience and data he obtained from PRISMA.

PRISMA's mission ended in late 2014, but the acquired data is still being analyzed and used to support future missions. Gaias claims, "Post analysis of the flight data collected during ARGON, in fact, presented some minor discrepancies with respect to the expected modeled profiles" [148]. Gaias developed a corrected STM for nearly circular orbits (still with first-order, secular J_2 effects) that fixes the discrepancies uncovered in D'Amico's STM. Gaias also shows how to include the mean differential aerodynamic drag in the ROE time profiles.

The results of ARGON supported the latest LOS navigation experiment called AVANTI (Autonomous Vision Navigation and Target Identification) which was conducted by DLR late last year. The results and lessons learned are only just beginning to surface at conferences and in the literature [149]. AVANTI relied upon good prior knowledge of the target and a cold gas thruster to perform an autonomous spiral approach.

Non-augmented, Maneuver-free LOS Navigation

This section describes the progress that has been made toward maneuver-free LOS navigation by exploiting nonlinear dynamics, natural perturbations, and nonlinear measurement models.

Gaias et al. [145] demonstrated that the simplified J_2 effects in their STM improved the observability properties of the dynamical system, but not sufficiently to solve the problem without maneuvering. The condition number of the observability Gramian (see Section 3.6) is widely accepted as a metric for observability in which Gaias' numerical simulations demonstrate no better than $\mathcal{O}(10^{14})$. Yim, Crassidis and Junkins suggest that a condition number of 10^{16} is the limit for practical observability [150]. In theory, the condition number for the Clohessy-Wiltshire STM is infinite, so high condition numbers indicate weak range observability at best. Note that Gaias used a linear mapping from ROEs to rectilinear coordinates for the observability analysis. Shortly we will see why that matters.

Nonlinear relative dynamics also have been shown to provide weak range observability. Lovell and Lee derived observability criteria for nonlinear dynamics based upon Lie derivatives of the line-of-sight [151]. The authors show analytically and numerically for certain geometric conditions in nearly circular orbits that nonlinear dynamics should render range observable for LOS navigation. This agrees with Chari's LINCOV analysis.

The dynamics model selected for LOS navigation can be formulated with several different types of coordinates such as rectilinear, curvilinear, action-angle, orbital elements, relative orbital elements, etc., but all angle measurements originate from pixel coordinates in a Cartesian frame. A mapping is required to convert from angle measurements to the set of coordinates used by the dynamical model. This is known as the measurement model. Sullivan and D'Amico [152] recently showed the significance of the measurement model on range observability. Sullivan, a student of D'Amico, also uses ROEs, so he looked at six different measurement models[28] that ultimately map the ROEs to rectilinear coordinates. Each of the measurement models were used with Gaias' improved STM (with and without J_2) to calculate the condition number of the Gramian. The best results—condition number of $\mathcal{O}(10^3)$—are obtained with a nonlinear mapping from ROEs, the J_2 STM, and large batches of data (10 orbits, in this case). The worst results—condition

[28] (1) linearly map mean ROEs to rectilinear coordinates; (2) linearly map mean ROEs to curvilinear coordinates, then nonlinearly map the curvilinear coordinates to rectilinear coordinates; (3) nonlinearly map mean ROEs to rectilinear coordinates; (4) map mean ROEs to osculating ROEs, then linearly map osculating ROEs to rectilinear coordinates; (5) map mean ROEs to osculating ROEs, then linearly map osculating ROEs to curvilinear coordinates, then nonlinearly map the curvilinear coordinates to rectilinear coordinates; and (6) map mean ROEs to osculating ROEs, then nonlinearly map osculating ROEs to rectilinear coordinates.

number of $\mathcal{O}(10^{17})$—are obtained with a linear mapping from ROEs, the Keplerian STM, and large batches of data (10 orbits). Sullivan validated these results with a high-fidelity simulation. He built an EKF and evaluated the errors from each of the different measurement models with Gaias' J_2 STM (using ROEs). All six measurement models exhibit a trend toward the solution over five orbits, and as before, the nonlinear mapping performed the best.

There is one major caveat with Sullivan's work; the filter is initialized using *a priori* knowledge about the Target's state, which Sullivan and D'Amico presume will come from a two-line element set (TLE) or radar campaign. This modus may be fine for satellites that are trackable, but without it, the Sullivan-D'Amico approach is insufficient especially as they conclude, "the navigation filters demonstrate strong sensitivity to the initialization error" (see [152] pp. 24). The ultimate objective of maneuver-free LOS navigation must be to solve the problem without external guidance.

2.3 Kolmogorov-Arnold-Moser Theorem

An Application

Most of the work we have reviewed for space rendezvous and proximity operations have used linear time-varying (LTV) plants to describe the relative orbital dynamics with simplifying assumptions that destroy or degrade range observability. Many have even augmented the state transition matrix with first-order effects from the Earth's oblateness. In all cases, the two-body orbital dynamics were used as the starting point before even considering perturbation theory. An alternative approach is to start from a non-Keplerian reference orbit in the full geopotential. This solution is a much better characterization of the long-term bulk motion while differential corrections from the equations of variation effectively describe the local motion of the nonlinear trajectory. Of course, this approach is only possible if an IOD solution can be found without augmentation or maneuvering.

Numerical integration is a tricky and costly solution to the proposed non-Keplerian reference orbit since LOS navigation is characterized by occasionally sparse observations and variable sample rates (reference and actual observations must be perfectly synchronized). A better approach is to use Kolmogorov-Arnold-Moser (KAM) theorem. William Wiesel has shown that Earth satellites in the full geopotential experience lasting quasi-periodic motion on a KAM torus [153][154][155][156]. If constructed correctly, the torus persistently describes an Earth satellite's motion in an Earth rotating reference frame without the need for routine and costly numerical integrations. At any instant in time, the torus describes the satellite dynamics in the expanded geopotential with exquisite accuracy (Wiesel has demonstrated centimeter-level precision for a decade). Therefore, the torus becomes the starting point for estimating local motion.

The Theorem

In his famous 1954 address to the International Congress of Mathematicians, Andrey Kolmogorov first posed the theory that a lightly perturbed, conservative, dynamical system will exhibit lasting quasi-periodic motion on an invariant N-torus [157]. Kolmogorov's student, Vladimir Arnold, and German-American mathematician Jürgen Moser rigorously proved the theory for Hamiltonian systems [158][159]. The theorem that now bears their names provides a method for analyzing the stability of systems and providing conditions that restrict chaos.

In its simplest mathematical representation, KAM theorem is not unlike classical perturbation theory with the initial state governed by

$$\mathcal{H}_{\varepsilon}(I,\theta) = h(I) + \varepsilon f(I,\theta) \tag{2.4}$$

where $\mathcal{H}_{\varepsilon}$ is a perturbed Hamiltonian which is 2π-periodic, h and f are real-analytic functions representing an unperturbed Hamiltonian and a perturbing function, respectively, ε is a small ($\ll 1$) real valued perturbing parameter, and (I,θ) are symplectic action-angle variables on the torus [160]. One distinct difference between classical perturbation theory and KAM theory is the speed at which the solution converges. Using a sequence of canonical transformations, the solution from classical theory is converged upon linearly, if at all. For example, in the first step, the initial Hamiltonian $\mathcal{H}_1 = h_1 + \varepsilon f_1$ can be transformed to $\mathcal{H}_2 = h_2 + \varepsilon^2 f_2$ in which the order of the perturbation grows linearly to ε^2. On the j^{th} iteration, the Hamiltonian is of the form $\mathcal{H}_j = h_j + \varepsilon^j f_j$ and the perturbation has grown to ε^j [161]. With every iteration, the denominator of the generating function can grow arbitrarily small, causing a divergence from the solution with higher orders of ε. This is the well-known small divisor problem that plagued mathematicians such as Henri Poincaré [162]. Kolmogorov's theorem overcomes the small divisor problem by converging upon the solution quadratically such that after the j^{th} iteration the Hamiltonian is of the form $\mathcal{H}_j = h_j + \varepsilon^{2^{j-1}} f_j$. This approach controls the small divisor in the sequence of canonical transformations so that infinitely many iterations may be used [161].

This super-convergent analytical technique was used to prove KAM theorem and show that solutions to a non-degenerate Hamiltonian will persist on an invariant torus as long as the perturbations remain small—an entirely new approach to perturbation theory. Classical perturbation theory seeks to approximate the solutions and then explore its evolution/stability from a fixed initial condition, whereas KAM theorem does not concern itself with the motion incurred from preassigned initial conditions, but instead explores the dynamic stability in phase space using a set of fixed frequencies that govern quasi-periodic motion [161]. In the integrable case when $\varepsilon = 0$, the phase space solution will lie on an invariant torus with a set of N fundamental frequencies. When ε is sufficiently small and the frequencies are sufficiently incommensurate (satisfying the diophantine inequality), a solution is quickly converged upon that remains on the invariant torus (a condition of stability). As ε grows, the torus is deformed until it ceases to exist [161].

Chapter 3

Line-of-Sight Navigation Fundamentals

In this chapter, we describe the basic process for making angle observations in the camera frame and mapping the observations into the LVLH frame using a measurement model. This process is not new, but it forms a foundation from which our novel insights and machinery in later chapters are formed. A pinhole camera model is assumed for the measurements. This is a reasonable assumption if the radial distortion from the star tracker or camera lens has been corrected with an optical calibration.

3.1 Observations in the Camera Frame

Observations originate from image pixel coordinates to describe the location of the Target in the camera frame. The measurements can be expressed with spherical coordinates, cylindrical coordinates or Cartesian coordinates in curvilinear or rectilinear form. Spherical coordinates are employed in this work. The spherical coordinate terminology of azimuth and elevation are generally used in the tradition of ground-based observations, but we avoid this language and strictly refer to these angles by the Greek symbols α and β as shown in the camera frame of Figure 7. Generally, references to α and β throughout the thesis are referring to angles in Hill's frame, not the camera frame, so the superscript o has been added for clarity here.

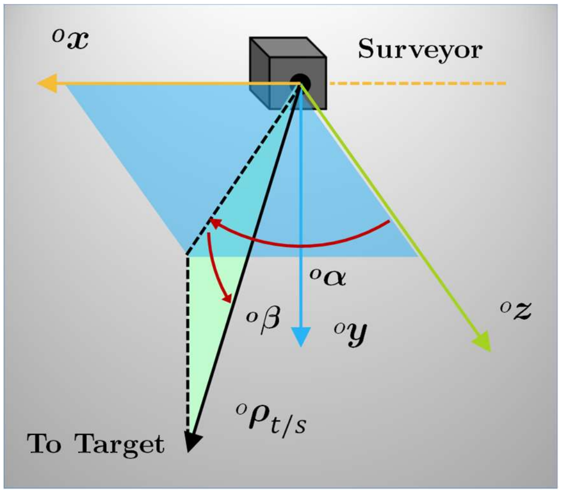


Figure 7: Spherical Coordinates in Camera frame. The camera frame is defined with the z-axis perpendicular to the image plane pointing toward the worldview. The x-axis and y-axis are parallel to the image plane coordinates.

Equations (3.1) and (3.2) define the angles with respect to the Cartesian coordinates of the Target in the camera frame. While the coordinates $({}^o x,\ {}^o y,\ {}^o z)$ are unknown, the ratios ${}^o x/{}^o z$ and ${}^o y/\sqrt{{}^o x^2 + {}^o z^2}$ can be found.

$$
\begin{aligned}
{}^o\alpha &= \tan^{-1}\left(\frac{{}^o x}{{}^o z}\right) && (3.1)\\
\sin {}^o\beta &= \frac{{}^o y}{\rho}\\
\cos {}^o\beta &= \frac{\sqrt{{}^o x^2 + {}^o z^2}}{\rho}\\
{}^o\beta &= \tan^{-1}\left(\frac{{}^o y}{\sqrt{{}^o x^2 + {}^o z^2}}\right) && (3.2)
\end{aligned}
$$

The ratios ${}^o x/{}^o z$ and ${}^o y/\sqrt{{}^o x^2 + {}^o z^2}$ are found from mapping coordinates on the image plane to the camera frame. Start by computing the centroid of the Target in the image and then counting pixels to the centroid similar to the image plane mapping depicted in Figure 8. At large distances, the image of the target could be a blob or point of light with no structure to be discerned, but the same could be true of a smaller object at shorter distances. Whatever the case may be, the Target size and range are ambiguous no matter how good the camera resolution is.

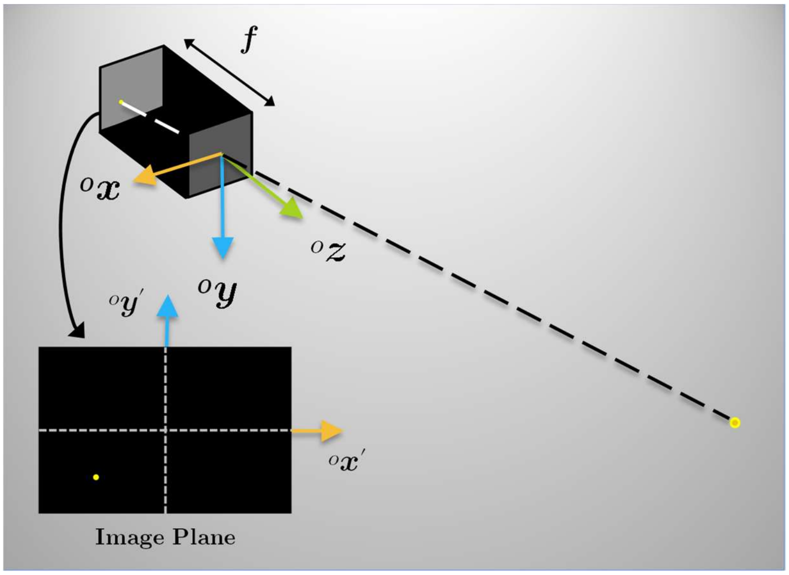


Figure 8: Image plane mapping. The 3D scene in the camera frame is mapped onto a 2D image with perspective projection and a 180° rotation.

The raw pixel count is then converted to distance units using the known dimension of pixels called pixel pitch. The resulting image plane coordinates $({}^o x', {}^o y')$ can now be mapped to the desired angle measurements in the camera frame.

In Figure 8, notice that the image plane coordinates $({}^o x', {}^o y')$ are rotated 180° from the camera frame coordinates $({}^o x, {}^o y)$. The pinhole camera model conveys this relationship mathematically, which is a first-order mapping from a three-dimensional scene to a two-dimensional image [56].

$$\frac{-{}^o x'}{f} = \frac{{}^o x}{{}^o z} \tag{3.3}$$

$$\frac{-{}^o y'}{f} = \frac{{}^o y}{{}^o z} \tag{3.4}$$

where f is the known focal length of the camera. The size of the objects in the image depends entirely on their distance from the focal point in the world since the camera focal length remains constant.

We can solve for the angles ${}^o\alpha$ and ${}^o\beta$ in the camera frame by using the pinhole camera model and the measured image plane coordinates.

$$ {}^o\alpha = \tan^{-1}\left(\frac{-{}^o x'}{f}\right) \tag{3.5} $$

$$ {}^o\beta = \tan^{-1}\left(\frac{-{}^o y'}{\sqrt{(-{}^o x')^2 + f^2}}\right) \tag{3.6} $$

The relative position vector in the camera frame is

$$ {}^{\boldsymbol{o}}\boldsymbol{\rho}_{t/s} = \rho \begin{bmatrix} \sin({}^o\alpha)\cos({}^o\beta) \\ \sin({}^o\beta) \\ \cos({}^o\alpha)\cos({}^o\beta) \end{bmatrix} \tag{3.7} $$

The LOS unit vector in the camera frame is

$$ {}^{\boldsymbol{o}}\hat{\boldsymbol{\rho}}_{t/s} = \frac{{}^{\boldsymbol{o}}\boldsymbol{\rho}_{t/s}}{\rho} = \begin{bmatrix} \sin({}^o\alpha)\cos({}^o\beta) \\ \sin({}^o\beta) \\ \cos({}^o\alpha)\cos({}^o\beta) \end{bmatrix} \tag{3.8} $$

where the subscript t/s indicates the unit vector points from the Surveyor to the Target ("Target with respect to Surveyor"). The next step is to transform the LOS unit vector into the relative frame of motion.

3.2 Relative Frame of Motion

The relative frame that is employed in this work is a local vertical local horizontal (LVLH) frame that is always anchored to the Surveyor. No matter where the Surveyor is located in the inertial frame, the LVLH frame is exactly centered on the Surveyor with the radial vector from the Earth-Centered Inertial (ECI) frame passing through the Surveyor, which defines the local vertical direction that we call the r-axis. The r-axis is the first basis vector in the LVLH orthogonal set, so it can be thought of as x in Cartesian space. Skip the second basis vector for a moment; we will come back to it. The third basis vector or the w-axis is orthogonal to the Surveyor's orbital plane. Even as the orbit osculates, the w-axis direction changes such that it is always perpendicular to the orbital plane and therefore perpendicular to the radial vector. Since the w-axis is the third basis vector, it can be thought of as z in Cartesian space. That leaves us with the second basis vector or the s-axis which must complete the orthogonal set. In the case of a circular orbit, the s-axis is exactly aligned with the velocity vector, but in the case of non-zero eccentricity, the s-axis crudely represents the direction of the velocity vector. For this reason, the s-axis is referred to as the along-track direction. Similarly, the w-axis is referred to as the cross-track direction, and the r-axis is referred to as the radial direction. It is common to refer to this system as the RSW frame or Hill's frame, but we will use the latter throughout this work.

Hill's frame is depicted in Figure 9 with the Surveyor at the center and the Target depicted along a line of site vector $\hat{\boldsymbol{\rho}}$

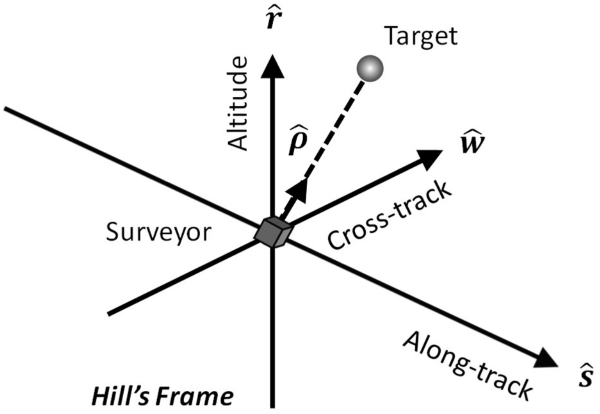


Figure 9: Hill's frame. The Cartesian basis set consists of the r-axis, s-axis and w-axis and can be mapped as $[\mathrm{r}, s, w] \rightarrow [x, y, z]$; although, it is also common

to see $[s, w, \mathrm{r}] \rightarrow [x, y, z]$. The r-axis is aligned with the radial direction, which passes directly through the Surveyor from the origin of the ECI frame. The s-axis is nearly aligned with the velocity vector (exact for circular orbits). The w-axis is aligned with the orbit angular momentum vector, which is perpendicular to the orbital plane.

The basis vectors for Hill's frame are mathematically expressed as

$$\hat{\mathbf{r}} = \frac{{}^{i}\mathbf{r}}{|{}^{i}\mathbf{r}|} \tag{3.9}$$

$$\widehat{\boldsymbol{w}} = \widehat{\boldsymbol{H}} = \frac{{}^{i}\mathbf{r} \times {}^{i}\mathbf{v}}{|{}^{i}\mathbf{r} \times {}^{i}\mathbf{v}|} \tag{3.10}$$

$$\hat{\boldsymbol{s}} = \widehat{\boldsymbol{w}} \times \hat{\mathbf{r}} \tag{3.11}$$

where the superscript i indicates vectors in the ECI frame.

The 3 × 3 rotation matrix that transforms coordinates from Hill's frame to the ECI frame is simply

$$\boldsymbol{R}^{ih} = [\hat{\mathbf{r}} \quad \hat{\boldsymbol{s}} \quad \widehat{\boldsymbol{w}}] \tag{3.12}$$

where the superscript ih represents a transformation from Hills' frame (h) to the inertial frame (i). The seemingly reversed order (ih instead of hi) reminds us that the vector in Hill's frame appears to the right of the matrix when applied correctly. $\boldsymbol{R}^{ih}$ is applied to an LOS unit vector $({}^{h}\hat{\mathbf{r}}_{t/s})$ in the same way that it is applied to the Target's position vector $({}^{h}\mathbf{r}_{t/s})$ in the moving frame.

$${}^{i}\mathbf{r}_{t/s} = \boldsymbol{R}^{ih}\,{}^{h}\mathbf{r}_{t/s} \tag{3.13}$$

$${}^{i}\hat{\mathbf{r}}_{t/s} = \boldsymbol{R}^{ih}\,{}^{h}\hat{\mathbf{r}}_{t/s} \tag{3.14}$$

Take the transpose of $\boldsymbol{R}^{ih}$ to produce the rotation matrix that converts a vector in the inertial frame to Hill's frame

$$\boldsymbol{R}^{hi} = \boldsymbol{R}^{ih^T} \tag{3.15}$$

Relative velocities cannot be transformed the same way since the rate of change of the relative position is a different quantity when viewed from the inertial frame. The transport theorem describes the Target's relative velocity in the inertial frame:

$$^{i}\mathbf{v}_{t/s} = {}^{h}\mathbf{v}_{t/s} + \boldsymbol{\omega}_{h/i} \times {}^{h}\mathbf{r}_{t/s} \tag{3.16}$$

The Target's inertial velocity in the inertial frame is then

$$^{i}\mathbf{v}_{t} = {}^{i}\mathbf{v}_{s} + {}^{i}\mathbf{v}_{t/s} \tag{3.17}$$

$$^{i}\mathbf{v}_{t} = {}^{i}\mathbf{v}_{s} + {}^{h}\mathbf{v}_{t/s} + \boldsymbol{\omega}_{h/i} \times {}^{h}\mathbf{r}_{t/s} \tag{3.18}$$

Do not be attempted to apply the rotation matrix to the relative velocities unless you wish to prove to yourself that ${}^{i}\mathbf{v}_{t/s} \neq \boldsymbol{R}^{ih}\,{}^{h}\mathbf{v}_{t/s}$.

The methodologies that are proposed require a highly accurate $\boldsymbol{R}^{ih}$ matrix. Range observability challenges persist even with perfect knowledge of the Surveyor's state and perfect observations. GPS is likely the best current option for obtaining $\boldsymbol{R}^{ih}$, especially if the Surveyor's inertial position and velocity must be known in near real-time. Alternatives such as laser ranging, Doppler ranging and vision-based triangulation from a catalog of known satellites or landmarks may be considered, but the uncertainty in the Surveyor state estimate should be kept as small as possible.

3.3 Transforming Observations to Hill's Frame

Observations of the Target with respect to the Surveyor's camera frame are converted to an LOS unit vector in Hill's frame, ${}^{h}\hat{\boldsymbol{\rho}}_{t/s}$, with a string of known rotation matrices.

$$^{h}\hat{\boldsymbol{\rho}}_{t/s} = [\boldsymbol{R}^{hi}][\boldsymbol{R}^{ib}][\boldsymbol{R}^{bo}]\,{}^{o}\hat{\boldsymbol{\rho}}_{t/s} \tag{3.19}$$

where ${}^{o}\hat{\boldsymbol{\rho}}_{t/s}$ is the LOS unit vector of the Target with respect to Surveyor in the camera frame, $\boldsymbol{R}^{hi}$ is the rotation matrix from the ECI frame to Hill's frame (as calculated in Section 3.2), $\boldsymbol{R}^{ib}$ is the rotation matrix from the Surveyor's body frame to the ECI frame (this can be found with a star tracker that has a known orientation in the body frame), and $\boldsymbol{R}^{bo}$ is the rotation matrix from the camera frame to the body frame (also known from the engineering design). Throughout this body of work, there is an occasional need for the LOS unit vector from the Surveyor to the Target in the inertial frame, ${}^{i}\hat{\boldsymbol{\rho}}_{t/s}$. It is just as easily found by leaving off $[\boldsymbol{R}^{hi}]$ in Equation (3.19).

The LOS unit vector from the Surveyor to the Target in Hill's frame can also be expressed as

$$ {}^{h}\hat{\boldsymbol{\rho}}_{t/s} = \begin{bmatrix} x/\rho \\ y/\rho \\ z/\rho \end{bmatrix} \tag{3.20} $$

Hereafter, we will drop the term "unit" and refer to this vector as the LOS vector. The normalized components of Equation (3.20) supply the necessary inputs for calculating the angles α and β in Hill's frame

$$ \alpha = \tan^{-1}\left(\frac{y/\rho}{x/\rho}\right) \tag{3.21} $$

$$ \beta = \tan^{-1}\left(\frac{z/\rho}{\sqrt{(x/\rho)^2 + (y/\rho)^2}}\right) \tag{3.22} $$

We can now write the relative position vector in Hill's frame as a function of α, β and ρ

$$ {}^{h}\boldsymbol{\rho}_{t/s} = \rho \begin{bmatrix} \cos(\alpha)\cos(\beta) \\ \sin(\alpha)\cos(\beta) \\ \sin(\beta) \end{bmatrix} \tag{3.23} $$

An analytical expression for the relative velocity vector is found from direct differentiation of Equation (3.23)

$$
{}^{h}\mathbf{v}_{t/s} = \begin{bmatrix} \dot{\rho}\cos(\alpha)\cos(\beta) - \rho\cos(\alpha)\sin(\beta)\,\dot{\beta} - \rho\sin(\alpha)\cos(\beta)\,\dot{\alpha} \\ \dot{\rho}\sin(\alpha)\cos(\beta) - \rho\sin(\alpha)\sin(\beta)\,\dot{\beta} + \rho\cos(\alpha)\cos(\beta)\,\dot{\alpha} \\ \dot{\rho}\sin(\beta) + \rho\cos(\beta)\,\dot{\beta} \end{bmatrix} \tag{3.24}
$$

If $\dot{\rho}/\rho$ is an observable quantity from angles alone, then the range-scaled version of ${}^{h}\mathbf{v}_{t/s}$ can be found

$$
\frac{{}^{h}\mathbf{v}_{t/s}}{\rho} = \begin{bmatrix} \frac{\dot{\rho}}{\rho}\cos(\alpha)\cos(\beta) - \cos(\alpha)\sin(\beta)\,\dot{\beta} - \sin(\alpha)\cos(\beta)\,\dot{\alpha} \\ \frac{\dot{\rho}}{\rho}\sin(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta)\,\dot{\beta} + \cos(\alpha)\cos(\beta)\,\dot{\alpha} \\ \frac{\dot{\rho}}{\rho}\sin(\beta) + \cos(\beta)\,\dot{\beta} \end{bmatrix} \tag{3.25}
$$

Using the same logic as the radial expression in the inertial frame $\mathbf{r} \cdot \dot{\mathbf{r}} = \mathrm{r}\dot{\mathrm{r}}$, the relative velocity projected onto the LOS vector is the range rate

$$
\frac{{}^{h}\mathbf{r}_{t/s}}{\rho} \cdot {}^{h}\mathbf{v}_{t/s} = \dot{\rho} \tag{3.26}
$$

$$
{}^{h}\mathbf{r}_{t/s} \cdot {}^{h}\mathbf{v}_{t/s} = \rho\dot{\rho} \tag{3.27}
$$

A little algebra reveals the exact equation for $\dot{\rho}/\rho$

$$
\frac{\dot{\rho}}{\rho} = \frac{{}^{h}\mathbf{r}_{t/s} \cdot {}^{h}\mathbf{v}_{t/s}}{\rho^2} = \frac{x\dot{x} + y\dot{y} + z\dot{z}}{\rho^2} = \frac{x\dot{x} + y\dot{y} + z\dot{z}}{x^2 + y^2 + z^2} = \frac{\frac{x}{x}\frac{\dot{x}}{x} + \frac{y}{x}\frac{\dot{y}}{x} + \frac{z}{x}\frac{\dot{z}}{x}}{(\frac{x}{x})^2 + \left(\frac{y}{x}\right)^2 + (\frac{z}{x})^2}
$$

$$
\frac{\dot{\rho}}{\rho} = \frac{\frac{\dot{x}}{x} + \frac{y}{x}\frac{\dot{y}}{x} + \frac{z}{x}\frac{\dot{z}}{x}}{1 + \left(\frac{y}{x}\right)^2 + (\frac{z}{x})^2} \tag{3.28}
$$

If OOP motion is negligible ($z \approx \dot{z} \approx 0$), the normalized relative state vector is approximately ${}^{h}\boldsymbol{X}_{t/s}/x \approx [1 \quad y/x \quad 0 \quad \dot{x}/x \quad \dot{y}/x \quad 0]^T$ and can be found using Schneider's method [132] for in-plane normalized state determination. Our methods do not require knowledge of $\dot{\rho}/\rho$, but it is an interesting quantity to note since it can be found without maneuvers in this case.

3.4 Angle-Rate and Range-Rate

The angle-rates of relative motion are calculated numerically. We choose to use the five-point central difference formula given by Equation (3.29) and four-point forward and backward differentiation at the beginning and end of the data sets—Equations (3.30) and (3.31), respectively [164]. The last term in each equation is the estimation error at time c. The equations are shown for $\dot{\alpha}$, but $\dot{\beta}$ is calculated in the same way.

$$\dot{\alpha}(t_0) \cong \frac{\alpha(t_0 - 2\Delta t) - 8\alpha(t_0 - \Delta t) + 8\alpha(t_0 + \Delta t) - \alpha(t_0 + 2\Delta t)}{12\Delta t} + \frac{1}{30}\Delta t^4 \alpha^{(5)}(c) \quad (3.29)$$

where $c \in [t_0 - 2\Delta t,\ t_0 + 2\Delta t]$.

$$\dot{\alpha}(t_0) \cong \frac{-11\alpha(t_0) + 18\alpha(t_0 + \Delta t) - 9\alpha(t_0 + 2\Delta t) + 2\alpha(t_0 + 3\Delta t)}{6\Delta t} - \frac{1}{4}\Delta t^3 \alpha^{(4)}(c) \quad (3.30)$$

where $c \in [t_0,\ t_0 + 3\Delta t]$.

$$\dot{\alpha}(t_0) \cong \frac{11\alpha(t_0) - 18\alpha(t_0 - \Delta t) + 9\alpha(t_0 - 2\Delta t) - 2\alpha(t_0 - 3\Delta t)}{6\Delta t} - \frac{1}{4}\Delta t^3 \alpha^{(4)}(c) \quad (3.31)$$

where $c \in [t_0 - 3\Delta t,\ t_0]$.

Little extra effort is required to analytically calculate the range rate, $\dot{\rho}$, and the angular rates, $\dot{\alpha}$ and $\dot{\beta}$, if the relative state vector is known. This is usually not the case, but it can be done for simulated truth data to verify the angular rates calculated from numerical methods.

The scalar range rate and angular rates can be expressed as a function of the relative position and velocity Cartesian coordinates in Hill's frame

$$\dot{\rho} = \frac{{}^{h}\mathbf{r}_{t/s} \cdot {}^{h}\mathbf{v}_{t/s}}{\rho} \quad (3.32)$$

$$\dot{\alpha} = \frac{\dot{z}x - \dot{x}z}{-x^2 - z^2} \quad (3.33)$$

$$\dot{\beta} = \frac{\dot{y} - \dot{\rho}(y/\rho)}{\sqrt{z^2 + x^2}} \quad (3.34)$$

In situations where initial knowledge of the Target's orbit is known, we could analytically calculate all of the above with some uncertainty at the instant we initialize LOS navigation. The problem then would be to manage and reduce the uncertainty. With such information, the problem is less potent since it eliminates the difficult task of IOD.

3.5 Measurement Model Basics

A measurement model or measurement equation maps angle measurements from one coordinate system to another. For instance, Equation (3.21) and (3.22) are measurement models that expresses the exact nonlinear relationship between angle observations in spherical coordinates and the Target's relative state in Cartesian coordinates. Linearization of the measurement model for use with some modern estimation techniques (e.g. the Kalman filter and nonlinear least squares) degrades the weak range observability contributed by the nonlinearities just as with linearized dynamics models. The loss incurred from linearization can be managed with observations that are frequent enough and accurate enough for the linearization to closely approximate the nonlinear model.

In this section, we explore the magnitude of the linearization error from spherical coordinates to Cartesian coordinates (our choice for this effort). A measurement model in the camera frame—Equation (3.8)—is used here for basic intuition since it is devoid of any rotation matrices from Equation (3.19). The actual measurement model used in our estimator is developed in Section 5.5.2.

Let the measurement vector be

$$\boldsymbol{M} = [\alpha \quad \beta]^T \tag{3.35}$$

The sensitivity of the relative position due to small changes or residuals in angular measurements can be approximated in the neighborhood of an arbitrary point using a Taylor series expansion

$$\boldsymbol{x}(\boldsymbol{M}+\delta\boldsymbol{M}) \approx \boldsymbol{x}(\boldsymbol{M}) + \delta\boldsymbol{M}^T\boldsymbol{g}(\boldsymbol{M}) + \frac{1}{2}\delta\boldsymbol{M}^T\boldsymbol{G}(\boldsymbol{M})\delta\boldsymbol{M} + \cdots \tag{3.36}$$

where $\boldsymbol{x}$ is a coordinate describing the system state, g is the gradient of $\boldsymbol{x}$ and $\boldsymbol{G}$ is the Hessian or second derivative of $\boldsymbol{x}$. The residuals are defined in the usual sense

$$\delta\boldsymbol{M} = \begin{bmatrix}\delta\alpha \\ \delta\beta\end{bmatrix} = \begin{bmatrix}\alpha_{\text{obs}} - \alpha_{\text{ref}} \\ \beta_{\text{obs}} - \beta_{\text{ref}}\end{bmatrix} \tag{3.37}$$

where the subscript *obs* indicates an observed angle and the subscript ref indicates a reference angle from the expected trajectory.

The gradient is

$$\boldsymbol{g} = \left(\frac{\partial\boldsymbol{x}}{\partial\boldsymbol{M}}\right)^T = \begin{bmatrix}\partial\boldsymbol{x}/\partial M_1 \\ \vdots \\ \partial\boldsymbol{x}/\partial M_n\end{bmatrix} \tag{3.38}$$

The Hessian is

$$\boldsymbol{G} = \begin{bmatrix}\partial^2\boldsymbol{x}/\partial M_1^2 & \cdots & \partial^2\boldsymbol{x}/\partial M_1\partial M_n \\ \vdots & \ddots & \vdots \\ \partial^2\boldsymbol{x}/\partial M_n\partial M_1 & \cdots & \partial^2\boldsymbol{x}/\partial M_n^2\end{bmatrix} \tag{3.39}$$

The sensitivity of the relative position (to second-order) due to small changes in angular measurements is

$$\Delta\boldsymbol{x} \approx \delta\boldsymbol{M}^T\boldsymbol{g}(\boldsymbol{M}) + \frac{1}{2}\delta\boldsymbol{M}^T\boldsymbol{G}(\boldsymbol{M})\delta\boldsymbol{M} \tag{3.40}$$

Apply the Taylor series to each element of the relative position vector in the camera frame from Equation (3.8). See Appendix A for the derivations.

$$\begin{aligned}\delta x \approx{}& \rho\cos(\alpha)\cos(\beta)\,\delta\alpha - \rho\sin(\alpha)\sin(\beta)\,\delta\beta \\ &+\frac{1}{2}[-\rho\sin(\alpha)\cos(\beta)\,\delta\alpha^2 - 2\rho\cos(\alpha)\sin(\beta)\,\delta\alpha\delta\beta \\ &-\rho\sin(\alpha)\cos(\beta)\,\delta\beta^2]\end{aligned} \tag{3.41}$$

$$\delta y \approx \rho\cos(\beta)\,\delta\beta - \frac{1}{2}\rho\sin(\beta)\,\delta\beta^2 \tag{3.42}$$

$$\begin{aligned}\delta z \approx{}& -\rho\sin(\alpha)\cos(\beta)\,\delta\alpha - \rho\cos(\alpha)\sin(\beta)\,\delta\beta \\ &+\frac{1}{2}[-\rho\cos(\alpha)\cos(\beta)\,\delta\alpha^2 + 2\rho\sin(\alpha)\sin(\beta)\,\delta\alpha\delta\beta \\ &-\rho\cos(\alpha)\cos(\beta)\,\delta\beta^2]\end{aligned} \tag{3.43}$$

As long as $\delta\boldsymbol{M}$ is small, second order terms in the sensitivity equations are negligible and the remaining system of linear equations can be written in matrix form where $\partial\boldsymbol{X}/\partial\boldsymbol{M}$ is just the Jacobian of the measurement model

$$\delta\boldsymbol{X} = \frac{\partial\boldsymbol{X}}{\partial\boldsymbol{M}}\delta\boldsymbol{M} \tag{3.44}$$

$$\frac{\partial\boldsymbol{X}}{\partial\boldsymbol{M}} = \rho\begin{bmatrix}\cos(\alpha)\cos(\beta) & -\sin(\alpha)\sin(\beta) \\ 0 & \cos(\beta) \\ -\sin(\alpha)\cos(\beta) & -\cos(\alpha)\sin(\beta)\end{bmatrix} \tag{3.45}$$

In standard practice, $\partial\boldsymbol{M}/\partial\boldsymbol{X}$ is called the measurement sensitivity matrix, but we have intentionally inverted the matrix here by posing the measurement model backwards $\boldsymbol{X}(\boldsymbol{M})$ rather than $\boldsymbol{M}(\boldsymbol{X})$.

The second-order terms in Equations (3.41), (3.42) and (3.43) are useful as analytical metrics to assess observability degradation when using the linearized measurement model. This will occur when the residuals get too large.

3.5.1 Sensitivity to Residuals

Range uncertainty can be manifested in several ways from sensor biases and noise, Surveyor absolute navigation errors, dynamics model fidelity, and linearization of the measurement and dynamics models in the GN&C system. All sources of error contribute to the angle residuals that are mapped with the measurement model. In some sense, the α and β residuals are like solid angles subtended from the camera as depicted in Figure 10 for a case when the Target is located along the optical axis ($\alpha = \beta = 0$). The measurement model converts the residual angles into residual position coordinates, $\delta\boldsymbol{X}$.

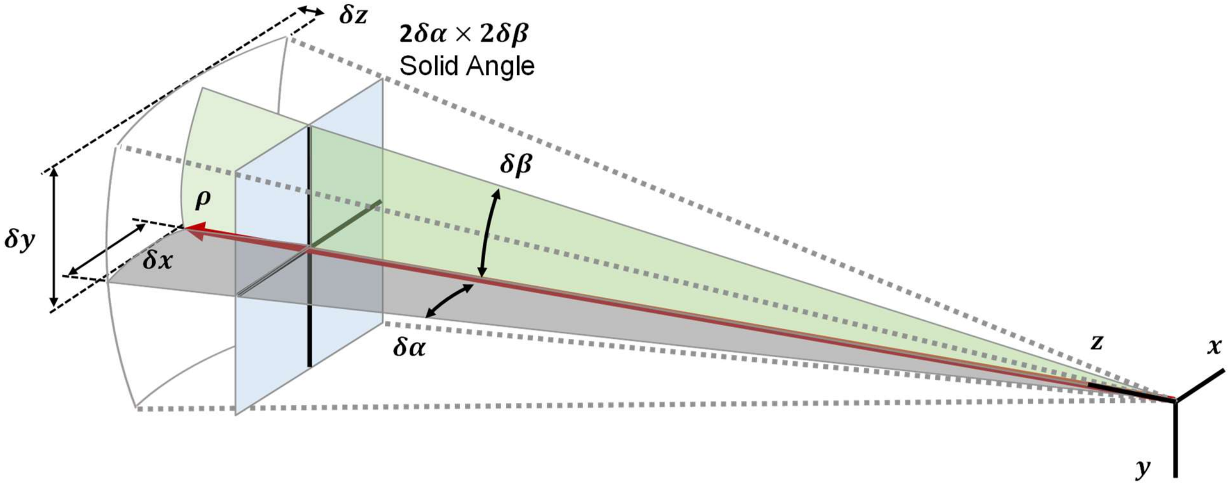


Figure 10: Measurement Sensitivity Geometry. The angle residuals are mapped to relative position uncertainty in the region subtended by the solid angle $\delta\alpha \times \delta\beta$.

For the case of $\alpha \approx \beta = \mathcal{O}(\varepsilon)$ which represents an attempt to keep the Target centroid in the center of the image frame, the nonlinear measurement model simplifies substantially with small angle assumptions

$$\delta x \approx \rho\delta\alpha + \mathcal{O}(\rho\varepsilon^2\delta\alpha, \rho\varepsilon^2\delta\beta, \rho\varepsilon\delta\alpha^2, \rho\varepsilon\delta\beta^2, \rho\varepsilon\delta\alpha\delta\beta) \tag{3.46}$$

$$\delta y \approx \rho\delta\beta + \mathcal{O}(\rho\varepsilon^2\delta\beta, \rho\varepsilon\delta\beta^2) \tag{3.47}$$

$$\delta z \approx -\rho\varepsilon\delta\alpha - \rho\varepsilon\delta\beta + \mathcal{O}(\rho\varepsilon^3\delta\alpha, \rho\varepsilon^3\delta\beta, \rho\delta\alpha^2, \rho\delta\beta^2, \rho\varepsilon^2\delta\alpha\delta\beta) \tag{3.48}$$

Notice that most of the sensitivity effects are manifested in the x and y directions where the analytical expressions are similar in nature to the spatial resolution equation of the camera described in Appendix B. As long as the residuals remain small, δz is $\mathcal{O}(\rho\varepsilon\delta\alpha, \rho\varepsilon\delta\beta)$ and since the range is effectively along the z-axis in this centroiding case, the range sensitivity to the higher order terms is *very* low. In other words, the higher order terms have very little effect on range observability.

A corollary from the centroiding case is that the uncertainty in the state estimate is inversely related to the measurement sensitivity. This fundamental observation is confirmed with the covariance matrix, which is a function of the inverse sensitivity matrix. Larger component values in the measurement sensitivity matrix means smaller values in the covariance matrix. As we have just seen, the centroiding case is conducive for maximizing δx and δy while minimizing δz which leads to range observability problems.

One immediate question that comes to mind is how the measurement sensitivity changes when the Target centroid is not precisely co-located along the optical axis. If minimal slewing is desired, the Target must be allowed to move across a significant portion of the image plane. There appears to be a gap in the LOS navigation literature pertaining to measurement sensitivity from tracking objects that are off-center from the optical axis. Our methodologies developed in Chapter 5 assume continuous slewing to keep the Target centered in the image plane, but is this a wise choice? The gap in knowledge of measurement sensitivity from off-axis tracking is explored here to see if there is a need for a change in the concept of operations (CONOPS) to maximize measurement sensitivity along the LOS.

Let us introduce a measurement model sensitivity metric, δs [29]

$$\delta s = \sqrt{\delta x^2 + \delta y^2 + \delta z^2} \tag{3.49}$$

Plugging in the nonlinear analytical expressions for δx, δy, and δz that were derived in Equations (3.41), (3.42) and (3.43), the range normalized value of δs is calculated as a function of α and β for off-center tracking

$$\begin{aligned}\left(\frac{\delta s}{\rho}\right)^2 = {} & \frac{\delta\beta^2(2\cos\beta - \delta\beta\sin\beta)^2}{4} \\ & + \Bigg(\frac{\cos\alpha\cos\beta\,\delta\alpha^2}{2} - \sin\alpha\sin\beta\,\delta\alpha\delta\beta + \sin\alpha\cos\beta\,\delta\alpha + \frac{\cos\alpha\cos\beta\,\delta\beta^2}{2} \\ & + \cos\alpha\sin\beta\,\delta\beta\Bigg)^2 \\ & + \Bigg(\frac{\sin\alpha\cos\beta\,\delta\alpha^2}{2} + \cos\alpha\sin\beta\,\delta\alpha\delta\beta - \cos\alpha\cos\beta\,\delta\alpha + \frac{\sin\alpha\cos\beta\,\delta\beta^2}{2} \\ & + \sin\alpha\sin\beta\,\delta\beta\Bigg)^2\end{aligned} \tag{3.50}$$

For the sake of greater insight into Equation (3.50), let us consider the sensitivity metric near the truth where the angle residuals are approximately the same order of magnitude. This allows us to eliminate one of the variables by assuming the two angle residuals are roughly equivalent, $\delta\alpha \approx \delta\beta = \mathcal{O}(\varepsilon)$. Substituting into Equation (3.50) and simplifying

$$\mathcal{O}\left[\left(\frac{\delta s}{\rho}\right)^2\right] = \frac{\varepsilon^2}{8}[12 + 4\cos(2\beta) - 4\varepsilon\sin(2\beta) + (9 - \cos(2\beta))\varepsilon^2] \tag{3.51}$$

$$\mathcal{O}(\delta s) = \rho\frac{\varepsilon}{2}\sqrt{6 + 2\cos(2\beta) - 2\varepsilon\sin(2\beta) + \frac{\varepsilon^2}{2}[9 - \cos(2\beta)]} \tag{3.52}$$

When the angle residuals are small ($\varepsilon \ll 1$), the sensitivity metric is reasonably approximated as

[29] As the residuals approach the angular resolution, the measurement sensitivity metric approaches the optical system's spatial resolution (the smallest detectable object).

$$\mathcal{O}(\delta s) \approx \rho \frac{\varepsilon}{2} \sqrt{6 + 2\cos(2\beta)} \tag{3.53}$$

Using the trigonometric identity, $\cos(2\beta) = 1 - 2\sin^2\beta$,

$$\mathcal{O}(\delta s) \approx \rho\varepsilon \sqrt{2 - \sin^2\beta} \tag{3.54}$$

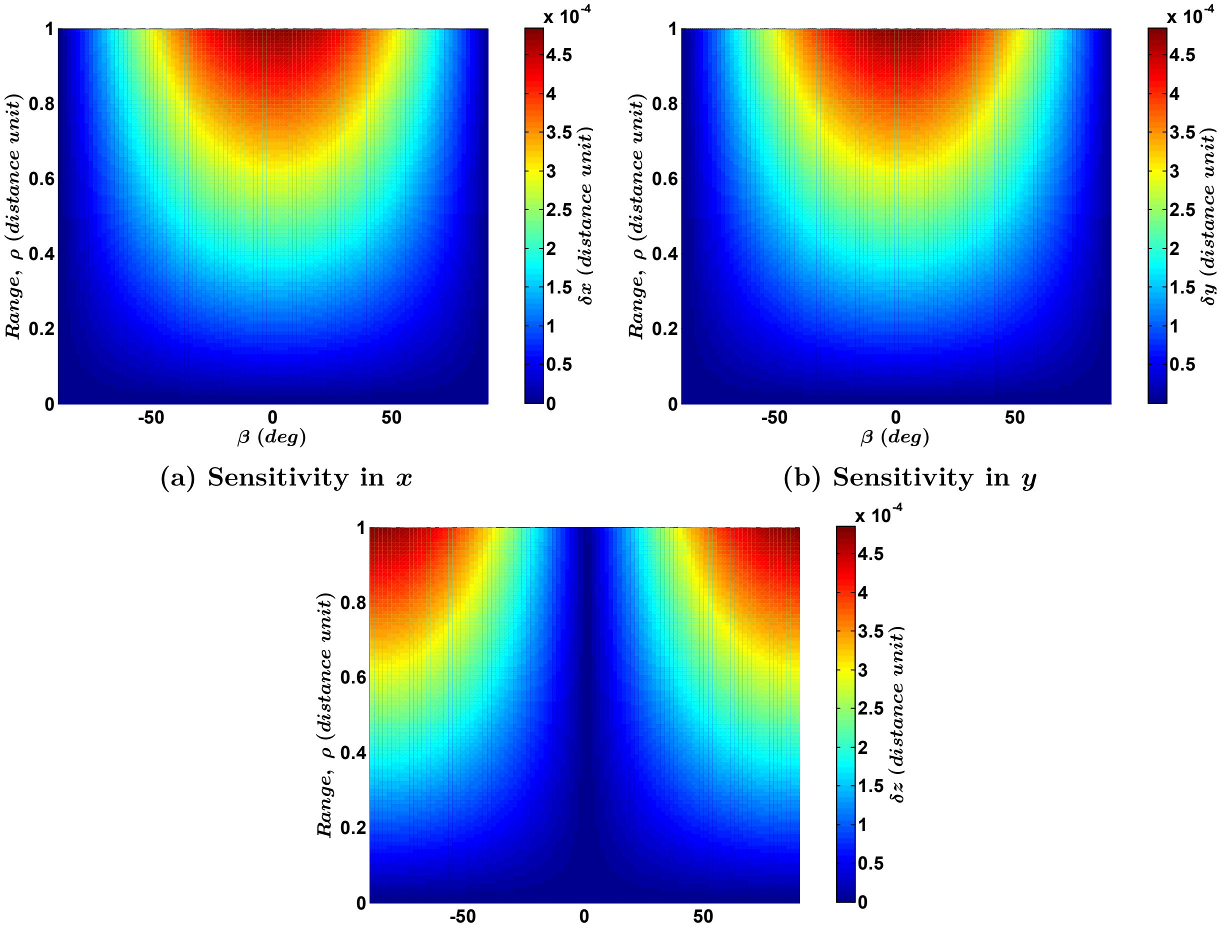
Notice that α has fallen out! This is an unexpected result, but the math is consistent with MATLAB's Symbolic Math Toolbox. Experimentation with the prior formulation of $\delta s\ (\alpha, \beta, \rho, \delta\alpha, \delta\beta)$ in Equation (3.50) indicates that δs is invariant to α. The invariance in δs seems to be due to α, β coupling in δx and δz components. For instance, when the δx and δz components are squared and combined, α is eliminated with a trigonometric identity: $y^2\cos^2(\alpha) + y^2\sin^2(\alpha) = y^2$. The y component is *only* a function of β: $y = \rho\sin\beta$.

Visualization of the sensitivity metric is helpful for intuition. Arbitrarily selecting $\varepsilon =$ 100 arc sec and setting $\alpha = 0$ since its cumulative effect on δs is nullified in δx and δz, Figure 11 displays the component sensitivities as β and range are varied. Range and sensitivity scale identically, so if the distance units for the range are selected as kilometers, the colorbar numbers are also given in kilometers.

As $|\beta|$ increases, $|\delta x|$ and $|\delta y|$ decrease while $|\delta z|$ increases. The cumulative effect of δx, δy and δz is a decrease in δs as $|\beta|$ increases. Equation (3.54) indicates δs is maximized as $\beta \to 0$. A mesh grid with variables in ρ, ε, and β convey the effect on δs in Figure 12 and Figure 13. Table 1 lists the maximum values of δs when $\beta = 0$ for several different ranges.

Table 1: Sensitivity metric at $\beta = 0$ and $\varepsilon = 100$ arc sec

Range	δs (m)
300 m	0.21
600 m	0.41
1 km	0.69
10 km	6.86
100 km	68.56



(a) Sensitivity in x

(b) Sensitivity in y

(c) Sensitivity in z

Figure 11: Component Sensitivity to Angle Residuals. The position component sensitivity to residuals are displayed as a function of range and β. The angular resolution is arbitrarily set as $\varepsilon = 100$ arc sec and $\alpha = 0$ to limit complexity.

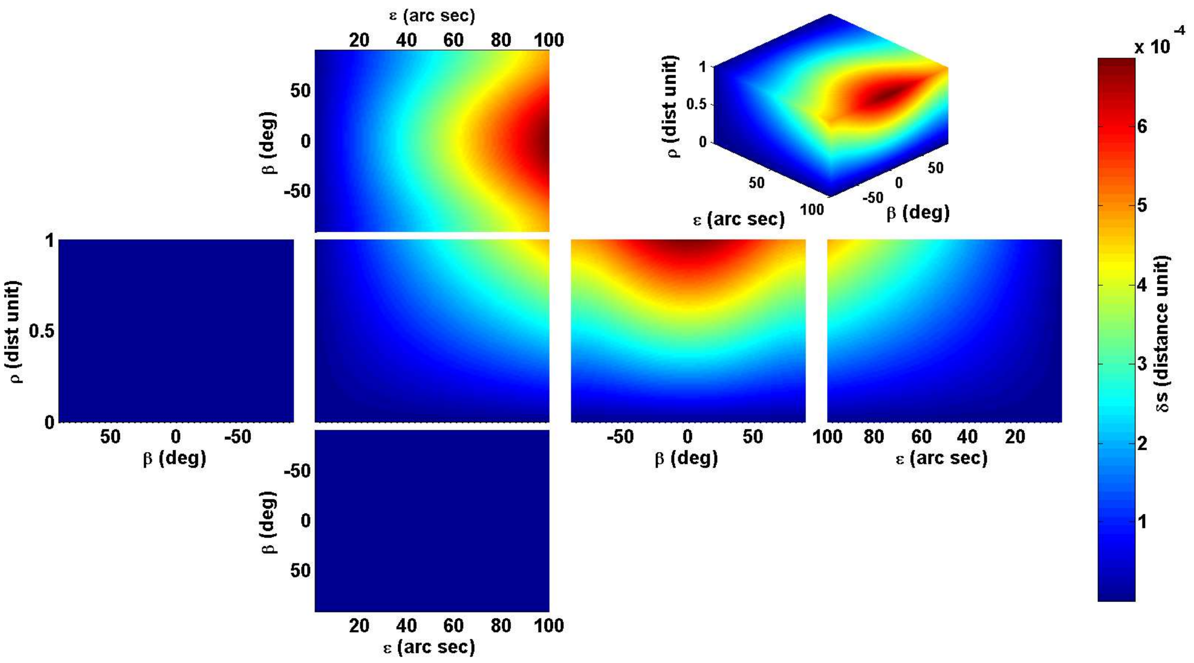


Figure 12: Sensitivity Metric in Wide Angle Field of View. The surfaces of 3D mesh grid are unfolded to reveal how the sensitivity metric changes in the variables ρ, ε, and β.

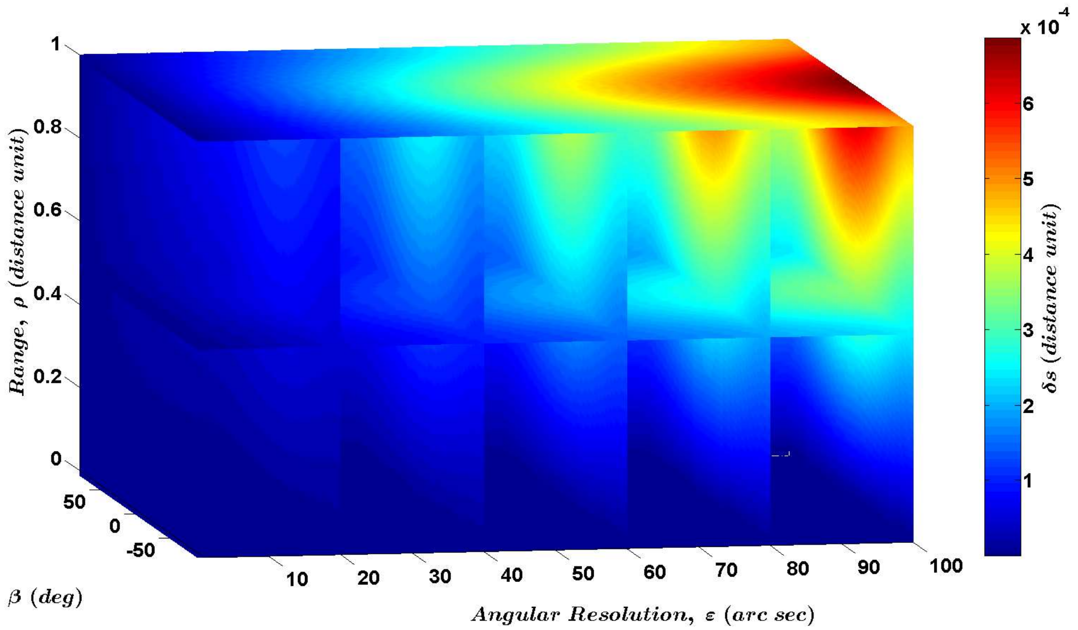


Figure 13: Slice Graph of Figure 12. Slices through the mesh grid depicted in Figure 12 to show the internal structure of the wide-angle field of view.

Overall, this quick study and the measurement sensitivity metric tells us that the cumulative effect of off-axis tracking is fairly inconsequential for narrow field-of-view systems as long as the measurement model is accurate and the residuals are small. Given the same angle residuals for optical axis tracking versus off-axis tracking, the measurement model maps slightly smaller spatial values for the latter case. It is worth noting that off-axis tracking does improve the sensitivity in δz which may marginally assist in enhancing down-range observability (along the boresight) while sacrificing slightly higher uncertainty in the x and y states.

3.6 Numerical Observability Analysis

Range observability can be numerically evaluated in a high fidelity simulation environment in one of three ways. The first method is to use the trend of the range error since the truth is known. The second method is to use the covariance of the estimated state. These two methods are quantitative and provide a direct assessment of a particular method's ability to observe range. A qualitative alternative is to calculate the condition number of the associated observability matrix. A weakly observable system exhibits large condition numbers. Yim et al. studied observability for LOS navigation and found that condition numbers greater than $\mathcal{O}(10^{16})$ are typically unobservable [150]. Others such as Gaias [145] and Sullivan [152] have used this method, so it is useful for comparing structural observability between different measurement and dynamics models. The general approach for calculating the condition number is described here.

First, specify a measurement model that maps the dynamic model state representation to the angle measurements. One example is Equations (3.21) and (3.22) in Hill's frame

$$\boldsymbol{M}_{\text{ref}}\big(\boldsymbol{X}(t)\big) = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\dfrac{y}{x}\right) \\ \sin^{-1}\left(\dfrac{z}{\sqrt{x^2+y^2}}\right) \end{bmatrix} \tag{3.55}$$

Given an initial state vector, $\boldsymbol{X}(t_0)$, the nominal reference trajectory can be propagated with a dynamics model. In most cases, this involves a state transition matrix, $\boldsymbol{\Phi}(t, t_0)$. Calculate the measurement sensitivity matrix, $\partial\boldsymbol{M}\big(\boldsymbol{X}(t)\big)/\partial\boldsymbol{X}$, along the nominal state trajectory on the interval t_1 to t_n. The partial derivatives will typically be evaluated analytically, but may involve an expansion with several partial derivatives (potentially numerical) or rotation matrices as in [145].

$$\left.\frac{\partial\boldsymbol{M}}{\partial\boldsymbol{X}}\right|_{\boldsymbol{X}(t)}^{\boldsymbol{X}(t_0)} = \begin{bmatrix} \dfrac{\partial\boldsymbol{M}}{\partial\boldsymbol{X}}\big(\boldsymbol{X}(t_0)\big) \\ \dfrac{\partial\boldsymbol{M}}{\partial\boldsymbol{X}}\big(\boldsymbol{\Phi}(t_1, t_0)\boldsymbol{X}(t_0)\big) \\ \vdots \\ \dfrac{\partial\boldsymbol{M}}{\partial\boldsymbol{X}}\big(\boldsymbol{\Phi}(t_n, t_0)\boldsymbol{X}(t_0)\big) \end{bmatrix} \tag{3.56}$$

The observability matrix, $\boldsymbol{T}$, is formed by multiplying the sensitivity matrix and the state transition matrix at every time step

$$\boldsymbol{T}|_{\boldsymbol{X}(t)}^{\boldsymbol{X}(t_0)} = \begin{bmatrix} \dfrac{\partial\boldsymbol{M}}{\partial\boldsymbol{X}}\big(\boldsymbol{X}(t_0)\big) \\ \dfrac{\partial\boldsymbol{M}}{\partial\boldsymbol{X}}\big(\boldsymbol{\Phi}(t_1, t_0)\boldsymbol{X}(t_0)\big)\boldsymbol{\Phi}(t_1, t_0) \\ \vdots \\ \dfrac{\partial\boldsymbol{M}}{\partial\boldsymbol{X}}\big(\boldsymbol{\Phi}(t_n, t_0)\boldsymbol{X}(t_0)\big)\boldsymbol{\Phi}(t_n, t_0) \end{bmatrix}$$

Full rank of $\boldsymbol{T}$ is a sufficient condition for local observability, but the qualitative notion of levels of observability is possible by taking the condition number of the observability Gramian, $\boldsymbol{T}^{\boldsymbol{T}}\boldsymbol{T}$.

Chapter 4

Estimating Range from Nonlinear Dynamics

Nonlinearities in the physical representation of satellite motion are essential for range observability. An understandably strong temptation to simplify orbital dynamics must be overcome to passively estimate range from LOS navigation without thrusting or augmentation from prior information. We begin this chapter by exploring the magnitude of simplifications made by Clohessy and Wiltshire [165] to describe relative motion in a two-body system. By understanding the order of what is lost, we can better appreciate what we have to gain from pristine nonlinear dynamics.

While most LOS research efforts have been spoiled by simplified relative motion models or linearization, we will show analytically and numerically how proximal spacecraft in NMC relative orbits can be uniquely identified by minor nonlinearities in the relative trajectory. In Chapter 5 we will take this proof a step farther by developing an IOD algorithm that exploits non-Keplerian, nonlinear motion and a nonlinear least squares estimator that dramatically improves the IOD solution.

4.1 Clohessy-Wiltshire Dynamics: Tracing the Range Observability Breakdown

Following the well-established path of Clohessy and Wiltshire [165], we transform the inertial motion of a Target satellite into a linear closed-form solution for two-body relative motion in the LVLH frame of a nearly circular reference orbit. Along the way, we pause to consider the approximations that deviate from the nonlinear motion. The magnitude of the error induced from linearization is assessed to determine the practicality of using small nonlinear effects for line-of-sight navigation.

Begin with the two-body vector equation of motion with a forcing term $\boldsymbol{f}(t)$ to represent additional accelerations or perturbations acting on the Target

$$\ddot{\mathbf{r}}(t) = -\frac{\mu}{r^3(t)}\mathbf{r}(t) + \boldsymbol{f}(t) \tag{4.1}$$

In the inertial frame, we can specify the Target's position relative to the Surveyor

$${}^i\mathbf{r}_t = {}^i\mathbf{r}_s + {}^i\mathbf{r}_{t/s} \tag{4.2}$$

$${}^i\mathbf{r}_t = {}^i\mathbf{r}_s + \rho\,{}^i\hat{\mathbf{r}}_{t/s} \tag{4.3}$$

where ${}^i\mathbf{r}_t$ is the Target position vector, ${}^i\mathbf{r}_s$ is the Surveyor position vector, ρ is the range between the two satellites and ${}^i\hat{\mathbf{r}}_{t/s}$ is the line of sight vector to the Target from the Surveyor.

Substitute Equation (4.2) into Equation (4.1) for the Target motion and drop the time dependence for succinctness

$${}^i\ddot{\mathbf{r}}_s + {}^i\ddot{\mathbf{r}}_{t/s} = -\frac{\mu}{\left\|{}^i\mathbf{r}_s + {}^i\mathbf{r}_{t/s}\right\|^3}\left[{}^i\mathbf{r}_s + {}^i\mathbf{r}_{t/s}\right] + \boldsymbol{f} \tag{4.4}$$

The approximations that are necessary for the linearized Clohessy-Wiltshire equations occur in the power expansion of Equation (4.4)

$$\begin{aligned}
\left\|{}^i\mathbf{r}_s + {}^i\mathbf{r}_{t/s}\right\|^{-3} &= \left[\left({}^i\mathbf{r}_s + {}^i\mathbf{r}_{t/s}\right)\cdot\left({}^i\mathbf{r}_s + {}^i\mathbf{r}_{t/s}\right)\right]^{-\frac{3}{2}} \\
&= \left[\left({}^i\mathbf{r}_s \cdot {}^i\mathbf{r}_s\right) + 2\left({}^i\mathbf{r}_s \cdot {}^i\mathbf{r}_{t/s}\right) + \left({}^i\mathbf{r}_{t/s} \cdot {}^i\mathbf{r}_{t/s}\right)\right]^{-3/2} \\
\left\|{}^i\mathbf{r}_s + {}^i\mathbf{r}_{t/s}\right\|^{-3} &= \left(1/{}^i\mathrm{r}_s^3\right)\left[1 + \frac{2\left({}^i\mathbf{r}_s \cdot {}^i\mathbf{r}_{t/s}\right)}{{}^i\mathrm{r}_s^2} + \frac{\left({}^i\mathbf{r}_{t/s} \cdot {}^i\mathbf{r}_{t/s}\right)}{{}^i\mathrm{r}_s^2}\right]^{-3/2}
\end{aligned} \tag{4.5}$$

The algebraic expansion of powers is performed with Newton's generalized binomial theorem [166]

$$\begin{aligned}(1+x)^{\alpha} &= \sum_{k=0}^{\infty}\binom{\alpha}{k}x^{k} \qquad\qquad (-1<x<1)\\ &= 1+\alpha x+\frac{\alpha(\alpha-1)}{2!}x^{2}+\frac{\alpha(\alpha-1)(\alpha-2)}{3!}x^{3}+\cdots\end{aligned} \tag{4.6}$$

where

$$\binom{\alpha}{k}=\frac{\alpha(\alpha-1)\cdots(\alpha-k+1)}{k!} \tag{4.7}$$

Returning to Equation (4.5)

$$\|{}^{i}\mathbf{r}_{s}+{}^{i}\mathbf{r}_{t/s}\|^{-3}=\left(\frac{1}{{}^{i}\mathrm{r}_{s}^{3}}\right)\left[1-\frac{3}{2}\left(\frac{2({}^{i}\mathbf{r}_{s}\cdot{}^{i}\mathbf{r}_{t/s})}{{}^{i}\mathrm{r}_{s}^{2}}+\frac{({}^{i}\mathbf{r}_{t/s}\cdot{}^{i}\mathbf{r}_{t/s})}{{}^{i}\mathrm{r}_{s}^{2}}\right)+\cdots\right]$$

$${}^{i}\mathrm{r}_{t}^{-3}\cong{}^{i}\mathrm{r}_{s}^{-3}\left[1-3\frac{({}^{i}\mathbf{r}_{s}\cdot{}^{i}\mathbf{r}_{t/s})}{{}^{i}\mathrm{r}_{s}^{2}}\right] \tag{4.8}$$

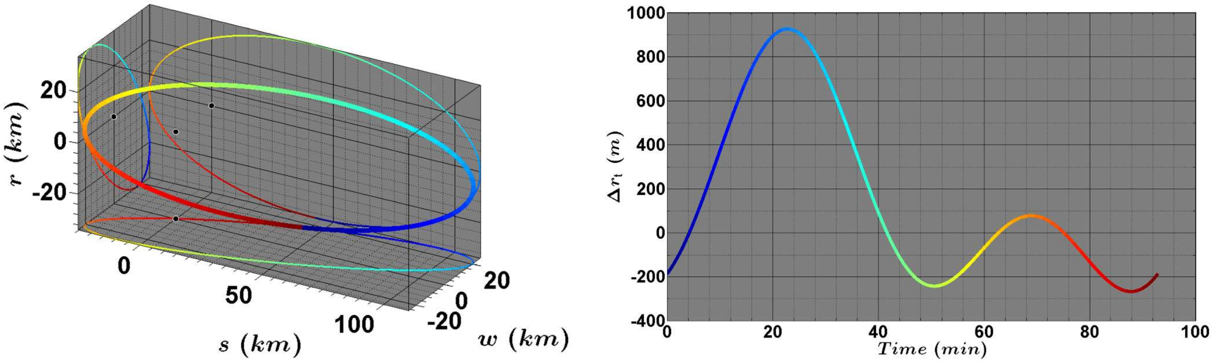
Only a subset of the first two terms of the binomial expansion are preserved in the Clohessy-Wiltshire equations as shown in Equation (4.8). This is where LOS navigation becomes impossible with linearized orbital dynamics and entire families of ambiguous relative trajectories are born. The approximation of ${}^{i}\mathrm{r}_{t}$ discards a small portion of the Target's radial distance in the inertial frame which is roughly equivalent to

$$\begin{aligned}\Delta{}^{i}\mathrm{r}_{t}\cong{}^{i}\mathrm{r}_{s}&\left[1-\frac{3}{2}\left(\frac{2({}^{i}\mathbf{r}_{s}\cdot{}^{i}\mathbf{r}_{t/s})}{{}^{i}\mathrm{r}_{s}^{2}}+\frac{({}^{i}\mathbf{r}_{t/s}\cdot{}^{i}\mathbf{r}_{t/s})}{{}^{i}\mathrm{r}_{s}^{2}}\right)+\frac{15}{8}\left(\frac{2({}^{i}\mathbf{r}_{s}\cdot{}^{i}\mathbf{r}_{t/s})}{{}^{i}\mathrm{r}_{s}^{2}}+\frac{({}^{i}\mathbf{r}_{t/s}\cdot{}^{i}\mathbf{r}_{t/s})}{{}^{i}\mathrm{r}_{s}^{2}}\right)^{2}\right.\\ &\left.-\frac{105}{48}\left(\frac{2({}^{i}\mathbf{r}_{s}\cdot{}^{i}\mathbf{r}_{t/s})}{{}^{i}\mathrm{r}_{s}^{2}}+\frac{({}^{i}\mathbf{r}_{t/s}\cdot{}^{i}\mathbf{r}_{t/s})}{{}^{i}\mathrm{r}_{s}^{2}}\right)^{3}\right]^{-\frac{1}{3}}-{}^{i}\mathrm{r}_{s}\left[1-3\frac{({}^{i}\mathbf{r}_{s}\cdot{}^{i}\mathbf{r}_{t/s})}{{}^{i}\mathrm{r}_{s}^{2}}\right]^{-\frac{1}{3}}\end{aligned}$$

$$
\begin{aligned}
&\cong {}^{i}\mathrm{r}_{s}\Big[1-\frac{3}{2}\,{}^{i}\mathrm{r}_{s}^{-2}\Big(2({}^{i}\mathbf{r}_{s}\cdot{}^{i}\mathbf{r}_{t/s})+({}^{i}\mathbf{r}_{t/s}\cdot{}^{i}\mathbf{r}_{t/s})\Big)\\
&\qquad\quad+\frac{15}{8}\,{}^{i}\mathrm{r}_{s}^{-4}\Big(2({}^{i}\mathbf{r}_{s}\cdot{}^{i}\mathbf{r}_{t/s})+({}^{i}\mathbf{r}_{t/s}\cdot{}^{i}\mathbf{r}_{t/s})\Big)^{2}\\
&\qquad\quad-\frac{105}{48}\,{}^{i}\mathrm{r}_{s}^{-6}\Big(2({}^{i}\mathbf{r}_{s}\cdot{}^{i}\mathbf{r}_{t/s})+({}^{i}\mathbf{r}_{t/s}\cdot{}^{i}\mathbf{r}_{t/s})\Big)^{3}\Big]^{-\frac{1}{3}}-{}^{i}\mathrm{r}_{s}[1-3\,{}^{i}\mathrm{r}_{s}^{-2}({}^{i}\mathbf{r}_{s}\cdot{}^{i}\mathbf{r}_{t/s})]^{-\frac{1}{3}}\\
&\cong\Big[{}^{i}\mathrm{r}_{s}^{-3}-\frac{3}{2}\,{}^{i}\mathrm{r}_{s}^{-5}\Big(2({}^{i}\mathbf{r}_{s}\cdot{}^{i}\mathbf{r}_{t/s})+({}^{i}\mathbf{r}_{t/s}\cdot{}^{i}\mathbf{r}_{t/s})\Big)\\
&\qquad\quad+\frac{15}{8}\,{}^{i}\mathrm{r}_{s}^{-7}(4({}^{i}\mathbf{r}_{s}\cdot{}^{i}\mathbf{r}_{t/s})^{2}+4({}^{i}\mathbf{r}_{s}\cdot{}^{i}\mathbf{r}_{t/s})({}^{i}\mathbf{r}_{t/s}\cdot{}^{i}\mathbf{r}_{t/s})+({}^{i}\mathbf{r}_{t/s}\cdot{}^{i}\mathbf{r}_{t/s})^{2})\\
&\qquad\quad-\frac{105}{48}\,{}^{i}\mathrm{r}_{s}^{-9}(8({}^{i}\mathbf{r}_{s}\cdot{}^{i}\mathbf{r}_{t/s})^{3}+12({}^{i}\mathbf{r}_{s}\cdot{}^{i}\mathbf{r}_{t/s})^{2}({}^{i}\mathbf{r}_{t/s}\cdot{}^{i}\mathbf{r}_{t/s})\\
&\qquad\quad+6({}^{i}\mathbf{r}_{s}\cdot{}^{i}\mathbf{r}_{t/s})({}^{i}\mathbf{r}_{t/s}\cdot{}^{i}\mathbf{r}_{t/s})^{2}+({}^{i}\mathbf{r}_{t/s}\cdot{}^{i}\mathbf{r}_{t/s})^{3})\Big]^{-\frac{1}{3}}\\
&\qquad\quad-\big[{}^{i}\mathrm{r}_{s}^{-3}-3\,{}^{i}\mathrm{r}_{s}^{-5}({}^{i}\mathbf{r}_{s}\cdot{}^{i}\mathbf{r}_{t/s})\big]^{-\frac{1}{3}}
\end{aligned}
$$

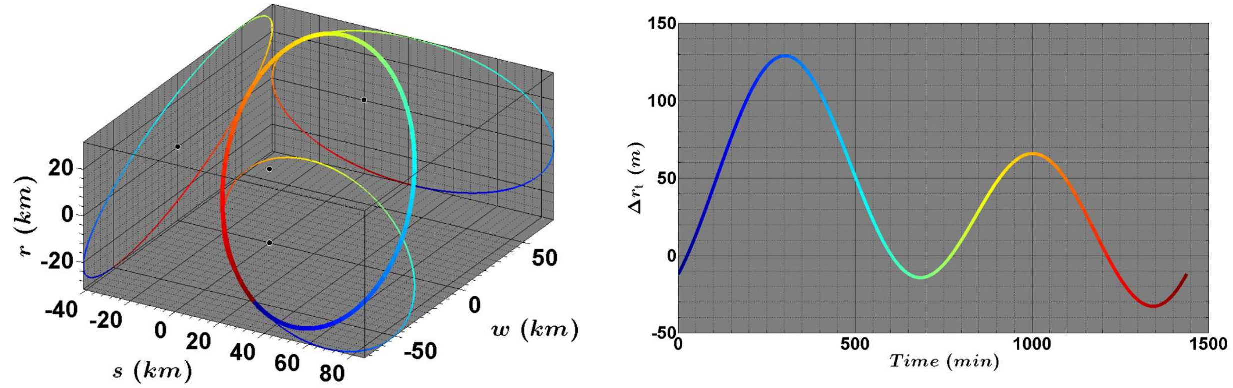
$$
\begin{aligned}
\Delta{}^{i}\mathrm{r}_{t}\cong\Big[&{}^{i}\mathrm{r}_{s}^{-3}-\frac{3}{2}\,{}^{i}\mathrm{r}_{s}^{-5}(2\,{}^{i}\mathrm{r}_{s}\rho({}^{i}\hat{\mathbf{r}}_{s}\cdot{}^{i}\hat{\mathbf{r}}_{t/s})+\rho^{2})\\
&+\frac{15}{8}\,{}^{i}\mathrm{r}_{s}^{-7}(4\,{}^{i}\mathrm{r}_{s}^{2}\rho^{2}({}^{i}\hat{\mathbf{r}}_{s}\cdot{}^{i}\hat{\mathbf{r}}_{t/s})^{2}+4\,{}^{i}\mathrm{r}_{s}\rho^{3}({}^{i}\hat{\mathbf{r}}_{s}\cdot{}^{i}\hat{\mathbf{r}}_{t/s})+\rho^{4})\\
&-\frac{105}{48}\,{}^{i}\mathrm{r}_{s}^{-9}(8\,{}^{i}\mathrm{r}_{s}^{3}\rho^{3}({}^{i}\hat{\mathbf{r}}_{s}\cdot{}^{i}\hat{\mathbf{r}}_{t/s})^{3}+12\,{}^{i}\mathrm{r}_{s}^{2}\rho^{4}({}^{i}\hat{\mathbf{r}}_{s}\cdot{}^{i}\hat{\mathbf{r}}_{t/s})^{2}\\
&+6\,{}^{i}\mathrm{r}_{s}\rho^{5}({}^{i}\hat{\mathbf{r}}_{s}\cdot{}^{i}\hat{\mathbf{r}}_{t/s})+\rho^{6})\Big]^{-\frac{1}{3}}-\big[{}^{i}\mathrm{r}_{s}^{-3}-3\,{}^{i}\mathrm{r}_{s}^{-4}\rho({}^{i}\hat{\mathbf{r}}_{s}\cdot{}^{i}\hat{\mathbf{r}}_{t/s})\big]^{-\frac{1}{3}}
\end{aligned}
\tag{4.9}
$$

The magnitude of the Target radius error, $\Delta{}^{i}\mathrm{r}_{t}$, is explored with sample NMC orbits in LEO (Case 04 referenced in Chapter 6) and GEO (Case 03 referenced in Chapter 6). The trajectories and radius errors are shown in Figure 14.



(a) Example LEO Target NMC trajectory at an altitude of ~400 km

(b) LEO Target radius error, $\Delta{}^{i}\mathrm{r}_{t}$, from linearization process



(c) Example GEO Target NMC trajectory at an altitude of ~35800 km

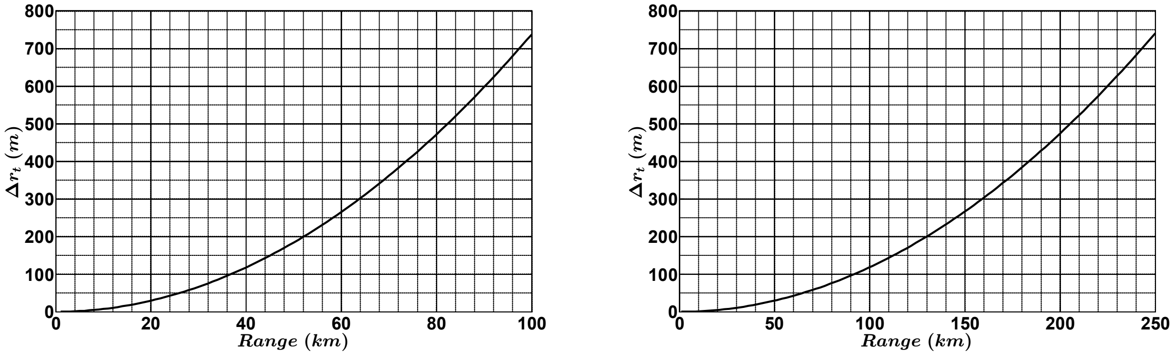
(d) GEO Target radius error, $\Delta {}^{i}\mathrm{r}_t$, from linearization process

Figure 14: NMC Trajectory Linearization Errors. The truncation of binomial expansion terms in the Clohessy-Wiltshire equations causes errors in the Target's inertial radius. Example NMC trajectories at LEO (a) and GEO (c) are propagated with nonlinear 2B dynamics and the radius error, $\boldsymbol{\Delta} {}^{i}\mathbf{r}_t$, is calculated for both as in (b) and (d), respectively. The error plots use a rainbow colored curve to visually synchronize time in the trajectory plots. The trajectory plots show the actual trajectory with the thicker line and orthogonal plane projections with the thinner lines for 3D perspective.

The errors show strong correlation with the range. For an NMC trajectory, the largest range occurs when ${}^{i}\hat{\mathbf{r}}_s \cdot {}^{i}\hat{\mathbf{r}}_{t/s} \approx 0$ which corresponds to $\alpha \approx \pm\frac{\pi}{2}$ in Hill's frame. These occurrences also happen to be near $\dot{\alpha}_{\min}$. If we substitute for ${}^{i}\hat{\mathbf{r}}_s \cdot {}^{i}\hat{\mathbf{r}}_{t/s} = 0$, the magnitude of the maximum radial error is approximately

$$\Delta {}^{i}\mathrm{r}_t \approx \left[{}^{i}\mathrm{r}_s^{-3} - \frac{3}{2}\, {}^{i}\mathrm{r}_s^{-5} \rho^2 + \frac{15}{8}\, {}^{i}\mathrm{r}_s^{-7} \rho^4 - \frac{105}{48}\, {}^{i}\mathrm{r}_s^{-9} \rho^6 \right]^{-\frac{1}{3}} - {}^{i}\mathrm{r}_s \tag{4.10}$$

The maximum radial error is plotted as a function of range for the LEO and GEO cases in Figure 15. Notice that the $\Delta {}^{i}\mathrm{r}_t$ is nearly equivalent at $\rho \approx 100$ km for the LEO case and $\rho \approx 250$ km for the GEO case. At $\rho = 100$ km in the GEO case, the $\Delta {}^{i}\mathrm{r}_t$ is a fraction of the LEO radial error. This is an indictment of the accuracy of the Clohessy-Wiltshire at lower altitudes where the curvature of the orbit is greater than higher-altitude orbits. The inversion of r_s has this effect in Equations (4.9) and (4.10)



(a) Maximum radial error at $\mathbf{r}_s = 6780$ km (LEO case)

(b) Maximum radial error at $\mathbf{r}_s = 42180$ km (GEO case)

Figure 15: Maximum Radial Errors in Clohessy-Wiltshire Equations. The radial errors from a binomial expansion truncation in the Clohessy-Wiltshire equations are maximized when ${}^{i}\hat{\mathrm{r}}_s \cdot {}^{i}\hat{\mathrm{r}}_{t/s} \approx 0$ for NMC trajectories. An approximation of the maximum radial errors are plotted for LEO (a) and GEO (c) as a function of range.

For an inspection orbit in LEO at 6780 km and a range of 10 km, $\Delta{}^{i}\mathrm{r}_t \approx 7.5$ m. This seems small, but the radial error translates to ~150 arc sec which is reasonably observed with an arc-second-resolution camera. If the process of developing the Clohessy-Wiltshire equations ended here, there might be hope that they could still be used with LOS navigation, but the loss of accuracy is a cascading phenomenon as we substitute Equation (4.8) and help ourselves to more simplifications.

In Equation (4.4), the inertial acceleration of the Target with respect to the Surveyor, ${}^{i}\ddot{\mathbf{r}}_{t/s}$, is recast as the Target acceleration in Hill's frame [167]

$${}^{i}\ddot{\mathbf{r}}_{t/s} = {}^{i}\ddot{\mathbf{r}}_{t/h} \tag{4.11}$$

$${}^{i}\ddot{\mathbf{r}}_{t/h} = {}^{h}\ddot{\mathbf{r}}_{t/h} + {}^{i}\ddot{\mathbf{r}}_{h/i} + 2\,{}^{h}\boldsymbol{\omega}_{h/i} \times {}^{h}\dot{\mathbf{r}}_{t/h} + {}^{h}\boldsymbol{\omega}_{h/i} \times \left({}^{h}\boldsymbol{\omega}_{h/i} \times {}^{h}\mathbf{r}_{t/h}\right) + {}^{h}\dot{\boldsymbol{\omega}}_{h/i} \times {}^{h}\mathbf{r}_{t/h} \tag{4.12}$$

In a circular Keplerian orbit, Hill's frame is not linearly accelerating, so ${}^{i}\ddot{\mathbf{r}}_{h/i} = \mathbf{0}$. This assumption introduces errors into the Clohessy-Wiltshire dynamics for any amount of eccentricity and non-Keplerian motion. Additionally, the coordinates in Hill's frame are selected such that ${}^{h}\boldsymbol{\omega}_{h/i} = n\,{}^{h}\hat{\boldsymbol{k}}$ and ${}^{h}\mathbf{r}_{h/i} = \mathrm{r}_{h}\,{}^{h}\hat{\boldsymbol{\imath}}$ where n is the mean motion of the Surveyor at the center of Hill's frame, $n = \sqrt{\mu/\mathrm{r}_h^3} = \sqrt{\mu/\mathrm{r}_s^3}$. Had we not assumed a constant angular rate in the orbital plane, we would have seen the effect of eccentricity creep into the dynamics. This assumption is akin to using a mean anomaly to describe the true anomaly. In a nearly circular orbit, the error is small, but as the differential eccentricity grows, so too does the error. We will resist the urge to quantify these errors here.

Since $\dot{n} = 0$, the angular acceleration of Hill's frame is ${}^{h}\dot{\boldsymbol{\omega}}_{h/i} = 0$. Equation (4.12) becomes

$$
{}^{i}\ddot{\mathbf{r}}_{t/s} = {}^{h}\ddot{\mathbf{r}}_{t/h} + 2\,{}^{h}\boldsymbol{\omega}_{h/i} \times {}^{h}\dot{\mathbf{r}}_{t/h} + {}^{h}\boldsymbol{\omega}_{h/i} \times \left({}^{h}\boldsymbol{\omega}_{h/i} \times {}^{h}\mathbf{r}_{t/h}\right) \tag{4.13}
$$

Substitute Equation (4.13) and Equation (4.8) into Equation (4.4)

$$
\begin{aligned}
{}^{i}\ddot{\mathbf{r}}_{s} + {}^{h}\ddot{\mathbf{r}}_{t/h} + 2\,{}^{h}\boldsymbol{\omega}_{h/i} \times {}^{h}\dot{\mathbf{r}}_{t/h} + {}^{h}\boldsymbol{\omega}_{h/i} \times \left({}^{h}\boldsymbol{\omega}_{h/i} \times {}^{h}\mathbf{r}_{t/h}\right) \\
= -\frac{\mu}{{}^{i}\mathrm{r}_s^3}\left[1 - 3\frac{\left({}^{i}\mathbf{r}_s \cdot {}^{i}\mathbf{r}_{t/s}\right)}{{}^{i}r_s^2}\right]\left[{}^{i}\mathbf{r}_s + {}^{i}\mathbf{r}_{t/s}\right] + \boldsymbol{f} \\
{}^{i}\ddot{\mathbf{r}}_{s} + {}^{h}\ddot{\mathbf{r}}_{t/h} + 2\,{}^{h}\boldsymbol{\omega}_{h/i} \times {}^{h}\dot{\mathbf{r}}_{t/h} + {}^{h}\boldsymbol{\omega}_{h/i} \times \left({}^{h}\boldsymbol{\omega}_{h/i} \times {}^{h}\mathbf{r}_{t/h}\right) \\
= -\frac{\mu}{{}^{i}\mathrm{r}_s^3}\left[{}^{i}\mathbf{r}_s + {}^{i}\mathbf{r}_{t/s} - 3\frac{\left({}^{i}\mathbf{r}_s \cdot {}^{i}\mathbf{r}_{t/s}\right)}{{}^{i}\mathrm{r}_s^2}\,{}^{i}\mathbf{r}_s - 3\frac{\left({}^{i}\mathbf{r}_s \cdot {}^{i}\mathbf{r}_{t/s}\right)}{{}^{i}\mathrm{r}_s^2}\,{}^{i}\mathbf{r}_{t/s}\right] + \boldsymbol{f}
\end{aligned} \tag{4.14}
$$

One final approximation is required for the derivation of the Clohessy-Wiltshire equations. The fourth term in the bracket of Equation (4.14) is ignored with the assumption that $\mathrm{r}_{t/s} \ll \mathrm{r}_s$. Assuming a circular orbit, we can substitute $n^2\mathrm{r}_s^3 = \mu$ and, recalling the coordinate definitions of Hill's frame, we can substitute ${}^{i}\mathbf{r}_s = \mathrm{r}_s\,{}^{h}\hat{\mathbf{r}}$. Also, notice that ${}^{i}\ddot{\mathbf{r}}_s = -\frac{\mu}{{}^{i}\mathrm{r}_s^3}\,{}^{i}\mathbf{r}_s$ and can be eliminated from both sides of Equation (4.14)

$$
\begin{aligned}
&{}^h\ddot{\mathbf{r}}_{t/h} + 2\,{}^h\boldsymbol{\omega}_{h/i} \times {}^h\dot{\mathbf{r}}_{t/h} + {}^h\boldsymbol{\omega}_{h/i} \times \left({}^h\boldsymbol{\omega}_{h/i} \times {}^h\mathbf{r}_{t/h}\right) \\
&\qquad = -\frac{n^2 \mathrm{r}_s^3}{\mathrm{r}_s^3}\left[{}^i\mathbf{r}_{t/s} - 3\frac{\left(\mathrm{r}_s\,{}^h\hat{\mathbf{r}} \cdot {}^i\mathbf{r}_{t/s}\right)}{\mathrm{r}_s^2}\mathrm{r}_s\,{}^h\hat{\mathbf{r}}\right] + \boldsymbol{f} \\
&{}^h\ddot{\mathbf{r}}_{t/h} + 2\,{}^h\boldsymbol{\omega}_{h/i} \times {}^h\dot{\mathbf{r}}_{t/h} + {}^h\boldsymbol{\omega}_{h/i} \times \left({}^h\boldsymbol{\omega}_{h/i} \times {}^h\mathbf{r}_{t/h}\right) \\
&\qquad = -n^2\left[{}^h\mathbf{r}_{t/s} - 3\left({}^h\hat{\mathbf{r}} \cdot {}^h\mathbf{r}_{t/s}\right){}^h\hat{\mathbf{r}}\right] + \boldsymbol{f}
\end{aligned}
$$

$$
{}^h\ddot{\mathbf{r}}_{t/h} + 2\,{}^h\boldsymbol{\omega}_{h/i} \times {}^h\dot{\mathbf{r}}_{t/h} + {}^h\boldsymbol{\omega}_{h/i} \times \left({}^h\boldsymbol{\omega}_{h/i} \times {}^h\mathbf{r}_{t/h}\right) = -n^2\left[{}^h\mathbf{r}_{t/s} - 3x_{t/s}\,{}^h\hat{\mathbf{r}}\right] + \boldsymbol{f} \tag{4.15}
$$

Finally, the resulting vector is shown as a scalar linear system in which the in-plane motion is decoupled from the cross-track motion. The frame notation has been dropped since the solution is obviously in Hill's frame

$$
\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} + \begin{bmatrix} -2n\dot{y} \\ 2n\dot{x} \\ 0 \end{bmatrix} + \begin{bmatrix} -n^2 x \\ -n^2 y \\ 0 \end{bmatrix} = \begin{bmatrix} -n^2 x + 3n^2 x \\ -n^2 y \\ -n^2 z \end{bmatrix} + \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \tag{4.16}
$$

$$
\begin{aligned}
\ddot{x} - 2n\dot{y} - 3n^2 x &= f_x \\
\ddot{y} + 2n\dot{x} &= f_y \\
\ddot{z} + n^2 z &= f_z
\end{aligned} \tag{4.17}
$$

Equation (4.17) represents the nonhomogeneous form of the Clohessy-Wiltshire equations. If the Target satellite is unperturbed and uncontrolled, the forcing term goes to zero, and the homogenous equations of motion are

$$
\begin{aligned}
\ddot{x} - 2n\dot{y} - 3n^2 x &= 0 \\
\ddot{y} + 2n\dot{x} &= 0 \\
\ddot{z} + n^2 z &= 0
\end{aligned} \tag{4.18}
$$

The linear differential equations can be written in state space form using the state vector $\boldsymbol{X} = [x \quad y \quad z \quad \dot{x} \quad \dot{y} \quad \dot{z}]^T$

$$
\dot{\boldsymbol{X}}(t) = \boldsymbol{A}\boldsymbol{X}(t) \tag{4.19}
$$

$$
\dot{\boldsymbol{X}}(t) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 0 & 2n & 0 \\ 0 & 0 & 0 & -2n & 0 & 0 \\ 0 & 0 & -n^2 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{X}(t) \tag{4.20}
$$

The solution of $\dot{\boldsymbol{X}}(t) = A\boldsymbol{X}(t)$ with $\boldsymbol{X}(t_0) = [x_0 \quad y_0 \quad z_0 \quad \dot{x}_0 \quad \dot{y}_0 \quad \dot{z}_0]^T$ is given by

$$\boldsymbol{X}(t) = \Phi(t, t_0)\boldsymbol{X}(t_0) \tag{4.21}$$

$$\Phi(t, t_0) = \frac{\partial \boldsymbol{X}(t)}{\partial \boldsymbol{X}(t_0)} = \exp\left[\int_{t_0}^{t} \boldsymbol{A}(\tau) d\tau\right] \tag{4.22}$$

For $t_0 = 0$, the state transition matrix, Φ, is

$$\Phi(t, 0) = \begin{bmatrix} 4 - 3\cos(nt) & 0 & 0 & \frac{1}{n}\sin(nt) & \frac{2}{n}[1 - \cos(nt)] & 0 \\ 6[\sin(nt) - nt] & 1 & 0 & \frac{2}{n}[\cos(nt) - 1] & \frac{4}{n}\sin(nt) - 3t & 0 \\ 0 & 0 & \cos(nt) & 0 & 0 & \frac{1}{n}\sin(nt) \\ 3n\sin(nt) & 0 & 0 & \cos(nt) & 2\sin(nt) & 0 \\ 6n[\cos(nt) - 1] & 0 & 0 & -2\sin(nt) & -3 + 4\cos(nt) & 0 \\ 0 & 0 & -n\sin(nt) & 0 & 0 & \cos(nt) \end{bmatrix} \tag{4.23}$$

In addition, the closed-form solution to Equation (4.21) is

$$\begin{gathered} x(t) = \left(4x_0 + \frac{2\dot{y}_0}{n}\right) - \left(3x_0 + \frac{2\dot{y}_0}{n}\right)\cos(nt) + \frac{\dot{x}_0}{n}\sin(nt) \\ y(t) = -(6nx_0 + 3\dot{y}_0)t + \left(y_0 - \frac{2\dot{x}_0}{n}\right) + \frac{2\dot{x}_0}{n}\cos(nt) + \left(6x_0 + \frac{4\dot{y}_0}{n}\right)\sin(nt) \\ z(t) = z_0\cos(nt) + \frac{\dot{z}_0}{n}\sin(nt) \\ \dot{x}(t) = \dot{x}_0\cos(nt) + (3nx_0 + 2\dot{y}_0)\sin(nt) \\ \dot{y}(t) = -6nx_0 - 3\dot{y}_0 + (6nx_0 + 4\dot{y}_0)\cos(nt) - 2\dot{x}_0\sin(nt) \\ \dot{z}(t) = \dot{z}_0\cos(nt) - nz_0\sin(nt) \end{gathered} \tag{4.24}$$

The Clohessy-Wiltshire solution is a classic that has been widely employed because of its simplicity. The most significant two-body dynamics are effectively modelled for nearly circular orbits on short intervals, but for any applications demanding accuracy, the Clohessy-Wiltshire solution is not trustworthy for more than a quarter of a period [126]. Further insight into these equations and analogous equations for elliptic reference orbits (e.g. Lawden and Tschauner-Hempel) can be accessed in [168].

4.1.1 2×1 Elliptical Motion

It is well known that the Clohessy-Wiltshire equations generate perfect 2×1 ellipses in the rs-plane under certain energy-matching initial conditions. The conditions are characteristic of the Surveyor and the Target having the same semi-major axis and differential eccentricity, δe. Since the Clohessy-Wiltshire equations assume a perfectly circular orbit for a non-accelerating Hill's frame, the Target must occupy a slightly eccentric orbit to manifest this motion. The idealized 2×1 ellipse is merely an artifice of linearized dynamics. The true nonlinear form diverges from the resemblance of an 2×1 ellipse as the differential eccentricity grows. This effect is exploitable for LOS navigation and masked entirely by the Clohessy-Wiltshire equations.

From the vantage point of the inertial frame with Keplerian dynamics, a small change in the Target's eccentricity leads to a change in the Target's radial vector and therefore the Target's distance in the relative orbit. Consider when the differential orbital elements are identical with the exception of δe such that the effect of the Target's eccentricity is dimensionally manifested as a variation in distance or $\delta\rho$

$$\begin{aligned} \frac{\partial \mathbf{r}_{t/s}}{\partial e_t} &= \frac{\partial}{\partial e_t}(\mathbf{r}_t - \mathbf{r}_s) \\ \delta\rho\ \hat{\mathbf{r}}_{t/s} &\approx \delta e_t \frac{\partial \mathbf{r}_t}{\partial e_t} \end{aligned} \tag{4.25}$$

As we allow the Target's eccentricity to grow, the Target range scales approximately linearly. This relationship is not useful with the Clohessy-Wiltshire dynamics since the LOS profile from 2×1 elliptical motion is unchanged as δe_t is scaled, but the relationship between δe_t and $\delta\rho$ can be exploited with nonlinear dynamics as we explore next.

4.2 Range Maps from Nonlinear Two-Body Relative Motion

In this section, we numerically demonstrate how range can be mapped from admissible $\rho|_{\dot{\alpha}}$ vs $\alpha|_{\dot{\alpha}}$ curves using the unabridged and unperturbed two-body motion. No maneuvers are required. The only requirements for the Surveyor are that it have knowledge of its own absolute navigation state and an arc-second-resolution camera for measuring angles to the Target. The IOD methodology in Chapter 5 leverages this insight for more complicated trajectories in the full geopotential.

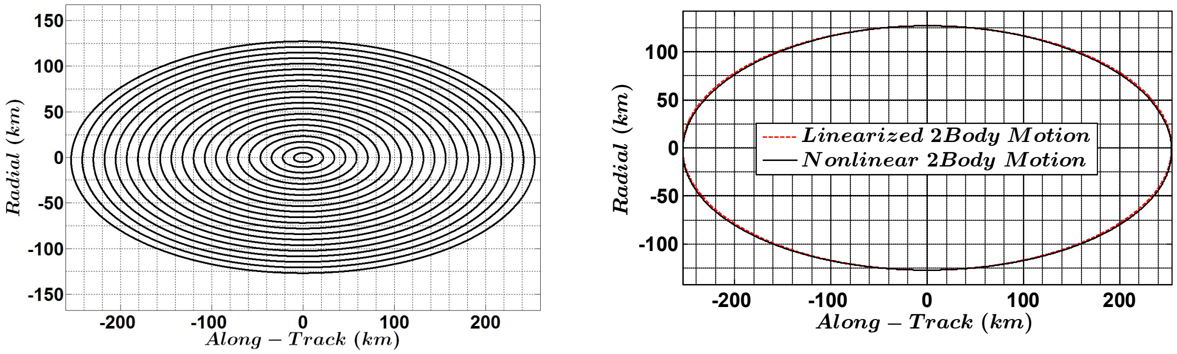
Start by generating a family of relative orbit hypotheses that span a parametrically constrained admissible region from which the true orbit is a candidate. Within the admissible region, all hypotheses exhibit similar, but not identical LOS profiles. The most basic trajectory for demonstrating this notion without our IOD method is the coplanar NMC trajectory centered in Hill's frame. We impose the co-planar condition by forcing $\delta i = 0$ and $\delta\Omega = 0$. The family of trajectories are made concentric and centered in Hill's frame (the Surveyor is at the origin) by imposing $\delta a = 0$, $\delta\Omega = 0$, $\delta\omega = 0$ and $\delta\nu = 0$. Eccentricity is the only element allowed to vary among hypotheses; all other classical orbital elements (COEs) are identical to those of the Surveyor. Table 2 indicates the COEs used for this case study, which happen to be from an old two-line elements (TLE) set for the ISS.

AGI's System Tool Kit (STK) is used to propagate the orbit hypotheses using two-body nonlinear dynamics. Figure 16 depicts 21 uniformly distributed hypotheses in the admissible region. At first glance, these appear to be the usual 2×1 ellipses in the rs-plane. Upon closer inspection, one begins to notice that the trajectory sags in the nadir direction as the relative motion approaches zero along-track velocity (radial speed is maximized) and converges toward the linearized Clohessy-Wiltshire solution at zero radial velocity (along track speed is maximized). The radial sag increases as the differential eccentricity, δe, increases.

Table 2: Orbital Elements for Centered, Coplanar NMC Family at LEO

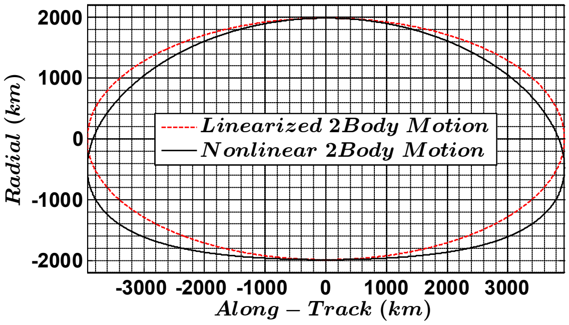
COEs	Target	Surveyor
a (km)	6777.329	6777.329
e (nd)	[0.002: 0.0009: 0.02]	0.001261
i (deg)	51.98716	51.98716
Ω (deg)	75.49392	75.49392
ω (deg)	102.6472	102.6472
ν (deg)	0	0

The ranges of interest for LOS navigation span ~100 km which are well within the family of orbits depicted in Figure 16(a). Clearly, the radial sag is not easy to observe at these ranges as illustrated in Figure 16(b) which contrasts the hypothesis with the largest δe to the Clohessy-Wiltshire solution. For the sake of intuition, a much larger eccentricity offset is depicted in Figure 16(c) by changing the Target's eccentricity to $e = 0.2$. The only other difference in COEs for Figure 16(c) is the semi-major axis, which is arbitrarily increased to 10,000 km to accommodate a higher eccentricity without dipping into the atmosphere.



(a) Set of admissible hypotheses for concentric, coplanar case

(b) CW solution vs. two-body solution for single hypothesis with $\delta e = 0.0187$ and $a = 6777.329$ km



(c) CW solution vs. two-body solution for $\delta e = 0.19$ and $a = 10000$ km

Figure 16: Nonlinear natural motion circumnavigation. A family of concentric coplanar relative trajectories is propagated with nonlinear two-body dynamics in (a). The trajectories appear to be 2x1 ellipses in the rs-plane just as the Clohessy-Wiltshire (CW) equations predict, but a comparison of the linear and nonlinear solution indicates subtle differences as in (b). As the eccentricity increases, the radial differences grow as in (c).

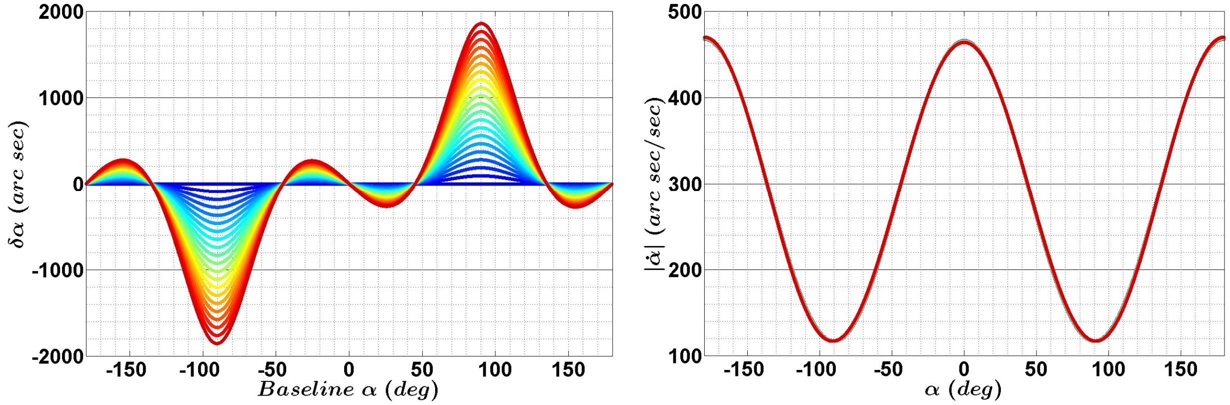
The nonlinear effects in the datasets lead to unique LOS profiles for each of the hypotheses. In this coplanar scenario, there is no cross-track motion, so $\beta = 0$ for all cases. The profile of α holds the key.

A baseline profile of α is selected for $e = 0.002$ since it is the lowest value of the Target's set of eccentricities from Table 2 (closest to a circular orbit). Each of the α profiles from the Target orbits of varying eccentricity is differenced with the baseline α profile at identical times throughout one orbit.[30] The δα results are depicted in Figure 17(a). Notice that the variation is largest twice per orbit near $\alpha = \pm 90°$. This corresponds to the farthest points in the along-track direction where the sagging occurs.

[30] In this case, hypotheses with eccentricities less than the Surveyor's eccentricity cannot be compared to the baseline since their angle profiles are 180° apart. If the Target's argument of perigee was rotated 180°, then the hypotheses would have eccentricities less than the Surveyor's eccentricity.

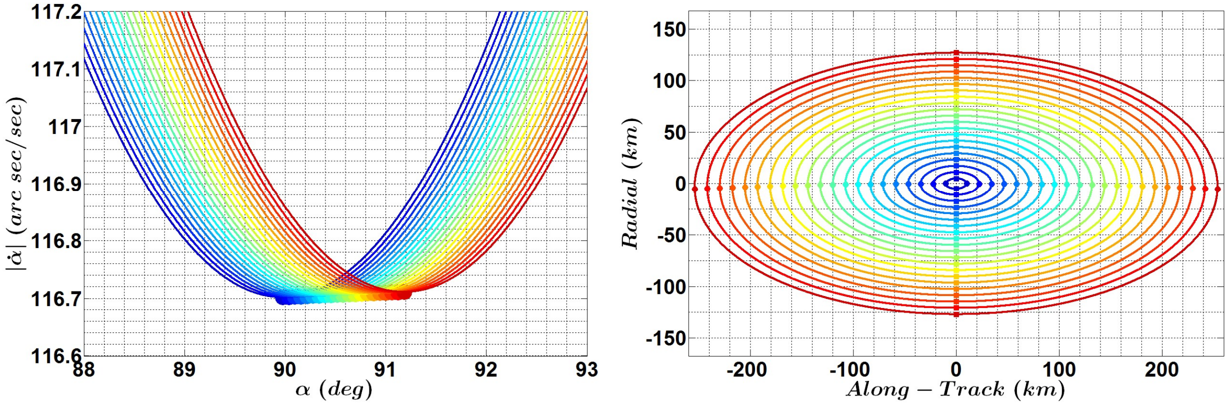
The angular rates shown in Figure 17(b) are quite similar for the full family of relative orbits since the curves appear to be overlapping. The absolute values of the angular rates are depicted, but the rates are actually negative. The rate will always be negative in Hill's frame due to the way we have defined $+\alpha$ (corresponds with the left-hand rule about the w-axis). This is independent of prograde or retrograde orbits; one gets the same convention in either case.

Upon closer inspection of the upper peaks in the angle-rate curves (where $\dot{\alpha}$ is minimized due to the negative sign), it is apparent that $\dot{\alpha}$ is roughly equivalent for the family of trajectories, *but the α location of the minimum angular rate differs* in the vicinity of ± 90 deg. Figure 17(c) shows this behavior clearly in a zoomed view at 90 deg. The variation in α is more than a degree for this small family of NMC orbits. When the location of the minimum angular rate is plotted atop the full trajectory in the rs-plane, a pattern emerges in Figure 17(d). As the eccentricity increases, the location of the minimum angular rate shifts in the negative radial direction. In other words, the sagging behavior correlates with the α location of $\dot{\alpha}_{\text{min}}$. The location of $\dot{\alpha}_{\text{max}}$ is virtually indistinguishable for all trajectories which is clear evidence that range observability varies throughout the orbit.



(a) $\delta\alpha$ from baseline α profile

(b) α rate of change



(c) Zoomed ziew of α rate of change

(d) Location of $\dot{\alpha}$ extrema.
Dots along horizontal are minima.
Squares along vertical are maxima.

Figure 17: Comparison of α and $\dot{\alpha}$ profiles. A baseline concentric coplanar NMC profile from $e = 0.011$ was compared to a family of NMC profiles to reveal where the largest differences occur in the orbit as in (a). The angular rates from each of the LOS profiles were also compared in (b) and (c). The minimum angular rates occur at slightly different locations in the relative orbit. These locations are depicted with dots in (d). The rainbow colors identify the eccentricity from lowest to highest: blue for $e = 0.002$ and red for $e = 0.02$.

Figure 18 shows a nearly linear relationship between the α location of the minimum angular rate ($\alpha|_{\dot{\alpha}_{\min}}$) and the range at those locations ($\rho|_{\dot{\alpha}_{\min}}$). A nearly linear relationship between range and eccentricity is also evident here, just as we predicted. Since the minimum angular rate for NMC trajectories is always at $\pm 90 + \Delta\alpha$ deg, our notation simply uses $\Delta\alpha$ in seconds of arc. A line is fit through $\Delta\alpha|_{\dot{\alpha}_{\min}}$ vs $\rho|_{\dot{\alpha}_{\min}}$ points to give range as a function of $\Delta\alpha$. We can also take advantage of the delineations of α at different $\dot{\alpha}$ values near $\dot{\alpha}_{\min}$ as shown in Figure 17(c). These fundamental observations are claimed as a major contribution since it enables a mapping from $\alpha|_{\dot{\alpha}} \rightarrow \rho$ within an admissible region and breaks the ambiguity problem. The technique for parameterizing the admissible region for more complicated trajectories is developed fully in Chapter 5.

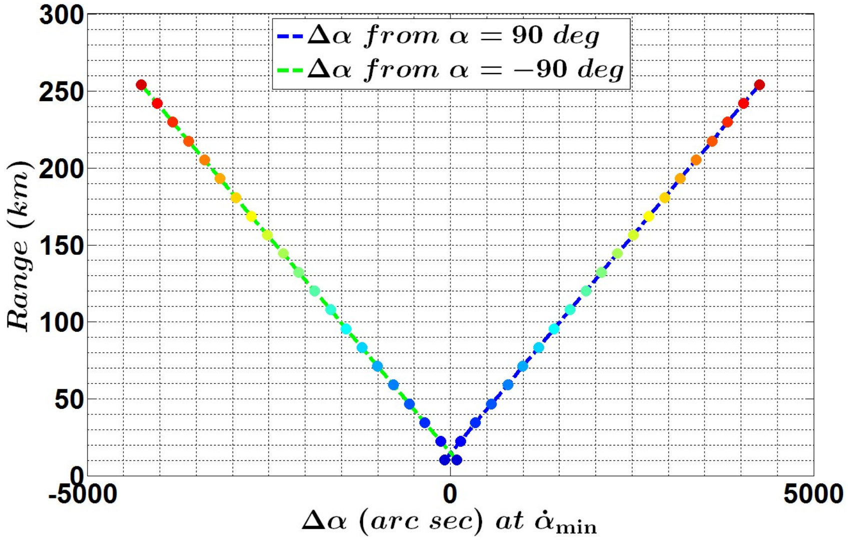


Figure 18: Relationship between $\dot{\alpha}_{\min}$ and range. The points of minimum angular rate, $\dot{\alpha}_{\min}$, along an NMC trajectory are near ±90 deg or more precisely $\pm 90 + \Delta\alpha$ deg. The value of $\Delta\alpha$ is almost linearly related to the range. The colored dots represent $\Delta\alpha$ for each of the hypotheses using the same color scheme as above. The two dashed lines are the curve fits used for mapping true observations at $\dot{\alpha}_{\min}$ for this particular case.

4.2.1 Range Resolution Metric

Range resolution describes how well range may be resolved at a particular orbital altitude or at different parts of an orbit when constrained by an angular resolution (system level). We introduce a range resolution metric to quantify the precision (not the accuracy) of our mapping techniques. Two orbit hypotheses within a particular admissible region are sufficient to calculate the range resolution metric. Use the relative position coordinates at the occurrence of a reference $\dot{\alpha}$ to calculate the range for each of the orbits

$$\rho|_{\dot{\alpha}} = \sqrt{x^2|_{\dot{\alpha}} + y^2|_{\dot{\alpha}} + z^2|_{\dot{\alpha}}} \tag{4.26}$$

The metric is simply the absolute value of the slope of $\rho|_{\dot{\alpha}}$ vs $\alpha|_{\dot{\alpha}}$ for the two hypotheses.

$$\mathrm{m}|_{\dot{\alpha}} = \left|\frac{\rho_2 - \rho_1}{\alpha_2 - \alpha_1}\right|_{\dot{\alpha}} \tag{4.27}$$

Since the $\rho|_{\dot{\alpha}}$ vs $\alpha|_{\dot{\alpha}}$ curve is approximately linear for a particular orbit altitude as in Figure 18, any random pair of hypotheses will suffice for determining the range resolution metric. The metric can be used on a global level to compare different orbit altitudes or on a local level to assess range resolvability at different parts of the orbit such as at $\dot{\alpha}_{\min}$ and $\dot{\alpha}_{\max}$. Note that the resolvability also says something about observability, which will be explored analytically and numerically in Section 4.3.4. As the denominator approaches zero, the resolvability metric approaches infinity, which physically means the range uncertainty is infinite.

Since the metric is a slope with standard units of km/arc secs, the actual range resolution is given by

$$\delta\rho|_{\dot{\alpha}} = \mathrm{m}|_{\dot{\alpha}}\delta\theta \tag{4.28}$$

where $\delta\theta$ is the system-level angle resolution. System-level is designated to account for the many sources of uncertainty from data collection to data reduction to admissible region construction. We think of angle resolution as the ability to observe a particular angle in time. For instance, even a perfect sensor and perfect process is limited by the sample rate. Sampling at 10 Hz at a 400 km altitude orbit results in an angle resolution of $\delta\theta \approx 11.7$ arc secs near $\dot{\alpha}_{\min}$. This would introduce system-level uncertainty since the discrete measurements could bypass $\dot{\alpha}_{\min}$.[31]

4.3 Analytical Proof of Range Observability

An analytical proof is developed next to provide a basic check of the numerical evidence for range maps ($\alpha|_{\dot{\alpha}} \to \rho$) introduced in Section 4.2. Numerical techniques are far superior

[31] We address this particular problem by fitting the α-profile with a sequence of cubic polynomials to reduce noise and sampling gaps (if the sensor has zero-mean random noise, then the uncertainty is substantially reduced with this process).

in terms of accuracy, but are understandably subject to skepticism without a simpler means to verify the results with first principles. The analytical solution presented here is limited to coplanar motion and uses assumptions and simplifications that make it less accurate than the pristine numerical methods, but the results are indisputable about range maps from nonlinear admissible hypotheses.

4.3.1 Assumptions and Geometry

The assumptions for the analytical solution are:

1. Two-body dynamics
2. Low eccentricity orbits
3. Admissible orbit constraints for simple coplanar, energy matching case ($\delta a = \delta i = \delta\Omega = \delta\omega = 0$)
4. Low differential true anomaly to ensure circumnavigation

Given these assumptions, we can realistically state $\Delta\nu$ and ${}^{h}y$ are maximized at $\dot{\alpha}_{\min}$. The geometry of the problem is depicted in Figure 19 with exaggerated scales and highly eccentric orbits since it amplifies the angle $\Delta\alpha|_{\dot{\alpha}_{\min}}$ and the radial distance ${}^{h}x$.

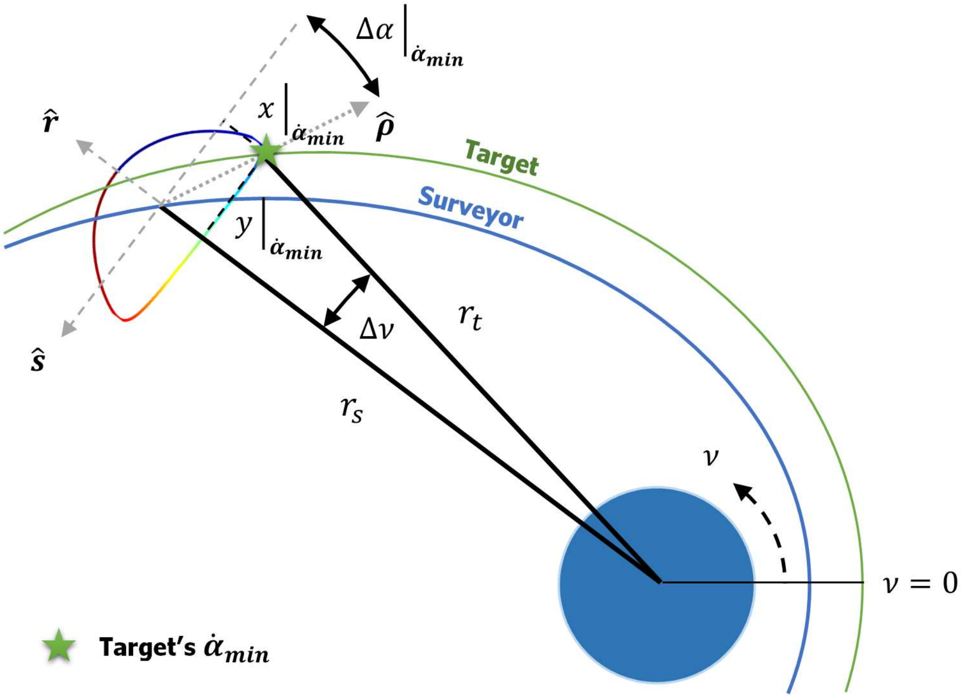

$\Delta\alpha\big|_{\dot{\alpha}_{min}}$
$x\big|_{\dot{\alpha}_{min}}$
$\hat{\boldsymbol{r}}$
$\hat{\boldsymbol{\rho}}$
Target
Surveyor
$y\big|_{\dot{\alpha}_{min}}$
r_t
$\Delta\nu$
$\hat{\boldsymbol{s}}$
r_s
ν
$\nu = 0$
Target's $\dot{\alpha}_{min}$

Figure 19: Analytical solution geometry. Both satellites' absolute orbits (blue and green curves) are shown with the same argument of perigee and different eccentricities in a coplanar case. The Target's relative motion in the LVLH frame is depicted with the rainbow curve.

The differential true anomaly is given by

$$\Delta\nu = \nu_t - \nu_s \tag{4.29}$$

where ν_t is the Target's true anomaly and ν_s is the Surveyor's true anomaly.

The Target's relative position in the radial (${}^{h}x$) and along-track (${}^{h}y$) directions are represented as a function of $\Delta\nu$ and the radial distance from the center of the Earth, r_t and r_s, for the Target and Surveyor, respectively.

$${}^{h}x = \mathrm{r}_t \cos\Delta\nu - \mathrm{r}_s \tag{4.30}$$

$${}^{h}y = \mathrm{r}_t \sin\Delta\nu \tag{4.31}$$

Finally, the angle offset, $\Delta\alpha$, from the the $\hat{\boldsymbol{s}}$-axis is

$$\Delta\alpha|_{\dot{\alpha}_{\min}} = \tan^{-1}\left(\frac{{}^{h}y}{{}^{h}x}\right) \tag{4.32}$$

4.3.2 Expansion of the Radius and True Anomaly

The first step is to describe the satellites' radial distances as a function of time in the Earth centered inertial frame. This is done by expanding the radius in terms of the mean anomaly to second order in eccentricity. Start with the expression for the radius as a function of the eccentric anomaly, E

$$\mathrm{r} = a(1 - e\cos E) \tag{4.33}$$

Substitute the first approximation for Kepler's equation

$$\mathrm{r} \approx a(1 - e\cos(M + e\sin M)) \tag{4.34}$$

The trigonometric sum identity given by Equation (4.35) can be substituted into Equation (4.34)

$$\cos(M + e\sin M) = \cos M\cos(e\sin M) - \sin M\sin(e\sin M) \tag{4.35}$$

$$\mathrm{r} \approx a(1 - e[\cos M\cos(e\sin M) - \sin M\sin(e\sin M)]) \tag{4.36}$$

Using small angle approximations where $\cos\theta \approx 1 - \theta^2/2$ and $\sin\theta \approx \theta$, the radius is expanded to $\mathcal{O}(e^3)$

$$\begin{aligned}
\mathrm{r} &\approx a\left(1 - e\cos M\left(1 - \frac{1}{2}e^2\sin^2 M\right) + e\sin M\,(e\sin M)\right)\\
\mathrm{r} &\approx a\left(1 - e\cos M + \frac{1}{2}e^3\cos M\sin^2 M + e^2\sin^2 M\right)\\
\mathrm{r} &\approx a\left(1 - e\cos M + \frac{1}{2}e^2 - \frac{1}{2}e^2\cos 2M + \mathcal{O}(e^3)\right)\\
\mathrm{r} &\approx a\left(1 - e\cos(M_0 + nt) + \frac{1}{2}e^2 - \frac{1}{2}e^2\cos(2M_0 + 2nt)\right)
\end{aligned} \tag{4.37}$$

From series expansions in elliptic motion we can also get a direct conversion between mean and true anomalies. Rather than deriving it here, we refer to [169]

$$\begin{aligned}\nu \approx M &+ \Big(2e - \frac{1}{4}e^3 + \frac{5}{96}e^5\Big)\sin M + \Big(\frac{5}{4}e^2 - \frac{11}{24}e^4\Big)\sin 2M \\ &+ \Big(\frac{13}{12}e^3 - \frac{43}{64}e^5\Big)\sin 3M + \frac{103}{96}e^4 \sin 4M + \frac{1097}{960}e^5 \sin 5M + \mathcal{O}(e^6)\end{aligned} \tag{4.38}$$

Simplifying to $\mathcal{O}(e^3)$ to match the accuracy of the radial expansion

$$\nu \approx M + 2e\sin M + \frac{5}{4}e^2 \sin 2M + \mathcal{O}(e^3) \tag{4.39}$$

where $M = M_0 + nt$ and $n = \sqrt{\mu/a^3}$

The differential true anomaly can now be written as a function of time. Substitute Equation (4.39) into Equation (4.29)

$$\begin{aligned}\Delta\nu(t) \approx M_t(0) - M_s(0) &+ 2e_t \sin(M_t(0) + nt) - 2e_s \sin(M_s(0) + nt) \\ &+ \frac{5}{4}e_t^2 \sin(2M_t(0) + 2nt) - \frac{5}{4}e_s^2 \sin(2M_s(0) + 2nt)\end{aligned} \tag{4.40}$$

4.3.3 Time and Radius at Minimum Alpha Rate

We want to maximize $\Delta\nu$ for finding $\Delta\alpha|_{\dot{\alpha}_{\min}}$, so take the partial derivative of $\Delta\nu$ to find the critical points at $\partial\Delta\nu/\partial t = 0$

$$\begin{aligned}\frac{\partial\Delta\nu}{\partial t} \approx &\frac{5}{2}ne_t^2 \cos(2M_t(0) + 2nt) + \ 2ne_t \cos(M_t(0) + nt) \\ &- \frac{5}{2}ne_s^2 \cos(2M_s(0) + 2nt) - 2ne_s \cos(M_s(0) + nt) = 0\end{aligned} \tag{4.41}$$

A closed form solution is not possible unless $M_s(0) = M_t(0) = M_0$, in which case

$$t|_{\dot{\alpha}_{\min}} \approx -\frac{1}{n}\left(M_0 - \operatorname{acos}\left(\frac{\left(\sqrt{50e_s^2 + 100e_s e_t + 50e_t^2 + 4} -\ 2\right)}{10(e_s + e_t)}\right)\right) \tag{4.42}$$

Our assumptions only restricted $\delta M(0)$ to be small to create the conditions for circumnavigation, so if $\delta M(0) \neq 0$, Equation (4.41) can be solved numerically.

Given $M_s(0)$, $M_t(0)$, n, e_s and e_t, calculate $t|_{\dot{\alpha}_{\min}}$. Plug in these quantities into Equation (4.37) to get the satellite radii at $\dot{\alpha}_{\min}$

$$\mathrm{r}_s|_{\dot{\alpha}_{\min}} \approx a\left(1 - e_s\cos(M_s(0) + nt_{\dot{\alpha}_{\min}}) + \frac{1}{2}e_s^2 - \frac{1}{2}e_s^2\cos(2M_s(0) + 2nt_{\dot{\alpha}_{\min}})\right) \tag{4.43}$$

$$\mathrm{r}_t|_{\dot{\alpha}_{\min}} \approx a\left(1 - e_t\cos(M_t(0) + nt_{\dot{\alpha}_{\min}}) + \frac{1}{2}e_t^2 - \frac{1}{2}e_t^2\cos(2M_t(0) + 2nt_{\dot{\alpha}_{\min}})\right) \tag{4.44}$$

The final step is to use the trigonometric relations from Section 4.3.1 to calculate the relative position coordinates and then $\Delta\alpha|_{\dot{\alpha}_{\min}}$. Multiple orbit hypotheses can be generated with the analytical method by repeating the process with different Target eccentricities.

4.3.4 Analytical Solution Case Studies

Let us revisit the NMC orbit scenario from Section 4.2, but this time we specify the Target's true orbit and reduce the number of hypotheses from 21 to 10 for less dense plots (see Table 3). The hypotheses are uniformly distributed at 10 km intervals along the radial axis as in Figure 20.

Table 3: Case 01A: Orbital Elements for Centered, Coplanar NMC at LEO

COEs	Target Truth	Target Hypotheses	Surveyor Truth
a (km)	6777.329	6777.329	6777.329
e (nd)	0.005000	*	0.001261
i (deg)	51.98716	51.98716	51.98716
Ω (deg)	75.49392	75.49392	75.49392
ω (deg)	102.6472	102.6472	102.6472
ν (deg)	0	0	0

* Approx.: 0.0160, 0.0145, 0.0131, 0.0116, 0.0101, 0.0086, 0.0072, 0.0057, 0.0042, 0.0027

The Surveyor and Target orbits are numerically propagated with two-body dynamics along with the ten hypotheses to find the "true" $\Delta\alpha|_{\dot{\alpha}_{\min}}$. The analytical calculation of

$\Delta\alpha|_{\dot{\alpha}_{\min}}$ is performed in parallel and plotted for comparison with the numerical results in Figure 21.

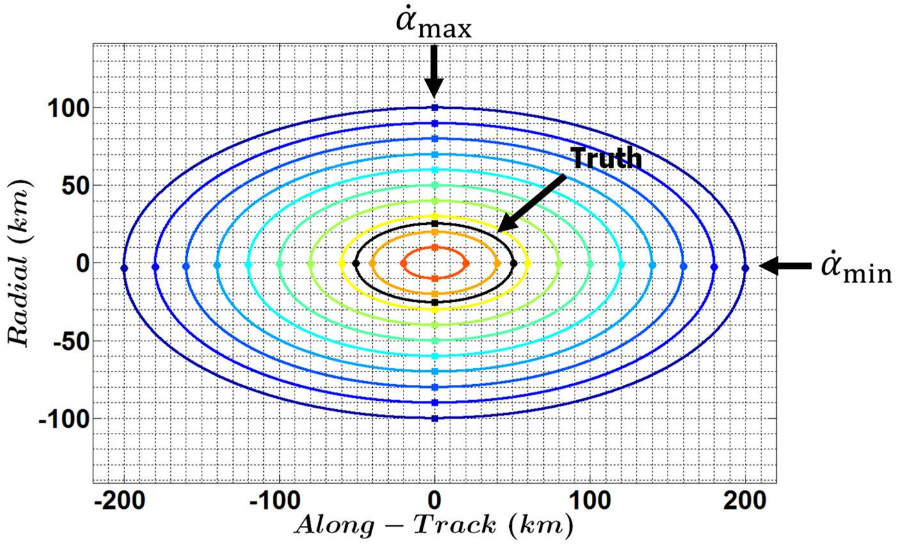


Figure 20: Numerically Propagated Truth and Hypotheses. The location of the minimum and maximum angle-rates are indicated along the s- and r-axis, respectively. The angle offset at $\dot{\alpha}_{\min}$ is visibly apparent, but not for $\dot{\alpha}_{\max}$.

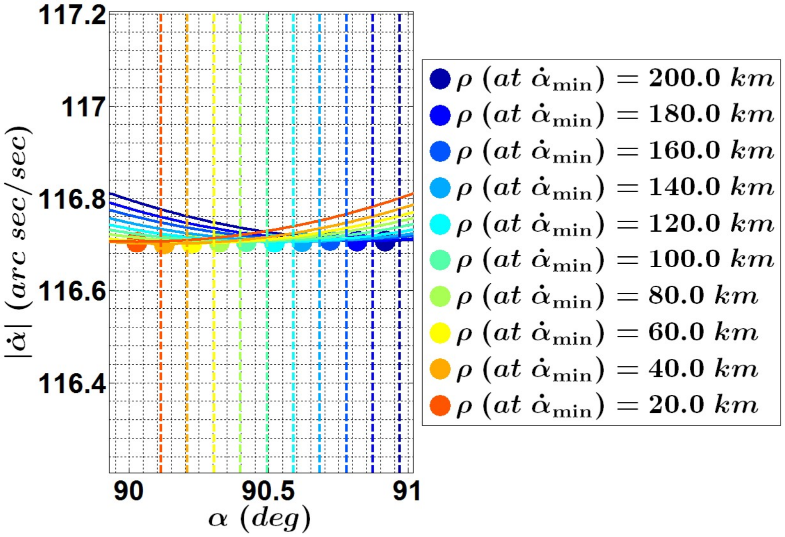


Figure 21: Comparison of Analytical and Numerical Calculations of $\Delta\alpha|_{\dot{\alpha}_{\min}}$. The dashed vertical lines represent the analytical values of $\Delta\alpha|_{\dot{\alpha}_{\min}}$ ($\dot{\alpha}_{\min}$ is not calculated analytically) and the dots represent the numerical values. The

colors are coordinated and the legend shows the associated range at the location of $\dot{\alpha}_{\text{min}}$.

The analytical solution has a positive bias (~0.1 deg), but the overall α spread agrees very well with the numerical solution. The cause of the analytical bias is due to our assumptions about the location of $\dot{\alpha}_{\text{min}}$ at $\Delta\nu_{\text{max}}$ and truncations in the radial and true anomaly expansions.

The range resolution metric at $\dot{\alpha}_{\text{min}}$ is also compared by using the smallest and largest range hypotheses (20 km and 200 km). The analytical metric is

$$\mathrm{m}_A|_{\dot{\alpha}_{\text{min}}} = \left|\frac{\rho_2 - \rho_1}{\Delta\alpha_2 - \Delta\alpha_1}\right| = \left|\frac{200.03 - 20.00}{3488.2 - 407.4}\right| = 0.0584\frac{\text{km}}{\text{arc secs}}$$

The numerical metric is nearly identical

$$\mathrm{m}_N|_{\dot{\alpha}_{\text{min}}} = \left|\frac{\rho_2 - \rho_1}{\Delta\alpha_2 - \Delta\alpha_1}\right| = \left|\frac{200.00 - 20.00}{3229.2 - 85.62}\right| = 0.0574\frac{\text{km}}{\text{arc secs}}$$

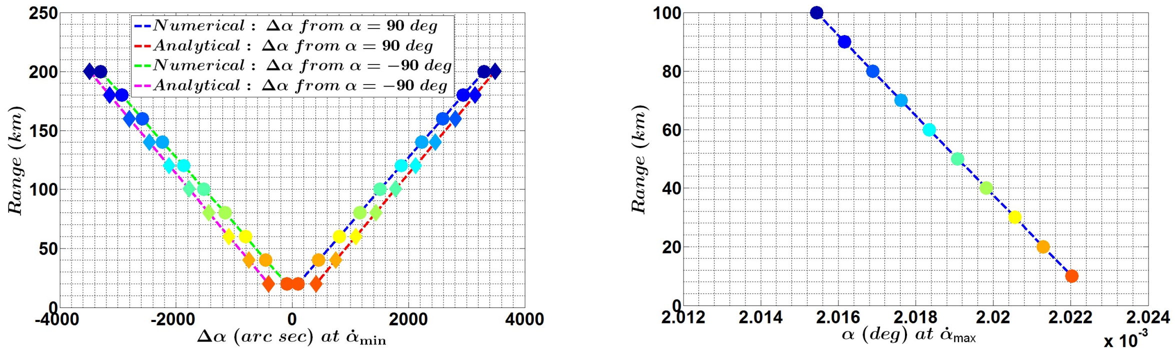
If range resolvability for a particular orbit is optimized at $\dot{\alpha}_{\text{min}}$, then the resolution metric should be smallest in that vicinity of the orbit. Likewise, if range resolvability is poorest at $\dot{\alpha}_{\text{max}}$, then the resolution metric should be largest in that vicinity of the orbit. If range is unobservable, then the resolvability approaches infinity with zero variation in the LOS profile

$$\mathrm{m} = \left|\frac{\rho_2 - \rho_1}{\Delta\alpha_2 - \Delta\alpha_1}\right| = \left|\frac{\Delta\rho}{0 - 0}\right| = \infty \tag{4.45}$$

Let us apply this thinking to the numerical data at $\dot{\alpha}_{\text{max}}$. The numerical metric is

$$\mathrm{m}_N|_{\dot{\alpha}_{\text{max}}} = \left|\frac{\rho_2 - \rho_1}{\Delta\alpha_2 - \Delta\alpha_1}\right| = \left|\frac{100.00 - 10.00}{7.25 - 7.28}\right| = 3000\frac{\text{km}}{\text{arc secs}}$$

Dimensionally, the metric is saying that in order to observe just 1 arc sec in variation within this particular admissible region, a minimum separation of 3000 km is required. Essentially, this is wholly unobservable. Range is completely unobservable when $\mathrm{m} = \infty$, but practical observability conditions could be established in the realm of m $\ll 1$. Therefore, by applying the metric to different parts of the orbit, we can assess the *quality* of range observability and *level* of resolvability using our range mapping method. The numerical range maps for $\dot{\alpha}_{\min}$ and $\dot{\alpha}_{\max}$ for the scenario above are shown in Figure 22. The metric is just the absolute value of the slope of $\rho|_{\dot{\alpha}}$ vs $\alpha|_{\dot{\alpha}}$ on these maps.



(a) Admissible region range maps at $\dot{\alpha}_{\min}$ (LEO case)

(b) Admissible region range maps at $\dot{\alpha}_{\max}$ near $\alpha = 0$ (LEO case)

Figure 22: Range maps. The range resolution metric is the slope of the curves fit to $\rho|_{\dot{\alpha}}$ vs $\Delta\alpha|_{\dot{\alpha}}$ within a particular admissible region. A smaller slope indicates better range observability and resolvability. The analytical maps are represented in (a) with red and magenta curves under diamonds. The numerical maps are represented in (a) and (b) with green and blue curves under dots.

The observed $\Delta\alpha|_{\dot{\alpha}_{\min}}$ from the truth orbit is -642.2 and 645.9 arc sec off the negative and positive *s*-axes, respectively. On the positive side, the analytical range map approximates that the Target has a range of 33.93 km while the numerical range map predicts 50.79 km (truth is 50.69 km). On the negative side, the analytical range map approximates 33.72 km and the numerical range map approximates 50.99 km (truth is 50.69 km). The bias in the analytical maps make the analytical solution unreliable under current assumptions, but the agreement in the range resolution metric suggests utility for comparing range observability/resolvability at different altitudes.

4.4 Effect of Earth's Oblateness

Until now, we have focused on the importance of nonlinearities for observing range, but the literature has shown that the Earth's oblateness (the J_2 term in the geopotential) also contributes to range observability. It is well known that the Earth's oblateness causes a secular drift in the mean anomaly, argument of perigee, and node. The drift rates are approximated by substituting the J_2 disturbing function into Lagrange's planetary equations [163]

$$\dot{M}_0 = -\frac{3\sqrt{\mu}J_2R_\oplus^2}{2a^{7/2}(1-e^2)^{3/2}}\left(\frac{3}{2}\sin^2(i)-1\right) \tag{4.46}$$

$$\dot{\omega} = -\frac{3\sqrt{\mu}J_2R_\oplus^2}{2a^{7/2}(1-e^2)^{2}}\left(\frac{5}{2}\sin^2(i)-2\right) \tag{4.47}$$

$$\dot{\mathit{\Omega}} = -\frac{3\sqrt{\mu}J_2R_\oplus^2}{2a^{7/2}(1-e^2)^{2}}\cos i \tag{4.48}$$

where $\dot{M}_0$ is the anomalistic frequency correction, $\dot{\omega}$ is the apsidal frequency and $\dot{\mathit{\Omega}}$ is the nodal frequency.

The mean anomaly, argument of perigee and right ascension of the ascending node (RAAN) grow with time as

$$M = M_0 + (n + \dot{M}_0)t \tag{4.49}$$

$$\omega = \omega_0 + \dot{\omega}t \tag{4.50}$$

$$\Omega = \Omega_0 + \dot{\mathit{\Omega}}t \tag{4.51}$$

Notice that the secular rates are a function of the non-secular elements: semi-major axis, eccentricity and inclination. If the Target and Surveyor have slightly different semi-major axes, eccentricities, and inclinations, the Lagrangian frequencies will vary between the two satellites and the effect should be observed in the LOS vector over multiple orbits.

Presently, it is unclear if the secular rates in the orbital plane ($\dot{M}_0$ and $\dot{\omega}$) can be distinguished from one another (especially for nearly circular orbits) since both are predominantly concealed in α and both have a similar effect in the along-track direction. Consider the case where the Target and Surveyor have the same semi-major axis and inclination but slightly different eccentricities as expected for NMC trajectories. Equations (4.46) - (4.48) show that the Target and Surveyor will experience different levels of drift from the J_2 disturbance. The satellite with the larger eccentricity will experience larger secular rates and the differential elements $\delta\omega$ and $\delta\nu$ grow in the orbital plane. Both $\delta\omega$ and $\delta\nu$ shift the NMC orbit in the along-track direction. Since $\dot{\omega}$ and $\dot{M}_0$ are so small, discriminating between their elemental effects over one orbit is an enormous challenge. The combined differential argument of latitude, $\delta u = \delta\omega + \delta\nu$, would be more easily observed over time. Unfortunately, range ambiguity is embedded in δu if one cannot discriminate its components. Therefore, while J_2 does theoretically help alter the two orbits over time just as micro-maneuvers do, it requires a dynamics model that can predict these effects and it requires care that the anomalistic and apsidal frequencies can be properly discriminated. The nonlinear effects are more promising.

Chapter 5

Methodology

Five major processes are involved in our LOS navigation research and each are described fully in this chapter. The first process encompasses orbit propagation of the Target and Surveyor's truth state vectors followed by data reduction to angle observations. Angle observations from certain parts of the orbit are recorded in the second process to establish a set of nondimensional parameters that constrain the Target's possible orbits to a so-called admissible region. A family of uniformly sampled orbit hypotheses within the admissible region are created from the nondimensional parameters. The hypotheses are propagated as a cluster and the inertial data for each hypothesis is reduced to angle and angle-rate data in the third major process. Virtual angle and eclipse observations from the hypotheses are analyzed and compared with the actual observations from the first process to map range during certain parts of the orbit. The range maps in the fourth major process enables an initial orbit determination (IOD). The IOD solution is converted into a KAM torus and passed into a nonlinear least squares filter to refine the Target's orbit estimate in the fifth and final process. A block diagram showing the processes and the data flow are presented in Figure 23

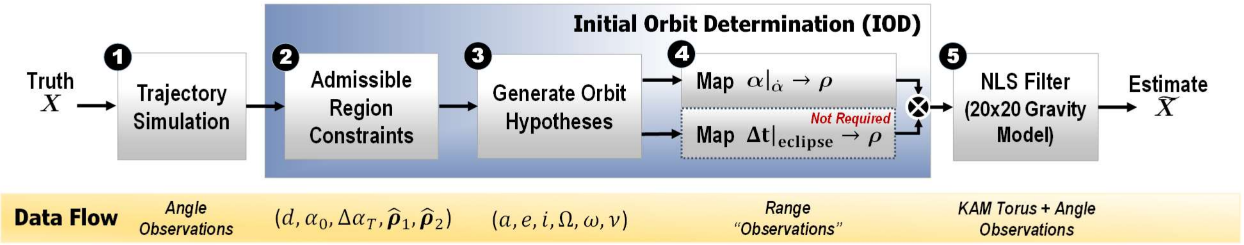

Initial Orbit Determination (IOD)
Truth
X
1
Trajectory Simulation
2
Admissible Region Constraints
3
Generate Orbit Hypotheses
4
Map $\alpha|_{\dot{\alpha}} \rightarrow \rho$
Not Required
Map $\Delta t|_{\text{eclipse}} \rightarrow \rho$
5
NLS Filter (20x20 Gravity Model)
Estimate
$\bar{X}$
Data Flow
Angle Observations
$(d, \alpha_0, \Delta\alpha_T, \hat{\boldsymbol{\rho}}_1, \hat{\boldsymbol{\rho}}_2)$
$(a, e, i, \Omega, \omega, \nu)$
Range "Observations"
KAM Torus + Angle Observations

Figure 23: Process Block Diagram & Data Flow. Seven major processes characterize our novel approach. Blocks 2-5 are novel contributions.

5.1 Trajectory Simulation

The code for this effort is a combination of MATLAB and C++. The experimental algorithms, named Proteus,[32] are written in MATLAB and orbital data is generated from a custom build of NASA's open source General Mission Analysis Tool (GMAT) 2011a written in C++. Orbits are propagated with an eighth order Runge-Kutta integrator and seventh order error control (Dormand-Prince coefficients) in GMAT. Version 2011a is the last release of GMAT with a MATLAB server for executing GMAT scripts from within the MATLAB environment. The server enables a fully automated and seamless solution across the five major processes. MATLAB R2010a (32-bit version) is used for compatibility with the GMAT interface. The choice of GMAT as the orbit propagator is strategic. Not only is GMAT faster than numerical integrations in MATLAB since the code is compiled ahead of time, but also has undergone significant verification and validation work by NASA and its partners. A comparable in-house tool would be suspect without considerable effort.

[32] Proteus was an elusive Greek god of the sea who had the gift of prophecy and shape shifting to avoid capture. The name is fitting since LOS navigation has eluded so many. Equally fitting, the name's origin comes from protos which literally means "the first." Proteus is believed to be the first solution to LOS navigation in NMC trajectories without prior data or thrusting.

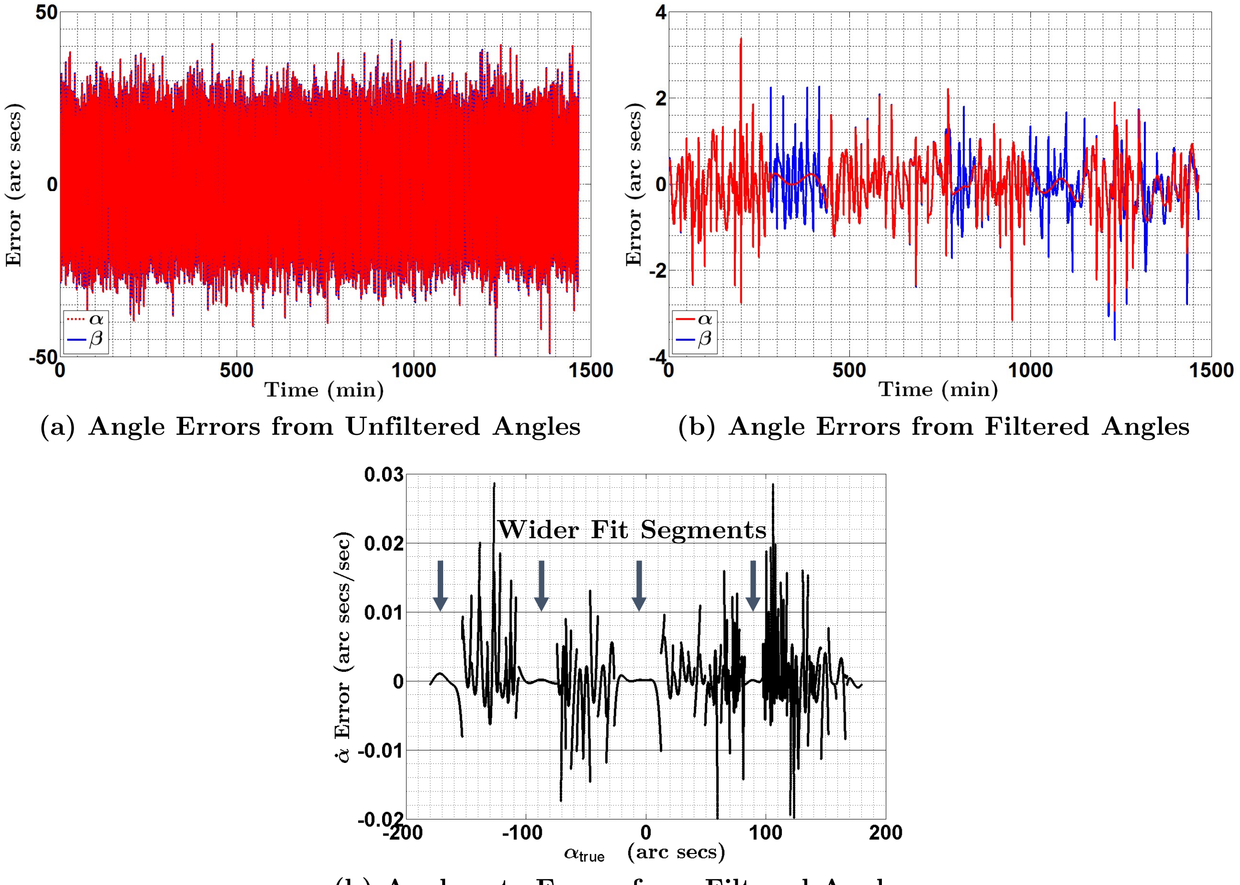
The only source of perturbations used for the propagations in this work is NASA's Earth Gravity Model 1996 (EGM96) [170] which is generally set to 20×20 (degree and order) when the "full geopotential" is simulated. The GMAT source code is modified by the author to neglect the default long-term effects of precession, nutation and polar motion during orbit propagations. Without these effects, a simple z-axis rotation converts between the Earth-center Earth-fixed (ECEF) and Earth-centered inertial (ECI) frames. This modification is required for constructing KAM tori to model the Earth's full geopotential in the NLS filter. Precession, nutation and polar motion have a negligible effect for single-period propagations, but over a long 1-year propagation—required for building KAM tori—these small perturbations slowly alter the orbit. The torus is only meant to model the deterministic geopotential forces in the ECEF frame. It is possible to add precession, nutation and polar motion (and even other perturbations) to the torus after construction, but it will not be significant over the orbital period timescale of interest and is ignored entirely.

The function `gmatOrbitProp` (Appendix E.3) automatically generates GMAT scripts from our input file and calls the GMAT server to execute the script. GMAT exports the simulation data to a .txt file from which `gmatOrbitProp` loads the data into MATLAB.

5.1.1 LOS Data Reduction

Once the trajectories are propagated and loaded into MATLAB, the function `losProc` (Appendix E.4) converts the absolute navigation data to Hill's frame and generates LOS products (angles, angle-rates, etc.) using the same methodologies described in Chapter 3.

Angle biases and standard deviations are added in `losProc` as specified in the input file. The noisy angle profile from one orbital period is broken into 1000 arcs that are small enough to fit cubic polynomials with high confidence. The resulting polyfit segments are pieced together as a crude filtration process in an attempt to limit the angle uncertainty to a bias, which is typically very small. Most cameras can calibrate for a bias which results in zero mean noise. Figure 24 shows the residuals of the initial noisy α data ($\sigma = 10$ arc sec) and the residuals of filtered α data from Case 03C described in Chapter 6. Near $\dot{\alpha}$ extrema or waypoints, the polynomial fit spans larger data segments to avoid endpoint spikes that could interfere with the numerical $\dot{\alpha}$ calculations.



(a) Angle Errors from Unfiltered Angles

(b) Angle Errors from Filtered Angles

(b) Angle-rate Errors from Filtered Angles

Figure 24: Filtered Angle & Angle-Rate Errors. The measured noisy data (a) is filtered with cubic polynomials in small segments to reduce the angle errors (b) and the angle-rate errors (c).

5.2 NMC Admissible Region Constraints

A set of dimensionless parameters defines the shape and orientation constraints from which a family of admissible orbit hypotheses exist. The family is characterized by similar, but not identical LOS profiles. These parameters are $(d, \alpha_0, \Delta\alpha_T, \hat{\boldsymbol{\rho}}_1, \hat{\boldsymbol{\rho}}_2)$ where d is the skewness factor of the projected trajectory in the rs-plane, α_0 is the angle at epoch in the rs-plane, $\Delta\alpha_T$ is the angle disparity in the rs-plane after one Kozai period (Surveyor's orbit), $\hat{\boldsymbol{\rho}}_1$ is a unit LOS vector pointing to the upper relative apsis and $\hat{\boldsymbol{\rho}}_2$ is a unit LOS vector pointing to the sw-plane crossing in either the positive or negative s-direction. Each of the parameters are indicated in Figure 25. Although the figure depicts $\Delta\alpha_T$ near the lower relative apsis, a disparity may be measured anywhere in the orbit as long as the epoch is consistently associated with α_0. The disparity will not be the same at different locations, hence the need for consistency during the hypothesis generation process.

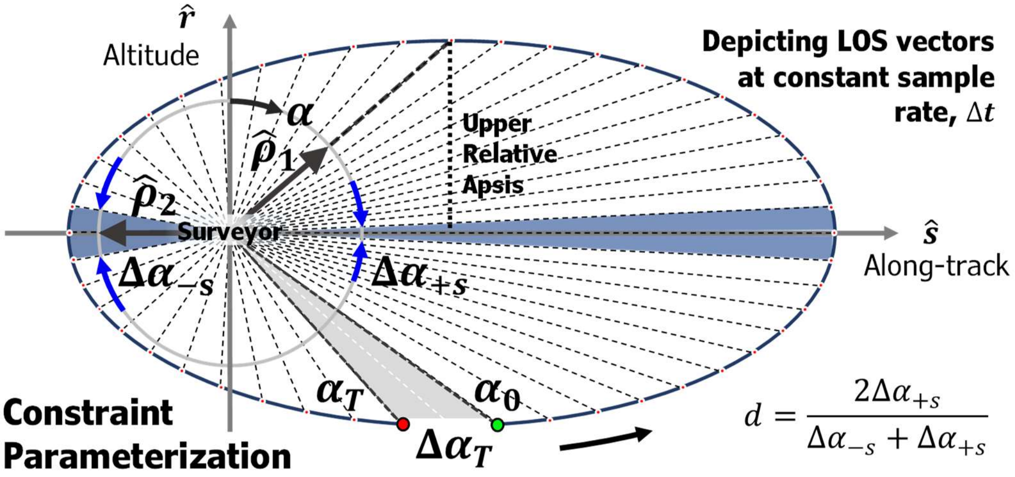


Figure 25: Admissible Region Parameterization. The shape and orientation of an NMC trajectory is constrained by the set of parameters $(d, \alpha_0, \Delta\alpha_T, \hat{\rho}_1, \hat{\rho}_2)$.

$\hat{\boldsymbol{\rho}}_1$ and $\hat{\boldsymbol{\rho}}_2$ are used to determine inclination and right ascension of the ascending node. Although other combinations of LOS vectors will suffice, it is preferable to have a quarter period separating them as will become apparent when determining the orbital angular momentum unit vector. $\hat{\boldsymbol{\rho}}_2$ is easy to find, but $\hat{\boldsymbol{\rho}}_1$ is identified after it has been measured by splitting the time interval between $\alpha = \pi/2$ and $\alpha = -\pi/2$. This is the location of the upper relative apsis. The lower relative apsis could also be used, but it is preferable to look out into space rather than at the Earth while taking measurements. This point was originally chosen while studying low differentials in argument of perigee. In those cases, the inertial position and velocity vector of the satellites are nearly perpendicular at the relative apsis, which improves eccentricity approximations. We do not constrain the differentials in the argument of perigee, but the practice of selecting this point has persisted.

The admissible region established by these parameters allows for one degree of freedom in the solution space for generating scaled hypotheses, but it is presently limited to NMC trajectories. In other words, the Target's projected trajectory in the rs-plane must include $\alpha = \pi/2$ and $\alpha = -\pi/2$ within one orbital period to ensure circumnavigation. Other RPO trajectories (co-elliptical, co-circular, r-bar station keeping, etc.) cannot meet this criterion.

Context of an Admissible Region

The concept of an admissible region in astrodynamics comes from efforts to detect and track near-Earth objects (NEOs). On a single night, a typical asteroid-hunting telescope will collect a small number of digital images in the same region of the celestial sphere. NEOs are detected by the line they produce after the images are digitally blinked. The processed astrometric observations consist of angle and angle-rate data, but if the angular arc is too short—known as a very short arc (VSA)—an orbit cannot be generated. These have also been called an ONS for "One Night Stand" because after the detections[33] are reported to the Minor Planet Center (MPC), some of them are never found again. This leads to issues over discovery rights if the object cannot be followed-up.

Milani et al. [171] devised a way to use the limited amount of data from a VSA to constrain the possible values of r and $\dot{r}$ (effectively the range and range rate) in what is called an admissible region. The admissible region represents the sampling space for a set of "virtual asteroids" or hypotheses to predict ephemerides for a short time into the future with the goal of collecting additional observations to perform orbit determination in the usual sense. Several spin-offs from Milani et al. have been used in recent years for space situational awareness (SSA) of Earth-orbiting satellites (see e.g., [172]).[34]

The exact machinery of Milani et al. is not employed here because it is structurally different (absolute orbits vice relative orbits), but the concept is translated loosely for constraining the relative orbit hypotheses.

5.2.1 Skewness Factor

The skewness factor characterizes the along-track shift or offset from the origin of Hill's frame due to differentials in the orbital elements. Several combinations of differential elements can induce an along-track shift, so the skewness factor captures the bulk effect manifested in disproportionate angles swept out in equal time on opposite sides of the s-axis.

[33] Not discoveries. There is a difference.

[34] The author was the Program Manager at the Air Force Research Laboratory for an effort that exploited these techniques in SSA.

Consider the perfectly coplanar trajectory with a small differential in eccentricity and zero along-track shift as depicted in Figure 26. If observations are made at a constant sample rate, the angles swept out between observations will not be equal throughout the orbit. In the linear equations of motion, the relative velocity is maximized at $s = 0$ and minimized at $\mathrm{r} = 0$ such that ${}^{h}\mathrm{v}_1 = 2\,{}^{h}\mathrm{v}_2$; therefore, the angle swept out along the r-axis is larger than the angle swept out along the s-axis. A symmetry is apparent across the r-axis and s-axis. In this case, we would say the trajectory has zero skewness in the α-profile.

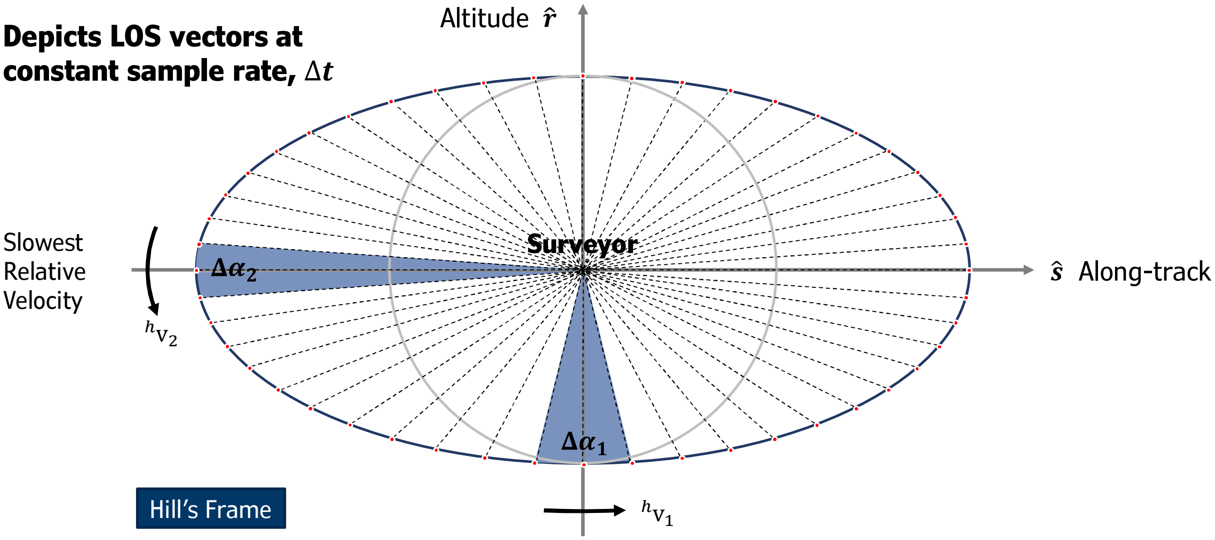


Figure 26: Coplanar NMC Centered in Hill's Frame. LOS vectors are depicted at a constant sample rate to show the symmetry of angles across the r-axis in this coplanar NMC having $\delta e \neq 0$ and zero along-track offset. Equal areas are swept out in equal time, but equal angles are *not* swept out in equal time: $\Delta\alpha_1 > \Delta\alpha_2$. The geometry is exact to show that $\mathrm{v}_1 = 2\mathrm{v}_2$ in this case.

If we introduce an s-axis offset in the relative trajectory (easily done with a differential true anomaly), the accumulation of LOS vectors to the Target is skewed even as the size and period of the relative trajectory remain the same (Figure 27). The linearized relative velocity at $\mathrm{r} = 0$ on both sides of the s-axis are still equal so the Target physically traverses equal distance in equal time, but due to the s-axis offset, the angles swept out along the s-axis in the positive and negative direction are not equal. There is no longer a symmetry in angles across the r-axis. In this case, we would say the trajectory has skewness in the α-profile.

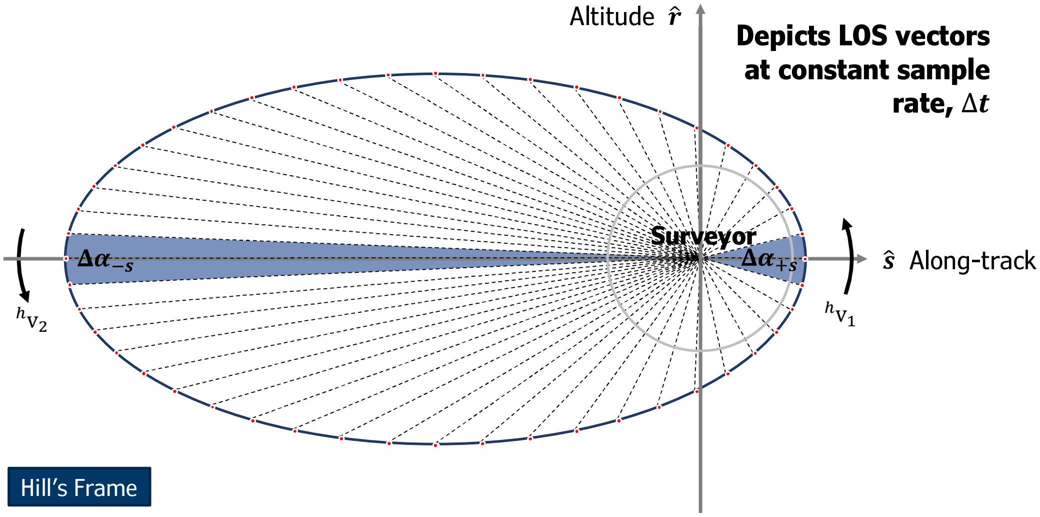


Figure 27: Coplanar NMC with Along-track Shift. LOS vectors are depicted at a constant sample rate to show how the LOS profile is skewed by an along-track shift. The angles are not symmetric across the r-axis and, instead, are concentrated in the direction of maximum range. Neither the angles nor the areas are swept out in equal time.

One way to mathematically define the skewness is by just taking the ratio of angles swept out in equal time on both sides of the s-axis

$$d = \frac{|\alpha(t_2) - \alpha(t_1)|_{+s}}{|\alpha(t_4) - \alpha(t_3)|_{-s}} = \frac{\Delta\alpha_{+s}}{\Delta\alpha_{-s}} \tag{5.1}$$

where $t_2 - t_1 = t_4 - t_3 = \varepsilon$.

The preferred alternative is to define the skewness factor, d, in a way that is bounded on the interval $(0, 2)$ and gives greater intuition when dealing with small differentials in the orbital elements. Under conditions of small differentials, the relative orbit's projection in the rs-plane is analogous to the 2×1 ellipse generated by the Clohessy-Wiltshire equations (see e.g., Figure 16). The approximate geometry of the projected ellipse is depicted in Figure 28.

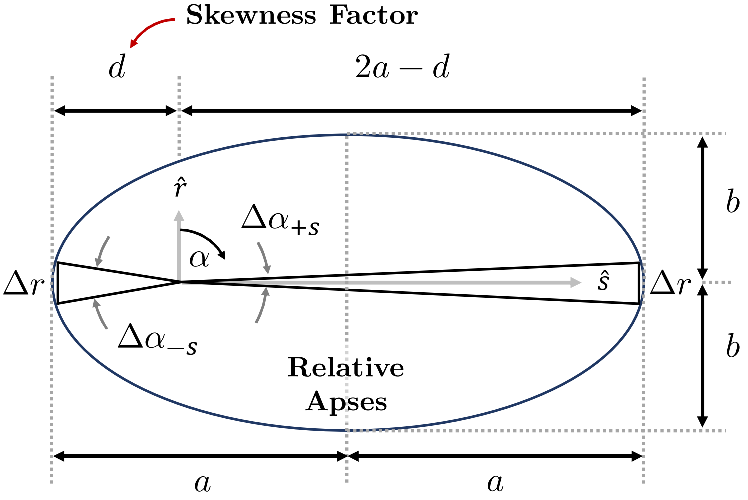


Figure 28: Skewness Factor for NMC Orbits. A "unit" 2×1 ellipse can be used to define d as a fraction of the unit ellipse length.

Assume a "unit" 2×1 ellipse in which $a = 1$ and $b = 0.5$. The objective is to find the value of d as a fraction of the unit ellipse length ($2a$). As was shown in Figure 27, the angles swept out in equal time are not necessarily equal, but approximately equal distances are carved out on short arcs of the projected trajectory in the rs-plane along the s-axis (where $\alpha = \pm 90$). The base triangle distance Δr opposite $\Delta\alpha_{+s}$ and $\Delta\alpha_{-s}$ in Figure 28 designates this phenomenon. $\Delta\alpha_{+s}$ and $\Delta\alpha_{-s}$ must be carefully measured with a constant and high sampling rate for accuracy. The relationship between the angles $\Delta\alpha_{+s}$, $\Delta\alpha_{-s}$, Δr and d are

$$\tan(\Delta\alpha_{-s}/2) = \frac{\Delta r\ /2}{d} \tag{5.2}$$

$$\tan(\Delta\alpha_{+s}/2) = \frac{\Delta r\ /2}{2a - d} \tag{5.3}$$

Combining Equations (5.2) and (5.3), we can eliminate the unknown Δr and solve for the skewness factor

$$\begin{aligned} d\tan(\Delta\alpha_{-s}/2) &= \Delta r\ /2 \\ (2a-d)\tan(\Delta\alpha_{+s}/2) &= \Delta r\ /2 \\ d\tan(\Delta\alpha_{-s}/2) &= (2a-d)\tan(\Delta\alpha_{+s}/2) \\ d[\tan(\Delta\alpha_{-s}/2)+\tan(\Delta\alpha_{+s}/2)] &= 2a\tan(\Delta\alpha_{+s}/2) \\ d &= \frac{2\tan(\Delta\alpha_{+s}/2)}{\tan(\Delta\alpha_{-s}/2)+\tan(\Delta\alpha_{+s}/2)} \end{aligned} \tag{5.4}$$

Using the small angle approximation for tangent, the definition is simplified to

$$d = \frac{2\Delta\alpha_{+s}}{\Delta\alpha_{-s}+\Delta\alpha_{+s}} \tag{5.5}$$

When $d = 1$, there is no skewness. A value on the interval $(0,1)$ indicates the maximum range is in the $+s$-direction ("skews right") and a value on the interval (1,2) indicates the maximum range is in the $-s$-direction ("skews left").

5.3 Method of Generating Hypotheses

Hypotheses are generated within an iterative process that approximates the COEs in a serial fashion. The process, shown in Figure 29, is improved by having a starting guess for the COEs, but it is *not* needed. Two iteration loops are included. The outer loop is initialized with the Surveyor's COEs and the nondimensional admissible region parameters. The inner loop refines the argument of perigee after the semi-major axis has been approximated. For small δa, the inner loop typically does not need iterated. One iteration through the outer loop is also typically sufficient for IOD as will be shown in Chapter 6, but it may be enhanced by using the previous iteration's COEs rather than the Surveyor's at initialization. The source code for the function `admRegHyp`, which generates the admissible hypotheses, is included in Appendix E.6.

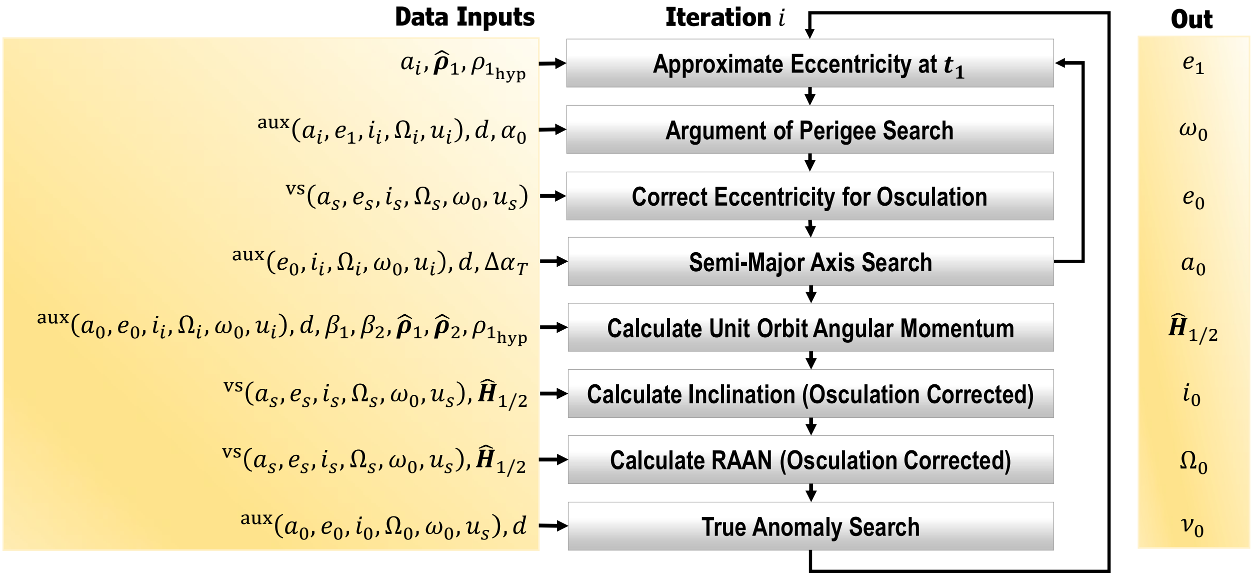

Data Inputs
Iteration i
Out
$a_i, \hat{\boldsymbol{\rho}}_1, \rho_{1_{\text{hyp}}}$
Approximate Eccentricity at $\boldsymbol{t_1}$
e_1
$^{\text{aux}}(a_i, e_1, i_i, \Omega_i, u_i), d, \alpha_0$
Argument of Perigee Search
ω_0
$^{\text{vs}}(a_s, e_s, i_s, \Omega_s, \omega_0, u_s)$
Correct Eccentricity for Osculation
e_0
$^{\text{aux}}(e_0, i_i, \Omega_i, \omega_0, u_i), d, \Delta\alpha_T$
Semi-Major Axis Search
a_0
$^{\text{aux}}(a_0, e_0, i_i, \Omega_i, \omega_0, u_i), d, \beta_1, \beta_2, \hat{\boldsymbol{\rho}}_1, \hat{\boldsymbol{\rho}}_2, \rho_{1_{\text{hyp}}}$
Calculate Unit Orbit Angular Momentum
$\hat{\boldsymbol{H}}_{1/2}$
$^{\text{vs}}(a_s, e_s, i_s, \Omega_s, \omega_0, u_s), \hat{\boldsymbol{H}}_{1/2}$
Calculate Inclination (Osculation Corrected)
i_0
$^{\text{vs}}(a_s, e_s, i_s, \Omega_s, \omega_0, u_s), \hat{\boldsymbol{H}}_{1/2}$
Calculate RAAN (Osculation Corrected)
Ω_0
$^{\text{aux}}(a_0, e_0, i_0, \Omega_0, \omega_0, u_s), d$
True Anomaly Search
ν_0

Figure 29: Hypothesis Generation Process. The COEs are approximated serially with an inner loop to refine the argument of perigee and an outer loop to iterate the entire process. Variables contained within parentheses with an *aux* superscript are used to generate auxiliary orbits and those contained within parentheses with a *vs* superscript are used to generate a virtual Surveyor.

Notice the superscript *aux* in Figure 29. This refers to an "auxiliary orbit" which is not necessarily a hypothesis. Auxiliary orbits are intermediary orbits that are used to evaluate search parameters against admissible region heuristics or to provide geometrically constrained relationships within the admissible region as is the case for calculating the unit orbital angular momentum. The superscript *vs* refers to a "virtual Surveyor" which modifies the argument of perigee in the Surveyor's COE set. The propagated virtual Surveyor is used to correct for osculation in the hypothesis orbital elements.

The individual processes for approximating the orbital element hypotheses are described in the ensuing subsections.

5.3.1 Eccentricity

Upon finding the admissible region parameters, choose a uniform distribution of range hypotheses for specifying the relative position of the target along $\hat{\boldsymbol{\rho}}_1$ (points toward the upper relative apsis) as depicted in Figure 30.

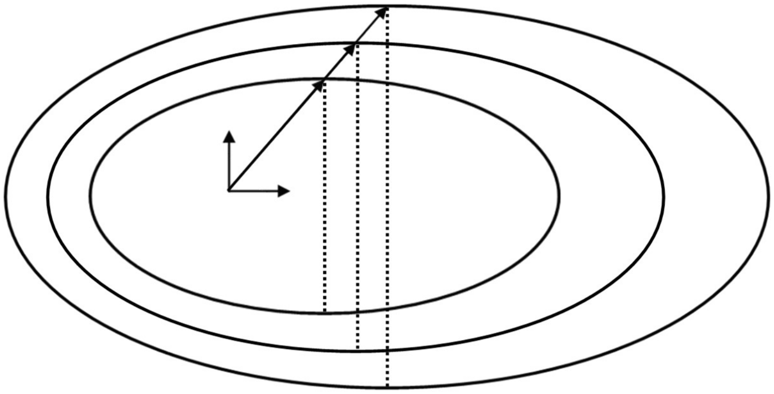


Figure 30: LOS Vector to Upper Relative Apsis. An array of possible ranges along the ${}^h\hat{\rho}_1$ LOS vector pointing to the relative apsis spans the solution space for ${}^i\mathrm{r}_t(t_1)$.

The Target's inertial position vector for each discrete hypothesis is

$$ {}^i\mathbf{r}_t(t_1) = \boldsymbol{R}^{ih}[{}^h\mathbf{r}_s(t_1) + \rho_1\, {}^h\hat{\boldsymbol{\rho}}_1] \tag{5.6}$$

Use $\|{}^i\mathbf{r}_t\|$ and conservation of energy to find $\|{}^i\mathbf{v}_t\|$. On the first iteration, we assume the Surveyor's semi-major axis

$$ \mathrm{v}_t(t_1) = \sqrt{\frac{2\mu}{\mathrm{r}_t(t_1)} - \frac{\mu}{a_t}} \tag{5.7}$$

Recall that the eccentricity vector is

$$ \boldsymbol{e} = \frac{1}{\mu}\big(\mathbf{v} \times (\mathbf{r} \times \mathbf{v})\big) - \frac{\mathbf{r}}{\mathrm{r}} \tag{5.8}$$

Using the triple product expansion, $\boldsymbol{a} \times (\boldsymbol{b} \times \boldsymbol{c}) = \boldsymbol{b}(\boldsymbol{a} \cdot \boldsymbol{c}) - \boldsymbol{c}(\boldsymbol{a} \cdot \boldsymbol{b})$, this can be rewritten as

$$\boldsymbol{e} = \left(\frac{|\mathbf{v}|^2}{\mu} - \frac{1}{|\mathbf{r}|}\right)\mathbf{r} - \frac{\mathbf{r}\cdot\mathbf{v}}{\mu}\mathrm{v} = \begin{bmatrix} \left(\dfrac{\mathrm{v}^2}{\mu} - \dfrac{1}{\mathrm{r}}\right) x - \dfrac{\mathrm{r}\mathrm{v}\cos\theta}{\mu}\dot{x} \\ \left(\dfrac{\mathrm{v}^2}{\mu} - \dfrac{1}{\mathrm{r}}\right) y - \dfrac{\mathrm{r}\mathrm{v}\cos\theta}{\mu}\dot{y} \\ \left(\dfrac{\mathrm{v}^2}{\mu} - \dfrac{1}{\mathrm{r}}\right) z - \dfrac{\mathrm{r}\mathrm{v}\cos\theta}{\mu}\dot{z} \end{bmatrix} \tag{5.9}$$

where θ is the angle between $\mathbf{r}$ and $\mathbf{v}$. The magnitude of eccentricity is

$$e = \sqrt{e_x^2 + e_y^2 + e_z^2} \tag{5.10}$$

The square of each element of the eccentricity vector is

$$e_x^2 = \frac{x^2}{\mathrm{r}^2} - \frac{2x^2\mathrm{v}^2}{\mu\mathrm{r}} + \frac{x^2\mathrm{v}^4}{\mu^2} + \frac{2x\dot{x}\mathrm{v}\cos\theta}{\mu} - \frac{2\mathrm{r}x\dot{x}\mathrm{v}^3\cos\theta}{\mu^2} + \frac{\mathrm{r}^2\dot{x}^2\mathrm{v}^2\cos^2\theta}{\mu^2} \tag{5.11}$$

$$e_y^2 = \frac{y^2}{\mathrm{r}^2} - \frac{2y^2\mathrm{v}^2}{\mu\mathrm{r}} + \frac{y^2\mathrm{v}^4}{\mu^2} + \frac{2y\dot{y}\mathrm{v}\cos\theta}{\mu} - \frac{2\mathrm{r}y\dot{y}\mathrm{v}^3\cos\theta}{\mu^2} + \frac{\mathrm{r}^2\dot{y}^2\mathrm{v}^2\cos^2\theta}{\mu^2} \tag{5.12}$$

$$e_z^2 = \frac{z^2}{\mathrm{r}^2} - \frac{2z^2\mathrm{v}^2}{\mu\mathrm{r}} + \frac{z^2\mathrm{v}^4}{\mu^2} + \frac{2z\dot{z}\mathrm{v}\cos\theta}{\mu} - \frac{2\mathrm{r}z\dot{z}\mathrm{v}^3\cos\theta}{\mu^2} + \frac{\mathrm{r}^2\dot{z}^2\mathrm{v}^2\cos^2\theta}{\mu^2} \tag{5.13}$$

Combine terms in Equation (5.10)

$$\begin{aligned} e^2 = 1 - \frac{2\mathrm{v}^2\mathrm{r}}{\mu} + \frac{\mathrm{v}^4\mathrm{r}^2}{\mu^2} + \left(\frac{2\mathrm{v}}{\mu} - \frac{2\mathrm{r}\mathrm{v}^3}{\mu^2}\right)(x\dot{x} + y\dot{y} + z\dot{z})\cos\theta \\ + \frac{\mathrm{r}^2\mathrm{v}^2}{\mu^2}(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)\cos^2\theta \end{aligned} \tag{5.14}$$

For low eccentricity orbits, $\cos\theta \ll 1$ and for low differential argument of perigee $\cos\theta \approx 0$ near the relative apsis, so we make a small error by assuming the magnitude of the Target's eccentricity for each of the possible combinations of $\mathrm{r}_t(t_1)$ and $\mathrm{v}_t(t_1)$

$$e_1 \cong \sqrt{1 - \frac{2\mathrm{v}_t^2\mathrm{r}_t}{\mu} + \frac{\mathrm{v}^4\mathrm{r}_t^2}{\mu^2}} \tag{5.15}$$

Notice that we have added the subscript 1 to indicate the approximation of eccentricity at time t_1 when $\hat{\boldsymbol{\rho}}_1$ was obtained. We actually want the eccentricity at epoch, e_0. The full geopotential causes eccentricity to osculate throughout the orbit, so e_1 needs an osculation correction back to t_0. If the Target and Surveyor have identical arguments of perigee, the osculation phase and magnitude will be approximately equal for both Satellites, and knowledge of the Surveyor's eccentricity osculation can be used to correct e_1. If the argument of perigee for the two satellites are different, then we can create a modified "virtual Surveyor" that uses an approximation of the Target's argument of perigee (determined in the next section) to find a reasonably accurate osculation correction. The epoch eccentricity is now possible with

$$e_0 \cong e_1 - ({}^{\mathrm{vs}}e_1 - {}^{\mathrm{vs}}e_0) \tag{5.16}$$

where the superscript vs has been added to indicate the virtual Surveyor.

Our assumption about $\cos\theta$ to approximate e_1 begins to break down when dealing with high differential argument of perigee, especially for higher eccentricities. If the IOD process succeeds in iterating upon the argument of perigee enough that $\hat{\boldsymbol{\rho}}_1$ can be redefined near perigee or apogee, then the $\cos\theta$ assumption is restored and the IOD process has greater utility than for low eccentricity orbits.

5.3.2 Argument of Perigee

The argument of perigee is very difficult to approximate with so little information at this stage, so we crudely approximate it with our knowledge of d and α_0 at epoch. Differential argument of perigee can easily be confused with a combination of differential eccentricity and differential true anomaly, especially in low-eccentricity orbits. In that sense, a differential argument of perigee influences the relative orbit's scale and along-track shift. By getting close enough to the true argument of perigee, any remaining along-track error can be absorbed into the true anomaly to match the skewness. The remaining scale error is manifested as a scale bias in the hypotheses. This can be improved upon through iteration.

Generate a cluster of 36 auxiliary trajectories by using the early approximation for eccentricity (e_1) and inheriting the Surveyor's semi-major axis, inclination and RAAN on the first iteration. Vary the Surveyor's argument of perigee in 10-degree increments from zero to 350 degrees. Since the problem is constrained to NMC trajectories, the Target must have a similar argument of latitude, $u = \omega + \nu$, so a coarse initial estimate of the auxiliary true anomaly can be found, but is not sufficient. Perform a convex optimization of true anomaly for each auxiliary trajectory by using the skewness factor as the heuristic (see Section 5.3.6). Upon matching the skewness, compare α_0 from the auxiliary orbit with that of the admissible region. Auxiliary orbits with the lowest errors in α_0 (the criteria $|\Delta\alpha_0| < 15°$ seems to suffice) are candidates for argument of perigee.

An additional "sanity check" is performed by comparing the value of ρ_1 from the candidate auxiliary orbit with the original ρ_1 hypothesis. The sanity check helps eliminate spurious candidates. For very low δe cases, an error in eccentricity can inadvertently induce a 180° phase shift in argument of perigee and true anomaly in order to sustain the correct LOS vector. The sanity check can identify when this happens, so an equal weighting may be given to the minimization of $|\Delta\alpha_0|$ and the ratio $\left|\rho_{1,\text{hyp}} - \rho_{1,\text{aux}}\right|/\rho_{1,\text{hyp}}$.

Let us put this into practice to make it clearer. Consider Case 01 from Table 3. In this simple case, the Surveyor and Target have identical arguments of perigee ($\omega = 102.64716$ deg) and arguments of latitude ($u = 102.64716$ deg), but we have no way of knowing this in practice! All we know is that the skewness factor is $d = 1$ and $\alpha_0 = 180°$. By following the process laid out above, we can see how α_0 changes across the different auxiliary trajectories shown in Figure 31. Each of the colored lines indicates the LOS vector at epoch. Clearly, α_0 is a function of the Target's argument of perigee at epoch. Select the argument of perigee with the smallest error in α_0 from this set. Precision is not important at this stage, but a price is paid for the granularity. Notice how scale is embedded in argument of perigee with the different sized ellipses! We inevitably assume a small range bias on the first iteration.

This search is only performed for the first hypothesis. All hypotheses thereafter use the same argument of perigee. Once the argument of perigee has been set, return to correct the eccentricity for osculation. This technique must be repeated upon estimating the semi-major axis if $\delta a \neq 0$. Two loops through the argument of perigee and semi-major axis search algorithm (Section 5.3.3) are typically sufficient to get within 10 degrees of ω_t.

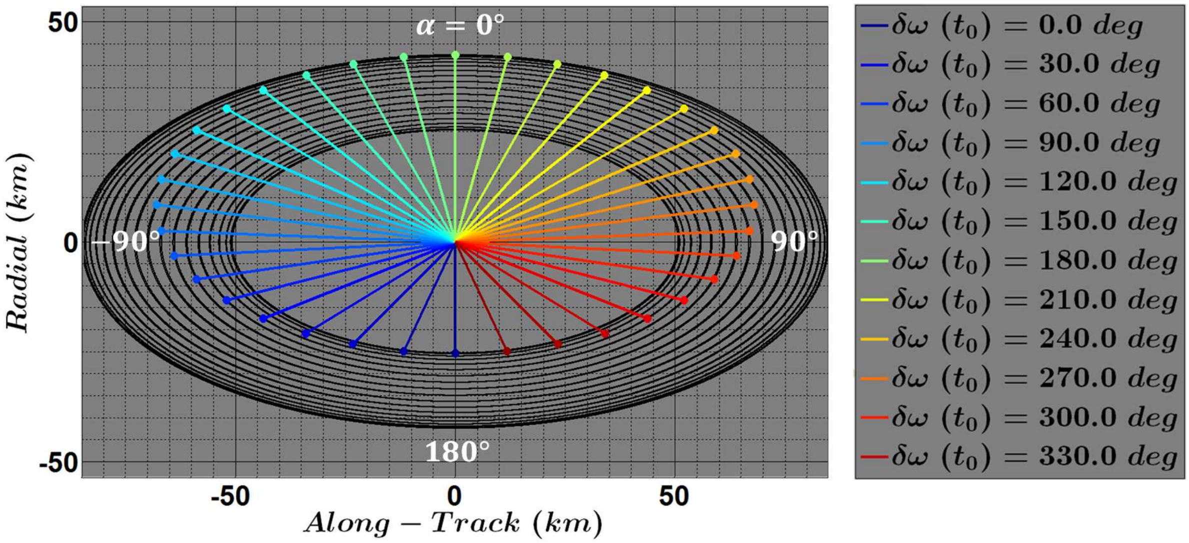


Figure 31: Demonstration of α_0 Variation from Argument of Perigee Search. 36 auxiliary trajectories (black curves) are propagated with $\delta\omega$ intervals of 10 deg in argument of perigee at epoch. The true anomaly is also varied in order to match the known skewness factor. The colored lines show the LOS vector at epoch for each of the trajectories.

5.3.3 Semi-Major Axis

The orbital energy or period of the Target must resemble that of the Surveyor if an NMC trajectory is being used to conduct proximity operations. A differential semi-major axis ($\delta a = a_t - a_s$) can be approximated from knowledge of d and $\Delta\alpha_T$

$$\Delta\alpha_T = \begin{cases} \alpha_0 - \alpha_T + 2\pi & \text{if } \alpha_0 < 0 \ \wedge \ \alpha_T > 0 \ \wedge \ \text{count}(s=0) > 1 \\ \alpha_0 - \alpha_T & \text{if } \alpha_0 > 0 \ \wedge \ \alpha_T > 0 \ \wedge \ \text{count}(s=0) > 1 \\ \alpha_0 - \alpha_T & \text{if } \alpha_0 < 0 \ \wedge \ \alpha_T < 0 \ \wedge \ \alpha_0 > \alpha_T \ \wedge \ \text{count}(s=0) > 1 \\ \alpha_0 - \alpha_T - 2\pi & \text{if } \alpha_0 > 0 \ \wedge \ \alpha_T < 0 \ \wedge \ \text{count}(s=0) = 1 \end{cases} \tag{5.17}$$

where α_T is the angle after one Kozai period of the Surveyor, α_0 is the angle at epoch and count$(s = 0)$ indicates the number of times the α-profile crosses 0 or 180 degrees during one Kozai period.

The Kozai period [163] accounts for the Earth's oblateness by including the anomalistic frequency correction from Equation (4.46)

$$T = \frac{2\pi}{n + \dot{M}_0} \tag{5.18}$$

where the mean motion is $n = \sqrt{\mu / a_s^3}$ and a_s is the Surveyor's semi-major axis.

In a Keplerian system, $\Delta\alpha_T = 0$ when the Target and Surveyor have the same semi-major axis because the NMC trajectory returns to the same starting point after one orbit. For a non-Keplerian system, secular drift will cause a small $\Delta\alpha_T$ in the two-body period of the orbit even when $\delta a = 0$. When $\delta a \neq 0$, the relative trajectory drifts forward or backward along the s-axis and $\Delta\alpha_T$ grows as δa grows. If the differential semi-major axis grows too large (on the order of kilometers), the Target will not be able to circumnavigate the Surveyor for very long.

By convention, the sign of $\Delta\alpha_T$ is the opposite of the sign of δa. If $\Delta\alpha_T$ is positive, the α profile spans more than 360 degrees in one period which means the Target's bulk along-track motion is faster than the Surveyor (Target appears to be moving forward over several orbits). Therefore, the Target has a smaller semi-major axis than the Surveyor has; δa is negative. Figure 32 indicates how $\Delta\alpha_T$ changes with $\delta a = -1$ km intervals for LEO Case 01 (COEs listed in Table 3). This is referred to as a positive α disparity in one Kozai period.

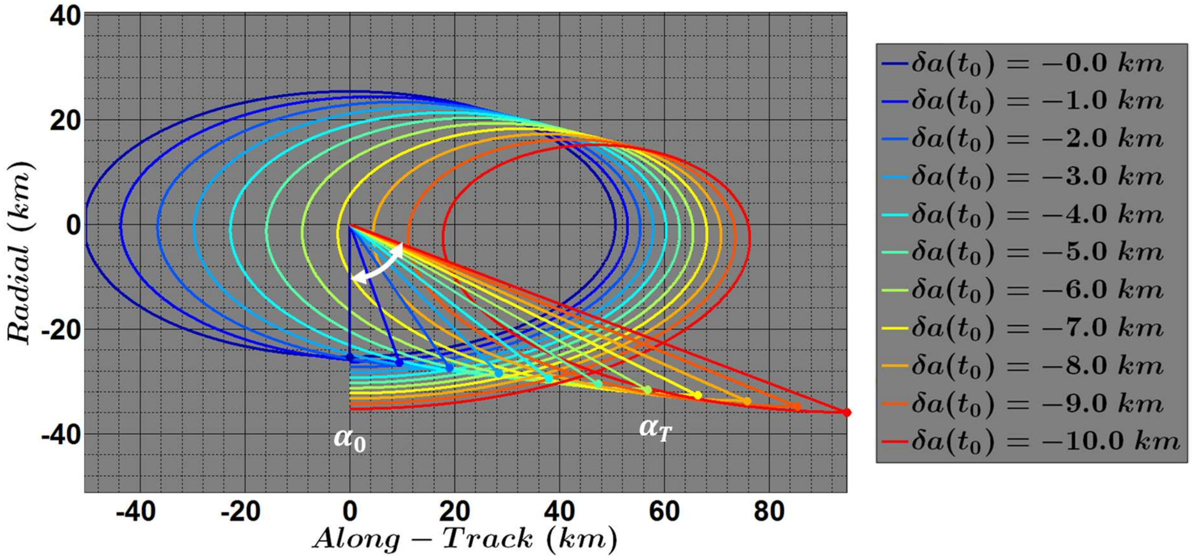


Figure 32: Positive Alpha Disparity. Eleven auxiliary trajectories are propagated with δa intervals of -1 km at epoch. Notice that α_0 is the same for all trajectories. The colored lines show the LOS vector after one Kozai period for each of the trajectories. The disparity between α_T and α_0 is $\Delta\alpha_T$.

If $\Delta\alpha_T$ is negative, the α profile spans less than 360 degrees in one period which means the Target's bulk along-track motion is slower than the Surveyor (Target appears to be moving backward over several orbits). Therefore, the Target has a larger semi-major axis than the Surveyor has; δa is positive. Figure 33 indicates how $\Delta\alpha_T$ changes with $\delta a = +1$ km intervals for LEO Case 01 (COEs listed in Table 3). This is referred to as a negative α disparity in one Kozai period.

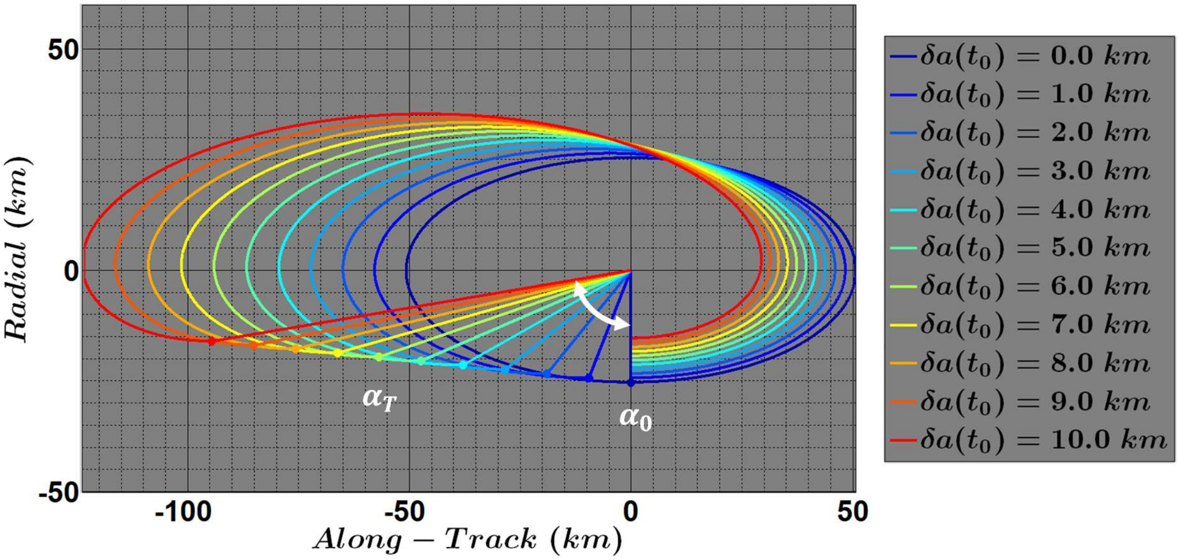


Figure 33: Negative Alpha Disparity. Eleven auxiliary trajectories are propagated with δa intervals of +1 km at epoch. Notice that α_0 is the same for all trajectories. The colored lines show the LOS vector after one Kozai period for each of the trajectories. The disparity between α_T and α_0 is $\Delta\alpha_T$.

The method for approximating the semi-major axis involves another convex optimization search of auxiliary trajectories with $\Delta\alpha_T$ as the heuristic. Each auxiliary trajectory inherits the Surveyor's inclination and RAAN. The semi-major axis is varied and the eccentricity is re-calculated for each semi-major axis hypothesis. The argument of perigee is fixed from the previous step and the true anomaly is adjusted for each auxiliary trajectory to match the skewness factor (function `skew2nu`). Compare $\Delta\alpha_T$ from the auxiliary orbit with that of the admissible region.

Rather than propagating a large cluster of auxiliary trajectories at once, there is an optimal way to search by first using 1 km intervals in δa. The sign of the observed $\Delta\alpha_T$

typically indicates the opposite sign on the interval δa.[35] Upon bypassing $\Delta\alpha_T$, take one step backward and increment by 100 m intervals in δa until bypassing $\Delta\alpha_T$. If the first step of +1 km already exceeds $\Delta\alpha_T$, you did not abide by the sign convention of $\Delta\alpha_T$ and need to increment in the opposite direction. Presently, we limit the precision of δa to 100 m to speed up the process, although, future iterations may find value in greater precision.

The auxiliary orbit with the lowest error in $\Delta\alpha_T$ is the candidate semi-major axis and the eccentricity is finally fixed for this hypothesis. This process is avoided if $\Delta\alpha_T < 2.5°$ in which case the Surveyor's semi-major axis is assumed. The effect of Earth's oblateness can be easily confused with small angle disparities, so we presently accept any errors associated with this threshold value on $\Delta\alpha_T$.

5.3.4 Inclination

The Target's inclination is solved analytically from $\boldsymbol{\rho}_1$ and $\boldsymbol{\rho}_2$. The range hypothesis along $\hat{\boldsymbol{\rho}}_1$ has given us half of the solution, but $\hat{\boldsymbol{\rho}}_2$ must be scaled such that the projection of the hypothesis trajectory onto the rs-plane roughly matches a coplanar auxiliary trajectory that uses the same eccentricity, argument of perigee, and semi-major axis previously found for $\boldsymbol{\rho}_1$. The true anomaly is also temporarily set to match the skewness factor. The magnitudes of the projections of $\boldsymbol{\rho}_2$ and $\boldsymbol{\rho}_1$ onto the rs-plane are just

$$\rho_2' = \rho_2 \cos\beta_2 \tag{5.19}$$

$$\rho_1' = \rho_1 \cos\beta_1 \tag{5.20}$$

[35] The Earth's oblateness causes some along-track drift, so it is possible to have $\delta a = 0$ and $\Delta\alpha_T \neq 0$ or vice versa. We presently do not try to calibrate for this; although, it could be done if the precision was important (it is not at this stage of IOD). Just adjust δa to try to match $\Delta\alpha_T$.

Figure 34 shows the geometry for the projections of $\boldsymbol{\rho}_2$ and $\boldsymbol{\rho}_1$. The projected values should be approximately equal to the time synchronized set ρ_1'' and ρ_2'' from the auxiliary trajectory

$$\rho_2' \cong \rho_2'' \tag{5.21}$$

$$\rho_1' \cong \rho_1'' \tag{5.22}$$

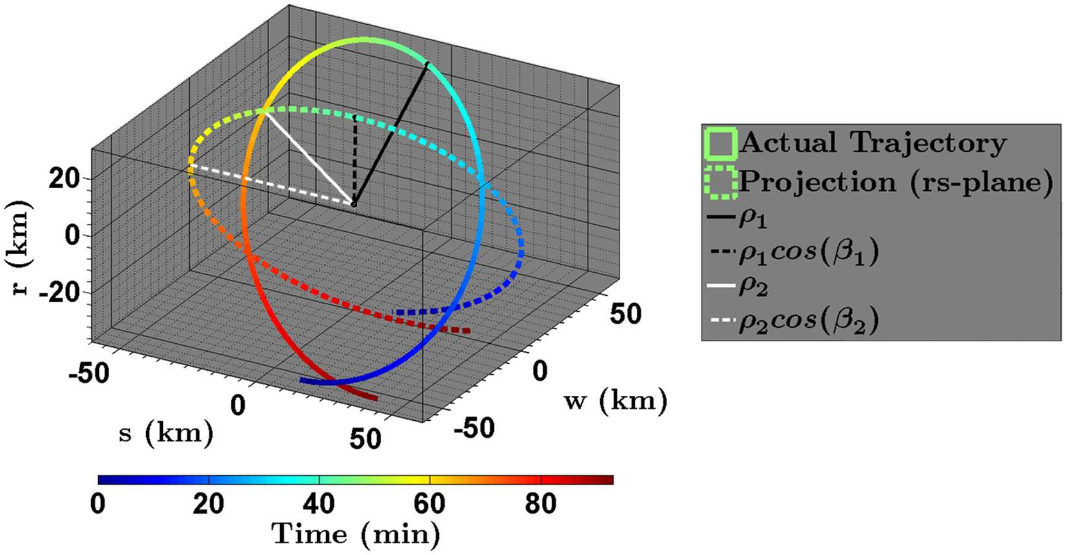


Figure 34: Range Projections in the *rs*-plane. The actual relative trajectory is depicted with the solid rainbow-colored curve (the colorbar indicates time). The projected trajectory in the *rs*-plane is depicted with the dashed rainbow-colored curve. The solid black line is ρ_1 and the dashed black line depicts its projection into the rs-plane. The solid white line is ρ_2 and the dashed white line depicts its projection into the rs-plane.

The values of $\boldsymbol{\rho}_1$ and $\boldsymbol{\rho}_2$, combine with the Surveyor's inertial knowledge to give us two position vectors in the Target's hypothesized orbital plane.

$${}^{i}\mathbf{r}_1 = \boldsymbol{R}^{ih}[{}^{h}\mathbf{r}_s(t_1) + {}^{h}\boldsymbol{\rho}_1] \tag{5.23}$$

$${}^{i}\mathbf{r}_2 = \boldsymbol{R}^{ih}[{}^{h}\mathbf{r}_s(t_2) + {}^{h}\boldsymbol{\rho}_2] \tag{5.24}$$

The Target's orbit inclination is classically defined as

$$\cos i = \frac{\hat{\boldsymbol{k}} \cdot \boldsymbol{H}}{|\boldsymbol{H}|} = \hat{\boldsymbol{k}} \cdot \widehat{\boldsymbol{H}} = \hat{\boldsymbol{k}} \cdot ({}^{i}\hat{\mathbf{r}}_t \times {}^{i}\hat{\mathbf{v}}_t) \tag{5.25}$$

where $\hat{\boldsymbol{k}}$ is the unit vector along the z-axis of the ECI frame and $\boldsymbol{H}$ is the Target's orbit angular momentum. ${}^{i}\hat{\mathbf{r}}_t$ and ${}^{i}\hat{\mathbf{v}}_t$ lie in the orbital plane, but are unknown if the range to the Target is unknown. Equivalently, we can find $\widehat{\boldsymbol{H}}$ from the cross product of two inertial position vectors (${}^{i}\mathbf{r}_1$ and ${}^{i}\mathbf{r}_2$) since they both lie in the orbital plane, but since the cross product is not commutative, some care is required

$$\widehat{\boldsymbol{H}}_{1/2} = \begin{cases} \dfrac{{}^{i}\mathbf{r}_1 \times {}^{i}\mathbf{r}_2}{|{}^{i}\mathbf{r}_1 \times {}^{i}\mathbf{r}_2|} & \text{if } (t_1 < t_2 \ \wedge \ |\alpha_1| + |\alpha_2| < \pi) \ \vee \ (t_1 > t_2 \ \wedge \ |\alpha_1| + |\alpha_2| > \pi) \\ \dfrac{{}^{i}\mathbf{r}_2 \times {}^{i}\mathbf{r}_1}{|{}^{i}\mathbf{r}_2 \times {}^{i}\mathbf{r}_1|} & \text{if } (t_1 > t_2 \ \wedge \ |\alpha_1| + |\alpha_2| < \pi) \ \vee \ (t_1 < t_2 \ \wedge \ |\alpha_1| + |\alpha_2| > \pi) \end{cases} \tag{5.26}$$

Now we can solve for inclination. An osculation correction is needed to find the inclination at epoch. Since we use two inertial position vectors at different times to generate $\widehat{\boldsymbol{H}}_{1/2}$, we apply an averaging scheme for the correction

$$i_a = i - \frac{({}^{\text{vs}}i_1 + {}^{\text{vs}}i_2 - 2\,{}^{\text{vs}}i_0)}{2} \tag{5.27}$$

$$i_b = i - \left({}^{\text{vs}}i\left(\frac{t_1 + t_2}{2}\right) - {}^{\text{vs}}i_0\right) \tag{5.28}$$

$$i_0 = \frac{i_a + i_b}{2} \tag{5.29}$$

where ${}^{\text{vs}}i$ is the inclination of our virtual Surveyor described in Section 5.3.1.

Due to uncertainty in the LOS vectors and non-Keplerian dynamics, other sets of inertial position vectors could be used in a least squares framework, but we currently only use ${}^{i}\mathbf{r}_1$ and ${}^{i}\mathbf{r}_2$.

5.3.5 Right Ascension of the Ascending Node

Once the unit angular momentum vector for the hypothesis is found for the inclination, the line of nodes can be found and the RAAN can be ascertained. The two-body line of nodes is defined as

$$\hat{\boldsymbol{n}} = \frac{\boldsymbol{k} \times \boldsymbol{H}}{|\boldsymbol{k} \times \boldsymbol{H}|} = \hat{\boldsymbol{k}} \times \widehat{\boldsymbol{H}} \tag{5.30}$$

Since the line of nodes is in the equator plane, it must have the form

$$\hat{\boldsymbol{n}} = \cos\Omega\,\hat{\boldsymbol{\imath}} + \sin\Omega\,\hat{\boldsymbol{\jmath}} \tag{5.31}$$

where Ω is the familiar angle for RAAN.

An osculation and regression correction is needed to find the RAAN at epoch. As with the inclination, we apply an averaging scheme for the correction

$$\Omega_a = \Omega - \frac{({}^{\mathrm{vs}}\Omega_1 + {}^{\mathrm{vs}}\Omega_2 - 2\,{}^{\mathrm{vs}}\Omega_0)}{2} \tag{5.32}$$

$$\Omega_b = \Omega - \left({}^{\mathrm{vs}}\Omega\left(\frac{t_1 + t_2}{2}\right) - {}^{\mathrm{vs}}\Omega_0\right) \tag{5.33}$$

$$\Omega_0 = \frac{\Omega_a + \Omega_b}{2} \tag{5.34}$$

where ${}^{\mathrm{vs}}\Omega$ is the RAAN of our virtual Surveyor described in Section 5.3.1 and Ω_0 is the hypothesis RAAN at epoch.

5.3.6 True Anomaly

A differential true anomaly has the effect of shifting the trajectory along the s-axis and skewing the LOS vectors in the direction of the shift as shown in Figure 35. A differential argument of perigee and differential RAAN have a similar effect, but they were determined in a sequence that allowed errors in $\delta\omega$ and $\delta\Omega$ to be absorbed by $\delta\nu$ such that the skewness factor is matched in each previous step with a convex optimization of ν in the auxiliary orbits. The function skew2nu (Appendix E.7) performs the optimization by adjusting $\delta\nu$ with d as the heuristic.

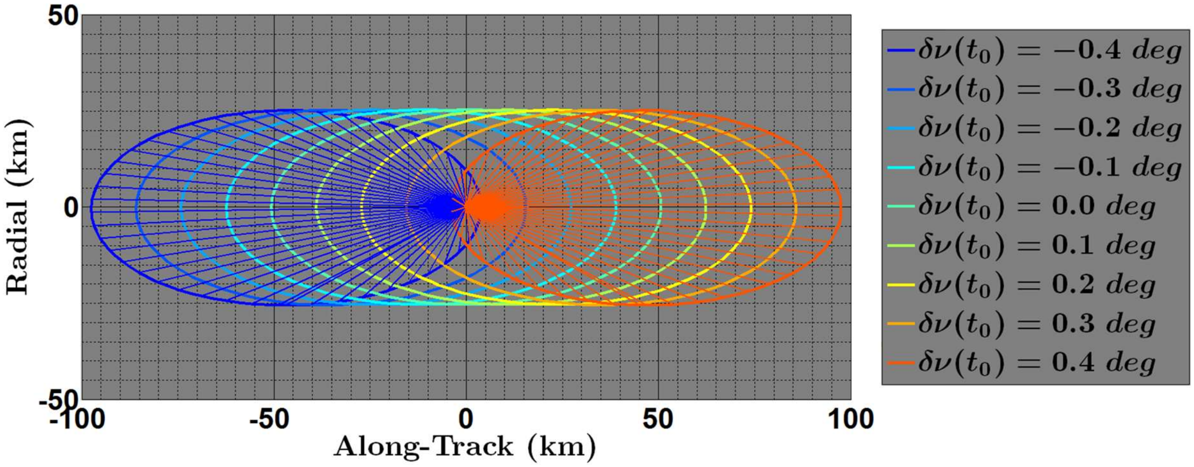


Figure 35: Along-Track Shift from Differential True Anomaly. Nine auxiliary trajectories are propagated with $\delta\nu$ intervals of 0.1 deg at epoch. Notice how the trajectories are shifted in the along-track direction and the LOS profile is skewed in the direction of the shift.

After all other elements have been set, perform a final convex optimization of true anomaly. The search cost can be reduced by starting close to the solution. Since the Target will have a similar argument of latitude as that of the Surveyor (the differential argument of latitude, $\delta u = \delta\omega + \delta\nu$, must be small for an NMC) start with a first guess of

$$\nu_t' = u_s - \omega_t \tag{5.35}$$

Then search near ν_t' with initial intervals of $\delta\nu = 0.1$ deg. A coarser interval can be used, but the difference in argument of latitude should not be larger than a couple degrees at most for NMC trajectories within the ranges of interest for LOS navigation. An auxiliary orbit is propagated for each true anomaly interval to compare its skewness factor with the observed skewness factor. If the inequality in Equation (5.36) is met, take one step backward and continue the search forward with intervals an order of magnitude smaller until the inequality in Equation (5.36) is met again. We repeat this process until achieving precision $\delta\nu = 1 \times 10^{-5}$.

$$\operatorname{sgn}(\delta\nu)[d_{\text{aux}} - d_{\text{obs}}] < 0 \tag{5.36}$$

where d is the skewness factor. If the first coarse step already satisfies the inequality, reverse the sign of $\delta\nu$ and start again. At the end of this process the true anomaly at epoch for this hypothesis is

$$\nu_0 = \nu' + \delta\nu \tag{5.37}$$

Readers may be tempted to use the true anomaly from the two-body orbit equation, but it is too sensitive to errors in a, e, and r to approximate the solution. The inverse cosine in Equation (5.39) has a habit of rendering imaginary numbers if attempted

$$r = \frac{a(1-e^2)}{1 + e\cos\nu} \tag{5.38}$$

$$\nu = \cos^{-1}\left(\frac{a(1-e^2)}{re} - \frac{1}{e}\right) \tag{5.39}$$

Once the full set of COEs are known, a sanity check can be performed to eliminate bad hypotheses. Simply propagate the hypothesis to t_1 and compare $\rho(t_1)$ with the initial range hypothesis, ρ_1. If the error is greater than 10% on the first iteration, the argument of perigee is the likely culprit and will need to be refined through iteration.

5.3.7 Verification

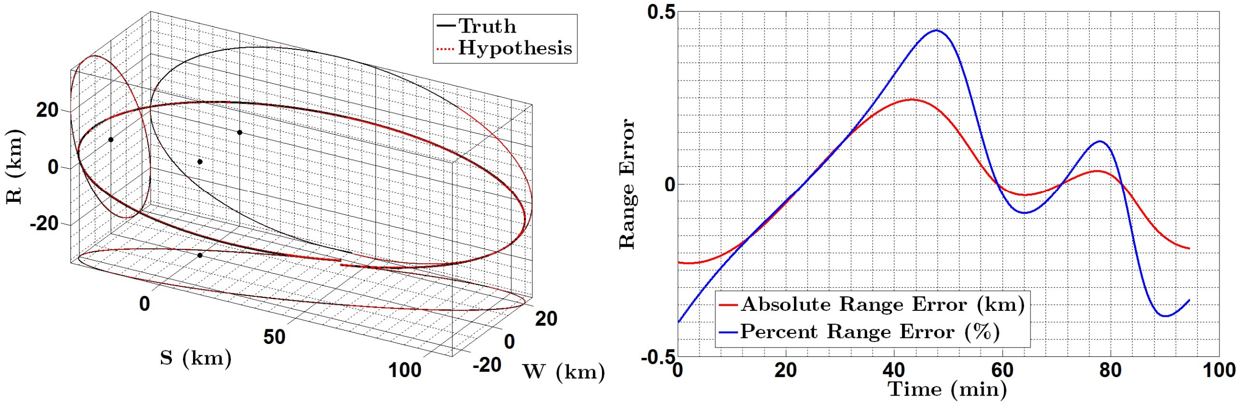
Two LEO cases are selected to demonstrate the accuracy of the hypothesis generation methodology after a *single iteration.* One uses a low differential argument of perigee and the other uses a high differential argument of perigee since this is the only element that is presently coarsely determined. Rather than selecting a random range for ρ_1, we use the true range and the admissible region parameters to see how well the relative trajectory is reconstructed. Each case is performed twice: once with just two-body dynamics and a second time with EGM96 20×20 dynamics. No uncertainty in observations or dynamics are assumed during verification.

LEO, Low Differential Argument of Perigee, Two-Body (Case 06A)

Table 4: Case 06A: Orbital Elements for Truth and Hypothesis

COEs	Surveyor	Target Truth	Hypothesis	Hypothesis Error
a (km)	6789.085640626	6790.085640626	6790.082937106	-0.002703519647
e (nd)	0.001601186396	0.006601186396	0.006601330019	0.000000143623
i (deg)	51.75241867994	51.95241867994	51.95271797764	0.000299297694
Ω (deg)	72.49304954653	72.59304954653	72.59266029623	-0.000389250305
ω (deg)	60.69588076088	60.49588076088	60.69588076088	0.200000000000
ν (deg)	0.000000000000	0.500000000000	0.297505000000	-0.202495000000

The hypothesis is plotted in Figure 36(a) with the truth. The lines are difficult to discriminate because of a relatively small range error shown in Figure 36(b). Notice the argument of perigee error in Table 4 from the coarse approximation. Any error in argument of perigee manifests in both scale and along-track shift. The true anomaly absorbs the along-track shift error in order to match the skewness factor in the admissible region. The scale error can be reduced with refinements in the argument of perigee on a second iteration if the results are less desirable than these.



(a) Trajectories & Projections

(b) Range Error of Hypothesis

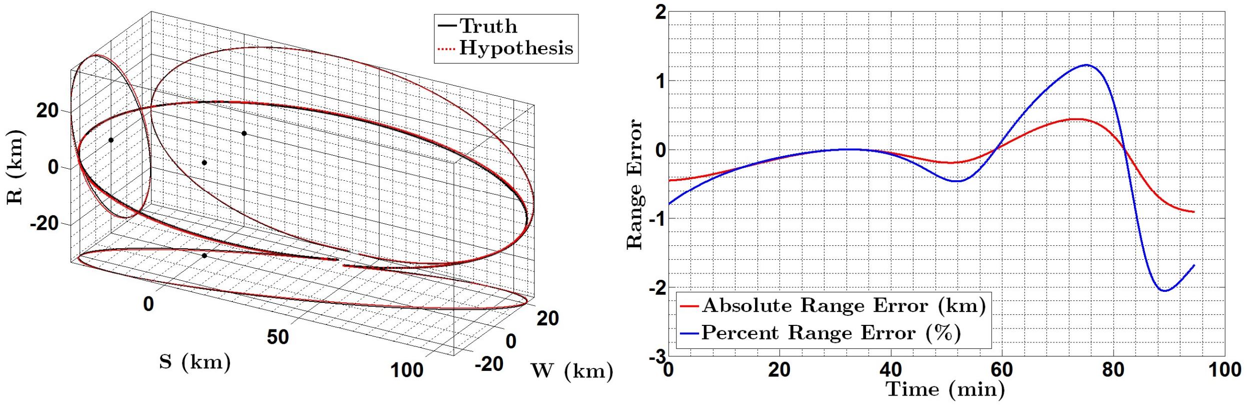
Figure 36: Hypothesis Accuracy for Case 06A. The trajectories are plotted in (a) with their projections in all three directions to gain perspective of the in-plane and out-of-plane motion. The black curves are the truth and the dashed red curves are the hypothesis. The black dots indicate the location of the Surveyor for the actual and projected trajectories. The range error is plotted in (b) as an absolute error (red curve) and as a percentage of the truth (blue curve).

LEO, Low Differential Argument of Perigee, EGM96 20×20 (Case 06B)

Table 5: Case 06B: Orbital Elements for Truth and Hypothesis

COEs	Surveyor	Target Truth	Hypothesis	Hypothesis Error
a (km)	6789.085640626	6790.085640626	6790.151484207	0.065843581052
e (nd)	0.001601186396	0.006601186396	0.006577644639	-0.000023541757
i (deg)	51.75241867994	51.95241867994	51.94994191317	-0.002476766772
Ω (deg)	72.49304954653	72.59304954653	72.60618845410	0.013138907563
ω (deg)	60.69588076088	60.49588076088	60.69588076088	0.200000000000
ν (deg)	0.000000000000	0.500000000000	0.290945000000	-0.209055000000

The hypothesis is plotted in Figure 37(a) with the truth. The lines are difficult to discriminate because of a relatively small range error shown in Figure 37(b). Notice the argument of perigee error in Table 5 from the coarse approximation. Any error in argument of perigee manifests in both scale and along-track shift. The true anomaly absorbs the along-track shift error in order to match the skewness factor in the admissible region. The scale error could be reduced with refinements in the argument of perigee on a second iteration. The results are only slightly worse than the two-body hypothesis. This is because of additional error from osculation and regression in the elements that weren't corrected perfectly from our virtual Surveyor.



(a) Trajectories & Projections

(b) Range Error of Hypothesis

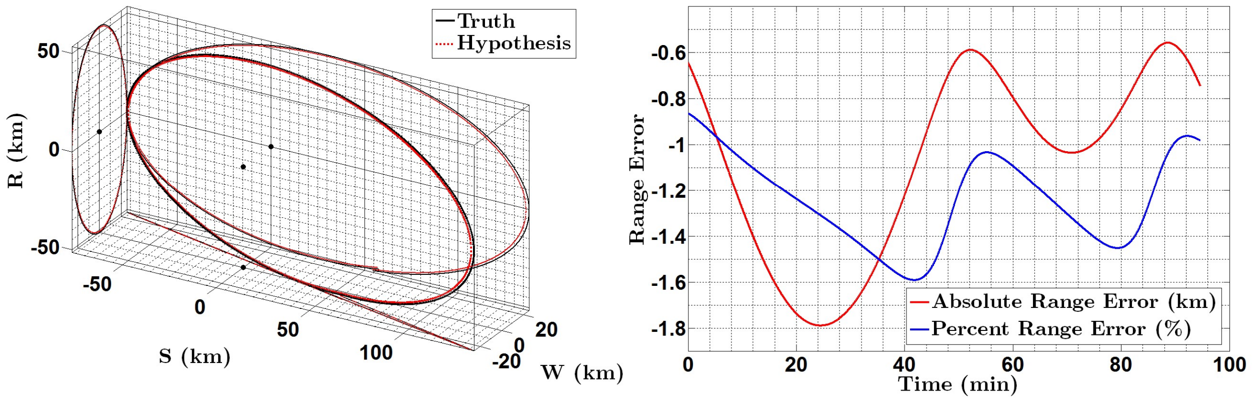
Figure 37: Hypothesis Accuracy for Case 06B. The trajectories are plotted in (a) with their projections in all three directions to gain perspective of the in-plane and out-of-plane motion. The black curves are the truth and the dashed red curves are the hypothesis. The black dots indicate the location of the Surveyor for the actual and projected trajectories. The range error is plotted in (b) as an absolute error (red curve) and as a percentage of the truth (blue curve).

LEO, High Differential Argument of Perigee, Two-Body (Case 08A)

Table 6: Case 08A: Orbital Elements for Truth and Hypothesis

COEs	Surveyor	Target Truth	Hypothesis	Hypothesis Error
a (km)	6789.085640626	6790.085640626	6790.082875046	-0.002765579951
e (nd)	0.001601186396	0.006601186396	0.006509836360	-0.000091350035
i (deg)	51.75241867994	51.95241867994	51.94991380729	-0.002504872648
Ω (deg)	72.49304954653	72.59304954653	72.59176309648	-0.001286450058
ω (deg)	60.69588076088	191.0958807608	190.6958807608	0.600000000000
ν (deg)	150.0000000000	20.00000000000	20.39873464991	0.398734649913

The hypothesis is plotted in Figure 38(a) with the truth. The range error shown in Figure 38(b) is higher than the low differential argument of perigee case. This additional error is partially due to the eccentricity since the $\cos\theta \approx 0$ assumption from Equation (5.14) is better for low differential argument of perigee near the relative apsis. The argument of perigee is also introducing some scale error from the coarse approximation, but the argument of perigee routine performed as it should with 10 deg precision despite the remaining error. As before, the true anomaly approximation absorbed the along-track shift error from the argument of perigee in order to match the skewness factor in the admissible region. It is believed that the slightly higher inclination and RAAN error is due to a less accurate auxiliary trajectory from the scaling error.



(a) Trajectories & Projections **(b) Range Error of Hypothesis**

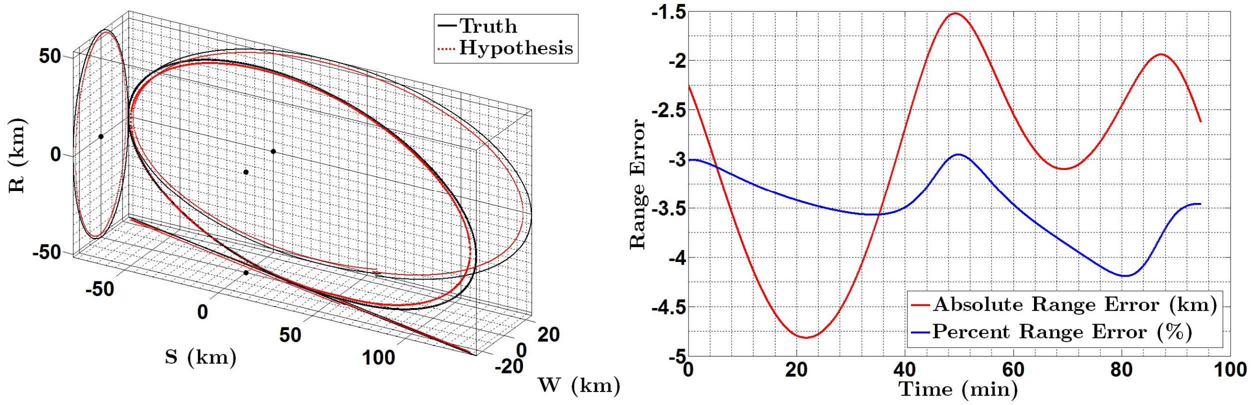
Figure 38: Hypothesis Accuracy for Case 08A. The trajectories are plotted in (a) with their projections in all three directions to gain perspective of the in-plane and out-of-plane motion. The black curves are the truth and the dashed red curves are the hypothesis. The black dots indicate the location of the Surveyor for the actual and projected trajectories. The range error is plotted in (b) as an absolute error (red curve) and as a percentage of the truth (blue curve).

LEO, High Differential Argument of Perigee, EGM96 20 × 20 (Case 08B)

Table 7: Case 08B: Orbital Elements for Truth and Hypothesis

COEs	Surveyor	Target Truth	Hypothesis	Hypothesis Error
a (km)	6789.085640626	6790.085640626	6790.101087731	0.015447105040
e (nd)	0.001601186396	0.006601186396	0.006343694000	-0.000257492396
i (deg)	51.75241867994	51.95241867994	51.94014533462	-0.012273345318
Ω (deg)	72.49304954653	72.59304954653	72.56994266086	-0.023106885674
ω (deg)	60.69588076088	191.0958807608	190.6958807608	-0.400000000000
ν (deg)	150.0000000000	20.00000000000	20.40125093291	0.401250932917

The hypothesis is plotted in Figure 39(a) with the truth. The range error shown in Figure 39(b) is higher than the low differential argument of perigee case and higher than the two-body version of Case 08. As before, the large differential in argument of perigee aggravates eccentricity since the $\cos\theta \approx 0$ assumption from Equation (5.14) is better for low differential argument of perigee near the relative apsis. The coarse estimation of argument of perigee is as good as can be expected with 10-degree precision. The true anomaly approximation absorbed the along-track shift error from the argument of perigee in order to match the skewness factor in the admissible region. The osculation corrections to the eccentricity, inclination and RAAN depend on the accuracy of the argument of perigee, so the fractional error in argument of perigee is causing some distortion to the hypothesis. Another source of error for the inclination and RAAN is due to a less accurate auxiliary trajectory since its dimensions constrain ρ_1 and ρ_2.



(a) Trajectories & Projections **(b) Range Error of Hypothesis**

Figure 39: Hypothesis Accuracy for Case 14B. The trajectories are plotted in (a) with their projections in all three directions to gain perspective of the in-plane and out-of-plane motion. The black curves are the truth and the dashed red curves are the hypothesis. The black dots indicate the location of the Surveyor for the actual and projected trajectories. The range error is plotted in (b) as an absolute error (red curve) and as a percentage of the truth (blue curve).

Verification Insight

A single iteration demonstrates how the absolute orbital elements can be used to reasonably describe the relative motion to within hundreds of meters in some cases and several kilometers in others. One way to improve this is to iterate on the argument of perigee and the semi-major axis. Once the argument of perigee has been refined, select $\hat{\boldsymbol{\rho}}_1$ to point roughly toward the Target's perigee or apogee so that the eccentricity is more accurate. Presently, $\hat{\boldsymbol{\rho}}_1$ points to the upper relative apsis which is roughly toward the Target's perigee or apogee for low differential argument of perigee. We have not made this change, but if the author had more time, this would be the first change in the queue! We will see shortly that these single-iteration IOD results are acceptable for initializing the nonlinear least squares filter.

5.4 Range Mapping

Once the COEs are found for the set of hypotheses in the admissible region, the ephemerides are synchronously propagated forward with identical time steps using a geopotential model that matches the truth data. Two options are available for parsing the results to find the scale of the relative trajectory. The first option uses angle-to-range maps ($\alpha|_{\dot{\alpha}} \rightarrow \rho$) introduced in Section 4.2. The second option uses time differentials of eclipse events to map range ($\Delta t|_{\text{eclipse}} \rightarrow \rho$), but it is merely complimentary to the angle-to-range maps since the time differentials may not be large enough to resolve range in all cases or there simply may not be an eclipse.

5.4.1 Angle-to-Range Maps

Reduce the numerically integrated data to ρ, α and $\dot{\alpha}$ using the procedures in Chapter 3. Identify local $\dot{\alpha}_{\text{min}}$ for each hypothesis and record α and ρ at these locations. Combine the ($\alpha|_{\dot{\alpha}_{\text{min}}}, \rho|_{\dot{\alpha}_{\text{min}}}$) pairs from each of the hypotheses to reveal a nearly linear relationship such as that shown in Figure 40. This particular case has two local minima per orbit and has a skewness factor of $d = 0.376$ which means the relative trajectory skews right and has better range resolution in the $+s$-direction. The combination of local minima in the $+s$-direction ($\alpha = 90$ deg) forms one curve, whereas the local minima in the $-s$-direction ($\alpha = -90$ deg) forms another curve. Recall the range resolution metric is just the slope of these curves! While both curves can be used to approximate range, the one with the larger $\Delta\alpha$ spread (typically, the one with the global $\dot{\alpha}_{\text{min}}$) should be the first choice since it has better resolvability.

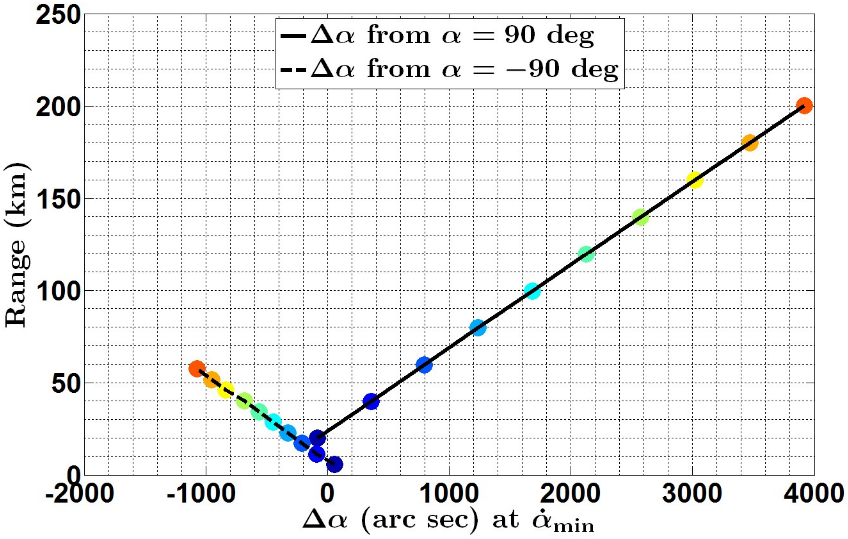


Figure 40: Example Angle-to-Range Map. The local $\dot{\alpha}$ minima from several hypotheses are used to construct an alpha-to-range map. The colored dots represent $\Delta\alpha$ at the local minima for each of the hypotheses. The solid black line passes through the combination of minima near $\alpha = 90$ deg and the dashed black line passes through the minima near $\alpha = -90$ deg.

Before mapping the range, perform a quick check for hypothesis outliers with the Theil-Sen estimator. This is accomplished by first finding the slope of all possible combinations of $(\alpha|_{\dot{\alpha}_{\text{min}}}, \rho|_{\dot{\alpha}_{\text{min}}})$ pairs

$$m_{ij} = \frac{\left(\rho|_{\dot{\alpha}_{\text{min}\,j}} - \rho|_{\dot{\alpha}_{\text{min}\,i}}\right)}{\left(\alpha|_{\dot{\alpha}_{\text{min}\,j}} - \alpha|_{\dot{\alpha}_{\text{min}\,i}}\right)} \tag{5.40}$$

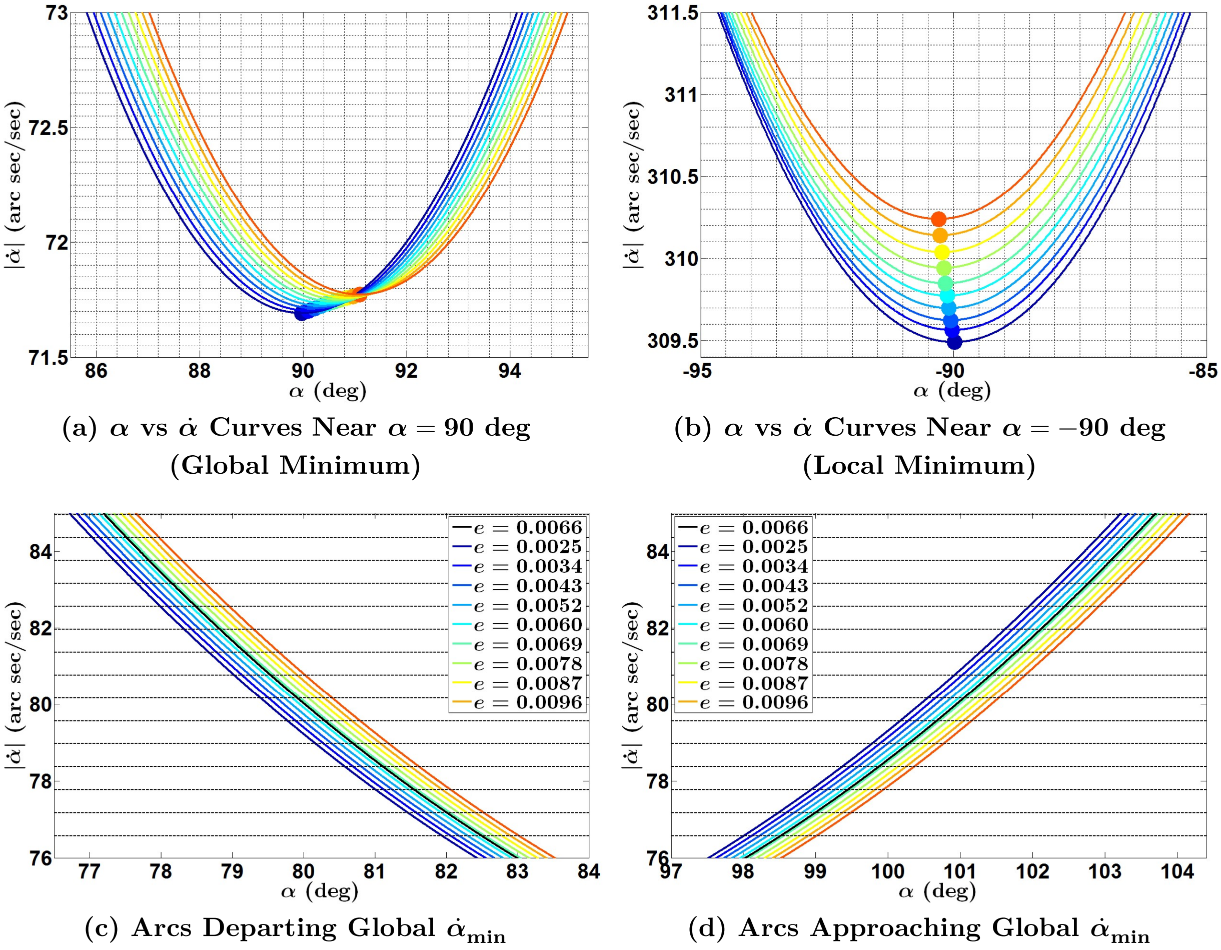
The median of all slope combinations is used to fit a linear curve to the data, which is more robust to outliers than ordinary least squares. Any hypotheses with $\rho|_{\dot{\alpha}_{\text{min}}}$ deviating more than 1σ from the Theil-Sen estimate is rejected. The remaining hypotheses are used to approximate range.

Since the minimum angular rate for NMC trajectories is always at $\pm 90 + \Delta\alpha$ deg, our notation simply uses $\Delta\alpha$ in seconds of arc to represent the minima. A quadratic polynomial is fit through the remaining admissible ($\Delta\alpha|_{\dot{\alpha}_{\min}}$, $\rho|_{\dot{\alpha}_{\min}}$) pairs to give range as a function of $\Delta\alpha$. The coefficients in Equation (5.41) are found using MATLAB's `polyfit` function

$$\rho|_{\dot{\alpha}_{\min}} = c_2\Delta\alpha^2|_{\dot{\alpha}_{\min}} + c_1\Delta\alpha|_{\dot{\alpha}_{\min}} + c_0 \tag{5.41}$$

The second order coefficient is small, but the result is slightly better than a linear map. The observed $\Delta\alpha|_{\dot{\alpha}_{\min}}$ can by substituted into Equation (5.41) to approximate $\rho|_{\dot{\alpha}_{\min}}$.

The location of the minimum angular rate has been discussed heavily for mapping range, but it is merely a starting point. Notice in Figure 41 that the α vs $\dot{\alpha}$ curves near $\dot{\alpha}_{\min}$ also have an ordered structure. These particular curves are associated with the example angle-to-range maps above, so the global minima near $\alpha = 90$ deg is the most skewed, providing the best resolvable structure in the arcs leading to and from $\dot{\alpha}_{\min}$. We choose $\dot{\alpha}$ waypoints along "straight" parts of these arcs to extract $(\alpha|_{\dot{\alpha}_{\mathrm{wp}}}, \rho|_{\dot{\alpha}_{\mathrm{wp}}})$ pairs. At each waypoint, fit a quadratic polynomial through the hypothesis pairs for the range map, $\alpha|_{\dot{\alpha}_{\mathrm{wp}}} \to \rho|_{\dot{\alpha}_{\mathrm{wp}}}$. Since the range varies at each of the waypoints, we need a common element to tie them all together in a least squares sense. The COEs in the admissible region scale almost linearly with range, but rather than mapping to each of them at each waypoint, we only map to eccentricity. Fit a quadratic polynomial through each of the $(\rho|_{\dot{\alpha}_{\mathrm{wp}}}, e_0)$ pairs for the eccentricity map, $\rho|_{\dot{\alpha}_{\mathrm{wp}}} \to e_0$. Finally, average the waypoint values for e_0.



(a) α vs $\dot{\alpha}$ Curves Near $\alpha = 90$ deg (Global Minimum)

(b) α vs $\dot{\alpha}$ Curves Near $\alpha = -90$ deg (Local Minimum)

(c) Arcs Departing Global $\dot{\alpha}_{\text{min}}$

(d) Arcs Approaching Global $\dot{\alpha}_{\text{min}}$

Figure 41: Characteristics of α vs $\dot{\alpha}$ Curves Near the Minima. The α vs $\dot{\alpha}$ curves of admissible hypotheses exhibit an ordered structure near the local minima. $\dot{\alpha}$ waypoints along the arcs leading to and from the global minima can be used for range maps as in (c) and (d) where waypoints are indicated by dashed horizontal lines. The true arc is the solid black line in (c) and (d). The eccentricities are listed in the legend.

The final step is to map the epoch eccentricity approximation to the other orbital elements. Fit a quadratic polynomial to each element and the eccentricity of the form

$$\text{COE}_0(e) = b_2 e^2 + b_1 \text{e} + b_0 \tag{5.42}$$

The epoch eccentricity is used to solve Equation (5.42) for each of the elements at epoch. A finishing touch is applied to the epoch true anomaly with the function `skew2nu` to make sure the skewness factor is matched in the final set.

Uncertainties are expected since this methodology is not exact, but the Target's range should be reasonably bounded and typically better than hundreds of meters. This knowledge can be used to warm-start the navigation filter near the truth.

5.4.2 Eclipse-to-Range Maps

Another unorthodox approach for determining the range of a Target within a multiple hypothesis framework is to use eclipse times. The differential time between both satellites becoming fully darkened upon entry into the Earth's shadow or fully illuminated upon exit from the Earth's shadow, can be simulated from multiple admissible orbit hypotheses and compared to the observed differential time. If the differential times are large enough across the set of hypotheses, threshold visual magnitudes can be used to approximate the scale of the Target's trajectory, or more likely, to prune improbable hypotheses with the map $\Delta t|_{\text{eclipse}} \rightarrow \rho|_{\text{eclipse}}$. This procedure can potentially be applied to any RPO trajectory as long as an admissible set of orbit hypotheses can be generated.

Eclipse Model

The shadow cast by the Earth includes three main regions: (1) a dark inner cone where the Sun's rays are completely occluded by the Earth called the umbra, (2) an outer hollow cone surrounding the umbra where a fraction of the Sun's light is blocked called the penumbra and (3) another partially eclipsed inner cone (annular eclipse) called the antumbra that extends from the umbra's vertex where the occulting Earth appears smaller than the Sun. The geometry is depicted in Figure 42; although, the antumbra is ignored since the umbra extends over 1.3 million km[36]—*well* beyond our region of interest.

Conical models of the umbra and penumbra are implemented for this feasibility study and the boundaries are binary. See Appendix C for analysis on the merits of a cylindrical model assumption at LEO and GEO. In reality, the photometric data will not be binary; the light curve will exhibit a gradual occlusion as the Target passes through the terminator into an eclipse. Operationally, a standard threshold visual magnitude for when the satellite enters or exits an eclipse is needed, but the principle of differential entry and exit times is the same for a simpler binary approach. Other lighting effects due to the Target's size, shape, orientation and material properties are also important considerations that have been neglected here. Determining these physical characteristics from photometry and angle data is an enormous challenge of its own [173].[37] The simulation to estimate these properties requires 3D spacecraft models with different facet materials and accurate lighting. The author has developed just such a simulation that ties together SketchUp and GMAT in Ubuntu, but the estimation routines are far beyond the scope of this research. Thus, we settle for binary light curve data in this simple feasibility study.

[36] See https://commons.wikimedia.org/wiki/File:Earth_umbral_cone_(partial).png

[37] The author supervised similar work by Andrew Dianetti (graduate student from the University at Buffalo) at the Air Force Research Laboratory's Space Vehicles Directorate in 2014.

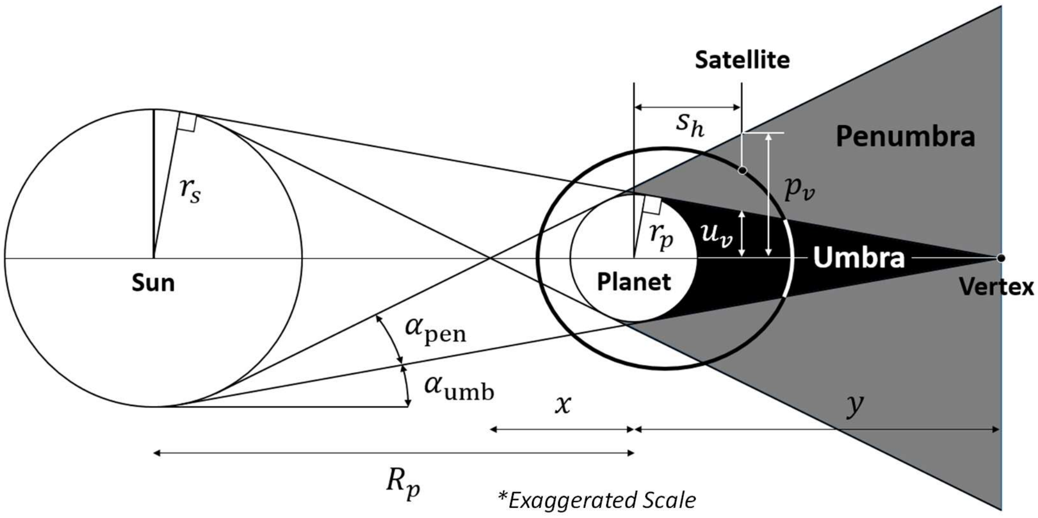


Figure 42: Earth Eclipse Geometry. The umbra is a conical region that is fully eclipsed by the planet, whereas the penumbra is a partially eclipsed region surrounding the umbral core.

At each time step in the numerical integration of our hypotheses, apply Vallado's geometrical `SHADOW` algorithm [118], which is a set of conditional statements to determine whether the satellite is in the penumbra or umbra.

First, check for a negative value in the Sun's position vector and the satellite's position vector

$$^{i}\boldsymbol{r}_{\odot} \cdot {}^{i}\mathbf{r}_{\text{sat}} < \mathbf{0} \tag{5.43}$$

If the dot product is positive, the satellite is being radiated by the Sun. If it is negative, there is a possibility of an eclipse, so determine the angle between the negative Sun vector and the satellite's position vector

$$\varsigma = \frac{-^{i}\boldsymbol{r}_{\odot} \cdot {}^{i}\mathbf{r}_{\text{sat}}}{|^{i}\boldsymbol{r}_{\odot}||^{i}\mathbf{r}_{\text{sat}}|} \tag{5.44}$$

Calculate the satellite's horizontal distance along the Earth-Sun line and the vertical distance perpendicular to the Earth-Sun line

$$s_h = r\cos\varsigma \tag{5.45}$$

$$s_v = r\sin\varsigma \tag{5.46}$$

Find the Earth primary/secondary point, x, and the Earth-vertex distance, y, shown in Figure 42 along the Earth-Sun line

$$x = \frac{R_\oplus}{\sin(\alpha_{\text{pen}})} \tag{5.47}$$

$$y = \frac{R_\oplus}{\sin(\alpha_{\text{umb}})} \tag{5.48}$$

where $R_\oplus$ is the Earth radius and the penumbra and umbra angles are $\alpha_{\text{pen}} = 0.269007205°$ and $\alpha_{\text{umb}} = 0.264121687°$, respectively.

Substitute the above quantities to calculate the penumbra and umbra distances

$$p_v = \tan(\alpha_{\text{pen}})\,(x + s_h) \tag{5.49}$$

$$u_v = \tan(\alpha_{\text{umb}})\,(y - s_h) \tag{5.50}$$

If $s_v \leq p_v$, then the satellite is at least in the penumbra, but if $s_v \leq u_v$, then the satellite is in the umbra. Due to discrete time steps in the numerical integration, the entry and exit points for the penumbra and umbra have uncertainty no greater than the sample rate in this binary model.

Generating Maps

Upon generating and propagating a cluster of admissible hypotheses, the umbra entry and exit times for the Surveyor and all Target hypotheses are recorded and differenced to generate the eclipse Δt_{entry} and Δt_{exit}.

$$\Delta t_{\text{entry}} = t_s|_{\text{umb}} - t_t|_{\text{umb}} \tag{5.51}$$

$$\Delta t_{\text{exit}} = t_s|_{\text{pen}} - t_t|_{\text{pen}} \tag{5.52}$$

Since the range for each of the admissible hypotheses are known at $t_t|_{\text{umb}}$ and $t_t|_{\text{pen}}$, a range map can be generated as in Figure 43. Depending on the amount of uncertainty associated with the observed Δt, the eclipse map can either be used to approximate the range or prune hypotheses from the admissible region, thereby bounding the range.

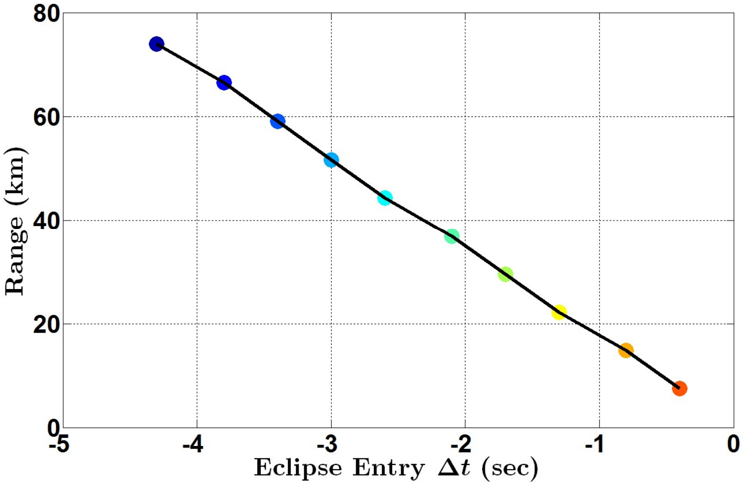


Figure 43: Example Eclipse-to-Range Map from Notional Admissible Region. The range for each of 10 admissible hypotheses are plotted with Δt from an eclipse event. A curve fit can be used to map range with the observed Δt. This particular map is from Case 06C.

Eclipse maps are not guaranteed to be available since they depend on the orbit geometry with respect to the Sun. Throughout the year the Sun appears to drift 23 degrees above and below the Earth's equatorial plane since the Earth's equator is not aligned with the ecliptic plane. This becomes particularly important for GEO since the amount and type of eclipsing can vary dramatically throughout the year. GEO satellites only experience Earth eclipse during the March or September equinox when the sun is aligned with the equatorial plane.

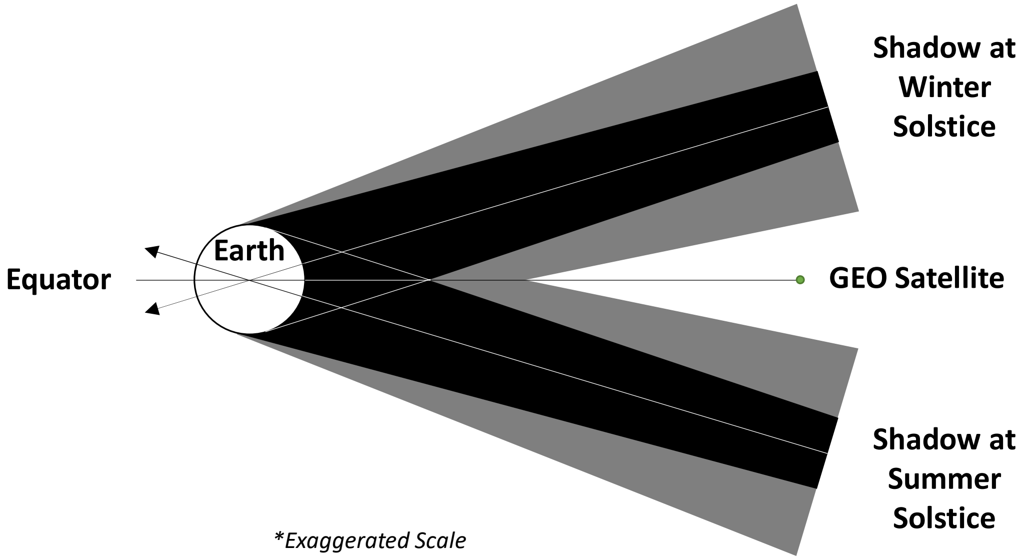


Figure 44: GEO Satellites at Solstice. All geostationary satellites experience constant sunlight except during the equinox periods.

5.5 Batch Filter Design

To improve upon the IOD solution in a navigation filter, a technique is needed that retains the nonlinear dynamics in the motion model or as much as possible. We also know from D'Amico's work at DLR that the inclusion of J_2 dynamics improves range observability with nonlinear least squares (NLS) batch processing. J_2 is the most dominant term among the zonal harmonics, indeed the entire gravitational field, contributing the equivalent of about 10^{-3} of the Newtonian point mass coefficient. All other zonal, sectoral and tesseral coefficients combined are two orders of magnitude smaller than J_2, so is there any good reason to include the smaller coefficients given the added complexity? There may be some value at GEO.

The precession of orbits at GEO (North-South drift) are caused in part by J_2, but mostly by the tug of the Moon and the Sun, which is corrected at a cost of ~50 m/s each year.

East-West drift is caused by the ellipticity of the Earth or the high landmass potentials over the Americas and Eurasia and the low potentials over the Atlantic and the Pacific. This longitudinal asymmetry is modelled by the sectoral harmonics and costs satellites approximately 2 m/s each year to counteract the East-West drift. Since one full NMC trajectory takes approximately 1/365 of a year at GEO, then the East-West drift is having the effect of ~5.5 mm/s per orbit. If a case can be made for micro-thrusters aiding observability, then maybe a higher fidelity geopotential model is warranted, particularly at GEO.

Our dynamics model for the filter uses a non-Keplerian reference orbit with an expansion of the geopotential to any order and degree. The state transition matrix (STM) is generated from the equations of variation about a reference orbit so that the linearization of the dynamics at every time step closely tracks the reference orbit. In this way, the STM only describes the local motion and not the global motion. The estimated trajectory is a combination of the global reference orbit dynamics and the local dynamics from small offsets described mathematically as

$$ {}^{e}\widetilde{\boldsymbol{X}}(t) = {}^{e}\boldsymbol{X}_{\text{ref}}(t) + \Phi(t, t_0)\, {}^{e}\delta\widetilde{\boldsymbol{X}}(t_0) \tag{5.53} $$

where ${}^{e}\widetilde{\boldsymbol{X}}$ is the estimated state vector in the ECEF frame, ${}^{e}\boldsymbol{X}_{\text{ref}}$ is the reference orbit in the ECEF frame, $\Phi(t, t_0)$ is the STM, and ${}^{e}\delta\widetilde{\boldsymbol{X}}(t_0)$ is the estimated correction to the reference trajectory at epoch. This technique for LOS navigation is only possible with a sensible IOD solution so that the corrections and, therefore, the linearization errors are small as shown in Figure 45.

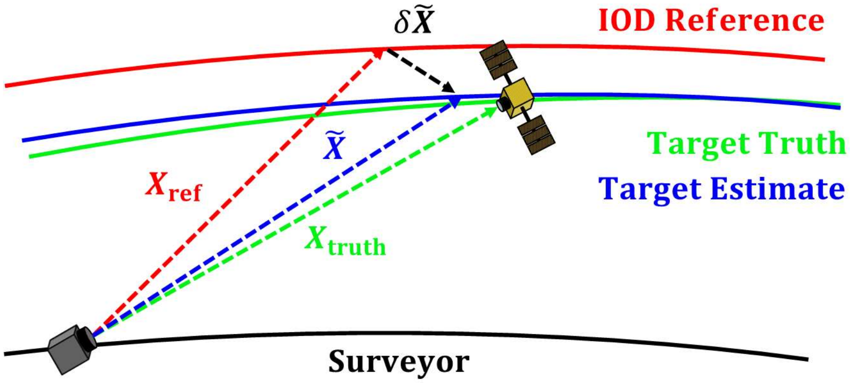


Figure 45: Global and Local Motion Representation. The reference orbit models the global nonlinear motion in the expanded geopotential. The local motion dynamics are formulated from the global motion state to estimate the correction, δX, to the reference orbit.

The reference orbit can be numerically integrated or modelled as a KAM torus in the full geopotential. One major advantage of the torus over numerical integration is that the satellite's dynamics are precisely time synchronized with observations regardless of sampling rate or observation gaps. While our simulated observations can be perfectly synchronized with a numerical integration, this is likely not the case operationally. A fraction of a second difference between the reference orbit propagation and a true observation corrupts the residuals. This *never* happens with a torus since its only input is the precise observation time.

5.5.1 Nonlinear Weighted Least Squares

Differential correction through batch nonlinear least squares is a centerpiece of modern astrodynamics and has been routinely applied to LOS navigation. The difference between this work and others is the use of a reference orbit and the equations of variations in the expanded geopotential for the dynamics model. This subsection describes the custom implementation of the NLS algorithm to use a reference orbit; the dynamics model and measurement model is developed in later subsections.

The reference orbit state vector in the Earth-centered Earth-fixed frame is specified as

$$^{\mathrm{e}}\boldsymbol{X}_{\mathrm{ref}} = [x \quad y \quad z \quad \dot{x} \quad \dot{y} \quad \dot{z}]^T \tag{5.54}$$

$$\dot{x} = {}^{\mathrm{e}}\dot{x} - \omega_{\oplus} y \tag{5.55}$$

$$\dot{y} = {}^{\mathrm{e}}\dot{y} + \omega_{\oplus} x \tag{5.56}$$

$$\dot{z} = {}^{\mathrm{e}}\dot{z} \tag{5.57}$$

where the velocity components are the inertial velocities resolved in the ECEF frame and the rotation rate of the Earth is given by $\omega_{\oplus}$.

An observation of the Target at time, t, is specified in Hill's frame by the measurement vector

$$^{\mathrm{h}}\boldsymbol{M}_{\mathrm{obs}}(t) = \begin{bmatrix} \alpha \\ \beta \\ \rho \end{bmatrix} \tag{5.58}$$

where the range, ρ, is purposefully set to zero. For independent measurements, the observation variance-covariance matrix or weight matrix is

$$\boldsymbol{Q} = \begin{bmatrix} \sigma_\alpha^2 & 0 & 0 \\ 0 & \sigma_\beta^2 & 0 \\ 0 & 0 & \sigma_\rho^2 \end{bmatrix} \tag{5.59}$$

where σ^2 is the variance or expectation of the squared deviation of a measurement variable from its mean, $E[(M-\mu)^2]$. Based upon a survey of state-of-the-art star trackers, we specify $\sigma_\alpha = 10$ arc sec and $\sigma_\beta = 10$ arc sec. The range uncertainty is arbitrarily set to $\sigma_\rho = 1 \times 10^9$ km to send a strong vote of no confidence in the range measurement since it does not exist!

For each observation, $^{h}\boldsymbol{M}_{\mathrm{obs}}(t)$, extract $^{\mathrm{e}}\boldsymbol{X}_{\mathrm{ref}}(t)$ from the KAM torus or numerically integrated reference orbit at the observation time, t, and obtain the state transition matrix

(STM), $\mathbf{\Phi}(t, t_0)$. In our case, the STM is only being used to propagate small deviations near the reference orbit and *not* the actual orbit itself. It is the torus' job to propagate the bulk of the satellite's motion. Two versions of the $\boldsymbol{A}$ matrix are formulated: a numerical version from a geopotential expansion to any degree and order and an analytical, osculating two-body version (osculating with the reference orbit). These are developed in Section 5.5.3.

The correction to the reference orbit at the time of the observation is brought forward in time with

$$^{\mathrm{e}}\delta\widetilde{\boldsymbol{X}}_{\mathrm{ref}}(t) = \mathbf{\Phi}(t, t_0)\,^{\mathrm{e}}\delta\widetilde{\boldsymbol{X}}_{\mathrm{ref}}(t_0) \tag{5.60}$$

And the corrected reference orbit is

$$^{\mathrm{e}}\boldsymbol{X}'_{\mathrm{ref}}(t) = {}^{\mathrm{e}}\boldsymbol{X}_{\mathrm{ref}}(t) + {}^{\mathrm{e}}\delta\widetilde{\boldsymbol{X}}_{\mathrm{ref}}(t) \tag{5.61}$$

Subtract the earth's rotation rate contribution from the velocity to get the ECEF velocity and transform the corrected reference orbit to Hill's frame

$$^{\mathrm{h}}\boldsymbol{X}'_{\mathrm{ref}}(t) = [R^{\mathrm{hi}}][R^{\mathrm{ie}}]\,^{\mathrm{e}}\boldsymbol{X}'_{\mathrm{ref}}(t) \tag{5.62}$$

Use the corrected reference orbit in Hill's frame to generate the expected angle and range observations

$$^{\mathrm{h}}\boldsymbol{M}_{\mathrm{ref}}(t) = \begin{bmatrix} \alpha_{\mathrm{ref}} \\ \beta_{\mathrm{ref}} \\ \rho_{\mathrm{ref}} \end{bmatrix} \tag{5.63}$$

Obtain the residual vector, $r(t)$, and reject true outliers, generally when the angle residuals exceed $3\sigma_{\alpha,\beta}$.

$$r(t) = {}^{\mathrm{h}}\boldsymbol{M}_{\mathrm{obs}}(t) - {}^{\mathrm{h}}\boldsymbol{M}_{\mathrm{ref}}(t) \tag{5.64}$$

Next, calculate the measurement sensitivity matrix or the Jacobian of the measurement model for the data point. The Jacobian is derived in Section 5.5.2. In this case, $\boldsymbol{H}$ is a 3×6 matrix

$$\boldsymbol{H}(t) = \frac{\partial \boldsymbol{M}(t)}{\partial \boldsymbol{X}(t)} \tag{5.65}$$

Form the 3×6 observation matrix

$$\boldsymbol{T}(t) = \boldsymbol{H}(t)\boldsymbol{\Phi}(t, t_0) \tag{5.66}$$

Add new terms to the running sum of the inverted 6×6 covariance matrix

$$\boldsymbol{P}^{-1} = \sum \boldsymbol{T}^T \boldsymbol{Q}^{-1} \boldsymbol{T} \tag{5.67}$$

Also, add new terms to the 6×1 vector sum

$$\sum \boldsymbol{T}^T \boldsymbol{Q}^{-1} \boldsymbol{r} \tag{5.68}$$

When all data has been processed, calculate the covariance of the correction and the state correction vector at epoch

$$\boldsymbol{P}_{\delta \widetilde{\boldsymbol{X}}} = \left(\sum \boldsymbol{T}^T \boldsymbol{Q}^{-1} \boldsymbol{T}\right)^{-1} \tag{5.69}$$

$$\delta \widetilde{\boldsymbol{X}}_i = \boldsymbol{P}_{\delta \boldsymbol{X}} \sum \boldsymbol{T}^T \boldsymbol{Q}^{-1} \boldsymbol{r} \tag{5.70}$$

Update the reference orbit correction by summing the state correction vectors from each iteration, i. Note that $\delta \widetilde{\boldsymbol{X}}_i = \mathbf{0}_{6 \times 1}$ at the onset of the iterative process

$$^{\mathrm{e}}\delta \widetilde{\boldsymbol{X}}_{\mathrm{ref}}(t_0) = \sum \delta \widetilde{\boldsymbol{X}}_i \tag{5.71}$$

Determine if the process has converged. If not, begin again with $^{\mathrm{e}}\delta \widetilde{\boldsymbol{X}}_{\mathrm{ref}}(t_0)$, else claim victory. We use the convergence criteria

$$\sqrt{\delta x_i^2 + \delta y_i^2 + \delta z_i^2} < 0.1 \text{ m} \tag{5.72}$$

5.5.2 Jacobian of the Measurement Model

The usual LOS measurement model is the analytical expression for the measurements in terms of the Cartesian coordinates in Hill's frame as posed in Equation (3.21) and (3.22). The Jacobian describes how the measurements change with a small change in the reference state. Since our reference state is in the ECEF frame and not Hill's frame, our Jacobian is slightly more complicated than normal since it involves rotation matrices. First, write the relative position in Hill's frame in terms of ECEF coordinates (x, y, z)

$$ {}^{h}\begin{bmatrix} \mathrm{r}_t \\ s_t \\ w_t \end{bmatrix} = \begin{bmatrix} R^{hi}_{13} z - {}^{h}\mathrm{r}_s \ + \ R^{hi}_{11}(x\cos\theta \ - \ y\sin\theta) + R^{hi}_{12}(y\cos\theta + x\sin\theta) \\ R^{hi}_{23} z + R^{hi}_{21}(x\cos\theta \ - \ y\sin\theta) + R^{hi}_{22}(y\cos\theta + x\sin\theta) \\ R^{hi}_{33} z + R^{hi}_{31}(x\cos\theta \ - \ y\sin\theta) + R^{hi}_{32}(y\cos\theta + x\sin\theta) \end{bmatrix} \tag{5.73} $$

where $\boldsymbol{R}^{hi}$ is the rotation matrix from ECI to Hill's frame as previously derived in Equation (3.15), ${}^{h}\mathrm{r}_s$ is the Surveyor's radial distance, and θ is the Greenwich mean sidereal time.

The partials of the Jacobian in Equation (5.74) can now be developed individually. Half of the matrix is trivial since the measurements are not a function of the reference velocity.

$$ \frac{\partial \boldsymbol{M}(t)}{\partial \boldsymbol{X}(t)} = \begin{bmatrix} \frac{\partial\alpha}{\partial x} & \frac{\partial\alpha}{\partial y} & \frac{\partial\alpha}{\partial z} & \frac{\partial\alpha}{\partial \dot{x}} & \frac{\partial\alpha}{\partial \dot{y}} & \frac{\partial\alpha}{\partial \dot{z}} \\ \frac{\partial\beta}{\partial x} & \frac{\partial\beta}{\partial y} & \frac{\partial\beta}{\partial z} & \frac{\partial\beta}{\partial \dot{x}} & \frac{\partial\beta}{\partial \dot{y}} & \frac{\partial\beta}{\partial \dot{z}} \\ \frac{\partial\rho}{\partial x} & \frac{\partial\rho}{\partial y} & \frac{\partial\rho}{\partial z} & \frac{\partial\rho}{\partial \dot{x}} & \frac{\partial\rho}{\partial \dot{y}} & \frac{\partial\rho}{\partial \dot{z}} \end{bmatrix} \tag{5.74} $$

$$ \frac{\partial \boldsymbol{M}(t)}{\partial \boldsymbol{X}(t)} = \begin{bmatrix} \frac{\partial\alpha}{\partial x} & \frac{\partial\alpha}{\partial y} & \frac{\partial\alpha}{\partial z} & 0 & 0 & 0 \\ \frac{\partial\beta}{\partial x} & \frac{\partial\beta}{\partial y} & \frac{\partial\beta}{\partial z} & 0 & 0 & 0 \\ \frac{\partial\rho}{\partial x} & \frac{\partial\rho}{\partial y} & \frac{\partial\rho}{\partial z} & 0 & 0 & 0 \end{bmatrix} \tag{5.75} $$

$$\frac{\partial\alpha}{\partial x} = \frac{\big(a(R_{21}\cos\theta + R_{22}\sin\theta) - b(R_{11}\cos\theta + R_{12}\sin\theta)\big)}{b^2 + a^2} \tag{5.76}$$

$$\frac{\partial\alpha}{\partial y} = \frac{\big(a(R_{22}\cos\theta + R_{21}\sin\theta) - b(R_{12}\cos\theta + R_{11}\sin\theta)\big)}{b^2 + a^2} \tag{5.77}$$

$$\frac{\partial\alpha}{\partial z} = \frac{(aR_{23} - bR_{13})}{b^2 + a^2} \tag{5.78}$$

where

$$a = R_{13}z - {}^{h}r_s + R_{11}(x\cos\theta - y\sin\theta) + R_{12}(y\cos\theta + x\sin\theta)$$

$$b = R_{23}z + R_{21}(x\cos\theta - y\sin\theta) + R_{22}(y\cos\theta + x\sin\theta)$$

$$\frac{\partial\beta}{\partial x} = \frac{\sqrt{a^2+b^2}(R_{31}\cos\theta + R_{32}\sin\theta) - d}{a^2 + b^2 + c^2} \tag{5.79}$$

$$\frac{\partial\beta}{\partial y} = \frac{\sqrt{a^2+b^2}(R_{32}\cos\theta + R_{31}\sin\theta) - e}{a^2 + b^2 + c^2} \tag{5.80}$$

$$\frac{\partial\beta}{\partial z} = \frac{\sqrt{a^2+b^2}R_{33} - \frac{acR_{13} + bcR_{23}}{\sqrt{a^2+b^2}}}{a^2 + b^2 + c^2} \tag{5.81}$$

where

$$c = R_{33}z + R_{31}(x\cos\theta - y\sin\theta) + R_{32}(y\cos\theta + x\sin\theta)$$

$$d = \frac{ac(R_{11}\cos\theta + R_{12}\sin\theta) + bc(R_{21}\cos\theta + R_{22}\sin\theta)}{\sqrt{a^2+b^2}}$$

$$e = \frac{ac(R_{12}\cos\theta - R_{11}\sin\theta) + bc(R_{22}\cos\theta - R_{21}\sin\theta)}{\sqrt{a^2+b^2}}$$

$$\frac{\partial\rho}{\partial x} = \frac{a(R_{11}\cos\theta + R_{12}\sin\theta) + b(R_{21}\cos\theta + R_{22}\sin\theta) + c(R_{31}\cos\theta + R_{32}\sin\theta)}{\sqrt{a^2+b^2+c^2}} \tag{5.82}$$

$$\frac{\partial\rho}{\partial y} = \frac{a(R_{12}\cos\theta - R_{11}\sin\theta) + b(R_{22}\cos\theta - R_{21}\sin\theta) + c(R_{32}\cos\theta - R_{31}\sin\theta)}{\sqrt{a^2+b^2+c^2}} \tag{5.83}$$

$$\frac{\partial\rho}{\partial z} = \frac{aR_{13} + bR_{23} + cR_{33}}{\sqrt{a^2+b^2+c^2}} \tag{5.84}$$

5.5.3 State Transition Matrix

Using Hamiltonian dynamics, formulate the satellite's equations of motion in an Earth-centered Earth-fixed (ECEF) frame. This reduces the satellite's motion to one constant of integration. We will note shortly that the Hamiltonian is independent of time in the stationary gravity field.

The generalized coordinates for a satellite in a rotating reference frame and the inertial velocity resolved along the axes of the rotating frame are

$$\begin{bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \dot{x} - \omega_{\oplus} y \\ \dot{y} + \omega_{\oplus} x \\ \dot{z} \end{bmatrix} \tag{5.85}$$

where $\omega_{\oplus}$ is the Earth rotation rate. In a rotating reference frame, the kinetic energy of the satellite per unit mass is

$$T = \frac{1}{2}((\dot{x} - \omega_{\oplus} y)^2 + (\dot{y} + \omega_{\oplus} x)^2 + (\dot{z})^2) \tag{5.86}$$

The Lagrangian is defined as

$$\mathcal{L} = T(q, \dot{q}) - V(q) \tag{5.87}$$

where V is potential energy; for our application, it is the truncated expansion of the geopotential in spherical harmonics

$$V(r, \psi, \lambda) = -\frac{\mu}{\mathrm{r}} \sum_{n=n_{\min}}^{n_{\max}} \left(\frac{R_{\oplus}}{\mathrm{r}}\right)^{n} \sum_{m=0}^{n} (\bar{C}_{n,m} \cos m\lambda + \bar{S}_{n,m} \sin m\lambda) \bar{P}_{n,m}(\sin \psi) \tag{5.88}$$

where $R_{\oplus}$ is Earth's equatorial radius, μ is Earth's gravitational parameter, n and m are the degree and order of the geopotential expansion, respectively, $\bar{P}_{n,m}$ are the fully normalized Legendre polynomials, $\bar{C}_{n,m}$ and $\bar{S}_{n,m}$ are full normalized dimensionless coefficients from the gravity model (we use NASA EGM96), r is the geocentric radius of the Target defined as $\mathrm{r} = \sqrt{x^2 + y^2 + z^2}$, ψ is the geocentric latitude[38] defined as $\psi = \sin^{-1}(z/r)$ and λ is east longitude defined as $\lambda = \tan^{-1}(y/x)$ where the x-axis is referenced to Greenwich in the ECEF frame.

For two-body motion, the gravitational potential is just

$$V(\mathrm{r}) = -\frac{\mu}{r} \tag{5.89}$$

The generalized momenta are[39]

$$p_i = \frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}_i} \tag{5.90}$$

$$\boldsymbol{p} = \begin{bmatrix} \dot{x} - \omega y \\ \dot{y} + \omega x \\ \dot{z} \end{bmatrix} \tag{5.91}$$

The Hamiltonian function with the full geopotential expansion is

$$\mathcal{H} = \textstyle\sum_i p_i \dot{q}_i - \mathcal{L} \tag{5.92}$$

$$\begin{aligned} \mathcal{H} = \frac{1}{2}(p_x^2 + p_y^2 + p_z^2) + \omega_\oplus(y p_x - x p_y) \\ - \frac{\mu}{\mathrm{r}} \sum_{n=n_{\min}}^{n_{\max}} \left(\frac{R_\oplus}{\mathrm{r}}\right)^n \sum_{m=0}^{n} \big(\bar{C}_{n,m} \cos m\lambda + \bar{S}_{n,m} \sin m\lambda\big) \bar{P}_{n,m}(\sin\psi) \end{aligned} \tag{5.93}$$

The two-body Hamiltonian function is

$$\mathcal{H} = \frac{1}{2}(p_x^2 + p_y^2 + p_z^2) + \omega_\oplus(y p_x - x p_y) - \frac{\mu}{\mathrm{r}} \tag{5.94}$$

Notice that the Hamiltonian is independent of time and is therefore a constant of motion.

The equations of motion (EOM) or Hamilton's equations are found from the partials

[38] Can also be replaced with the colatitude $\theta = \cos^{-1}(z/r) = \frac{\pi}{2} - \psi$

[39] The momenta are not with respect to the rotating frame as the generalized coordinates are. The momenta *are* the satellite's inertial velocity components. This is the case for all rotating reference frames.

$$\frac{dq_i}{dt} = \frac{\partial \mathcal{H}(q,p)}{\partial p_i} \tag{5.95}$$

$$\frac{dp_i}{dt} = -\frac{\partial \mathcal{H}(q,p)}{\partial q_i} \tag{5.96}$$

$$\dot{\boldsymbol{X}} = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{p}} \end{bmatrix} = \begin{bmatrix} p_x + \omega_\oplus q_y \\ p_y - \omega_\oplus q_x \\ p_z \\ \omega p_y - \dfrac{\partial V(q)}{\partial q_x} \\ -\omega p_x - \dfrac{\partial V(q)}{\partial q_y} \\ -\dfrac{\partial V}{\partial q_z} \end{bmatrix} \tag{5.97}$$

The EOM can also be expressed more conveniently with a symplectic matrix

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{0}_3 & \boldsymbol{I}_3 \\ -\boldsymbol{I}_3 & \boldsymbol{0}_3 \end{bmatrix} \tag{5.98}$$

$$\dot{\boldsymbol{X}} = Z \frac{\partial \mathcal{H}}{\partial \boldsymbol{X}} \tag{5.99}$$

where $\boldsymbol{I}_3$ is a 3×3 identity matrix and $\boldsymbol{0}_3$ is a 3×3 matrix of zeros.

The time variant linear differential equations or the *equations of variation* that describe displacements to the reference orbit are

$$\delta\dot{\boldsymbol{X}} = \boldsymbol{A}(t)\delta\boldsymbol{X} \tag{5.100}$$

$$\boldsymbol{A} = \boldsymbol{Z} \frac{\partial^2 \mathcal{H}(q,p)}{\partial \boldsymbol{X}^2} \tag{5.101}$$

The STM is the matrix exponential function

$$\boldsymbol{\Phi}(t, t_0) = e^{\boldsymbol{A}t} = I + \boldsymbol{A}t + \boldsymbol{A}^2 \frac{t^2}{2!} + \cdots + \boldsymbol{A}^n \frac{t^n}{n!} \tag{5.102}$$

For a constant coefficient linear system, the state transition matrix has the form

$$\boldsymbol{\mathit{\Phi}}(t, 0) = \boldsymbol{F} e^{\mathcal{J}t} \boldsymbol{F}^{\boldsymbol{-1}} \tag{5.103}$$

where $\boldsymbol{F}$ is a constant matrix of eigenvectors of $\boldsymbol{A}$ and $\boldsymbol{\mathcal{J}}$ is (usually) a diagonal matrix of eigenvalues of A. However, we are *not* dealing with a constant coefficient linear system! The $\boldsymbol{A}$ matrix is constantly changing around the nonlinear reference orbit, so we elect to solve $e^{\boldsymbol{A}t}$ using Equation (5.102) to $\mathcal{O}(\boldsymbol{A}^6)$ at each observation for a small interval of time

$$\boldsymbol{\Phi}(t_{i-1}+dt, t_{i-1}) \approx e^{\boldsymbol{A}dt} = I + \boldsymbol{A}dt + \boldsymbol{A}^2\frac{dt^2}{2!} + \cdots + \boldsymbol{A}^6\frac{dt^6}{6!} \tag{5.104}$$

In order to propagate a small change in the reference orbit from epoch to an arbitrary time of observations, the STM must be accumulated such that

$$\boldsymbol{\Phi}(t_i, t_0) \approx \boldsymbol{\Phi}(t_{i-1}+dt, t_{i-1}) \prod_{i=2}^{n} \boldsymbol{\Phi}(t_{i-1}, t_{i-2}) \tag{5.105}$$

It is computationally efficient to accumulate the past STMs as a single matrix rather than storing them or recalculating them at each observation.

One can elect to use the $\boldsymbol{A}$ matrix from the equations of variation for the full geopotential expansion or the osculating two-body form. We describe it as osculating because the instantaneous two-body dynamics are calculated from the reference orbit position and velocity, which are osculating in the full geopotential.

Osculating Two-Body *A* Matrix

Start by writing the partials of the Hamiltonian to aid with taking the second-order partial derivatives in the Hessian

$$\frac{\partial \mathcal{H}}{\partial \boldsymbol{X}} = \begin{bmatrix} \frac{\partial \mathcal{H}}{\partial q_x} & \frac{\partial \mathcal{H}}{\partial q_y} & \frac{\partial \mathcal{H}}{\partial q_z} & \frac{\partial \mathcal{H}}{\partial p_x} & \frac{\partial \mathcal{H}}{\partial p_y} & \frac{\partial \mathcal{H}}{\partial p_z} \end{bmatrix}^T \tag{5.106}$$

$$\frac{\partial \mathcal{H}}{\partial \boldsymbol{X}} = \begin{bmatrix} -\omega_{\oplus} p_y + \dfrac{\partial V(q)}{\partial q_x} \\ \omega_{\oplus} p_x + \dfrac{\partial V(q)}{\partial q_y} \\ \dfrac{\partial V(q)}{\partial q_z} \\ p_x + \omega_{\oplus} q_y \\ p_y - \omega_{\oplus} q_x \\ p_z \end{bmatrix}$$

The Hessian in Leibniz notation is

$$\frac{\partial^2 \mathcal{H}}{\partial \boldsymbol{X}^2} = \begin{bmatrix} \dfrac{\partial^2 \mathcal{H}}{\partial q_x^2} & \dfrac{\partial^2 \mathcal{H}}{\partial q_x \partial q_y} & \dfrac{\partial^2 \mathcal{H}}{\partial q_x \partial q_z} & \dfrac{\partial^2 \mathcal{H}}{\partial q_x \partial p_x} & \dfrac{\partial^2 \mathcal{H}}{\partial q_x \partial p_y} & \dfrac{\partial^2 \mathcal{H}}{\partial q_x \partial p_z} \\ \dfrac{\partial^2 \mathcal{H}}{\partial q_y \partial q_x} & \dfrac{\partial^2 \mathcal{H}}{\partial q_y^2} & \dfrac{\partial^2 \mathcal{H}}{\partial q_y \partial q_z} & \dfrac{\partial^2 \mathcal{H}}{\partial q_y \partial p_x} & \dfrac{\partial^2 \mathcal{H}}{\partial q_y \partial p_y} & \dfrac{\partial^2 \mathcal{H}}{\partial q_y \partial p_z} \\ \dfrac{\partial^2 \mathcal{H}}{\partial q_z \partial q_x} & \dfrac{\partial^2 \mathcal{H}}{\partial q_z \partial q_y} & \dfrac{\partial^2 \mathcal{H}}{\partial q_z^2} & \dfrac{\partial^2 \mathcal{H}}{\partial q_z \partial p_x} & \dfrac{\partial^2 \mathcal{H}}{\partial q_z \partial p_y} & \dfrac{\partial^2 \mathcal{H}}{\partial q_z \partial p_z} \\ \dfrac{\partial^2 \mathcal{H}}{\partial p_x \partial q_x} & \dfrac{\partial^2 \mathcal{H}}{\partial p_x \partial q_y} & \dfrac{\partial^2 \mathcal{H}}{\partial p_x \partial q_z} & \dfrac{\partial^2 \mathcal{H}}{\partial p_x^2} & \dfrac{\partial^2 \mathcal{H}}{\partial p_x \partial p_y} & \dfrac{\partial^2 \mathcal{H}}{\partial p_x \partial p_z} \\ \dfrac{\partial^2 \mathcal{H}}{\partial p_y \partial q_x} & \dfrac{\partial^2 \mathcal{H}}{\partial p_y \partial q_y} & \dfrac{\partial^2 \mathcal{H}}{\partial p_y \partial q_z} & \dfrac{\partial^2 \mathcal{H}}{\partial p_y \partial p_x} & \dfrac{\partial^2 \mathcal{H}}{\partial p_y^2} & \dfrac{\partial^2 \mathcal{H}}{\partial p_y \partial p_z} \\ \dfrac{\partial^2 \mathcal{H}}{\partial p_z \partial q_x} & \dfrac{\partial^2 \mathcal{H}}{\partial p_z \partial q_y} & \dfrac{\partial^2 \mathcal{H}}{\partial p_z \partial q_z} & \dfrac{\partial^2 \mathcal{H}}{\partial p_z \partial p_x} & \dfrac{\partial^2 \mathcal{H}}{\partial p_z \partial p_y} & \dfrac{\partial^2 \mathcal{H}}{\partial p_z^2} \end{bmatrix} \tag{5.107}$$

$$\frac{\partial^2 \mathcal{H}}{\partial \boldsymbol{X}^2} = \begin{bmatrix} \dfrac{\partial^2 V}{\partial q_x^2} & \dfrac{\partial^2 V}{\partial q_x \partial q_y} & \dfrac{\partial^2 V}{\partial q_x \partial q_z} & 0 & -\omega_{\oplus} & 0 \\ \dfrac{\partial^2 V}{\partial q_y \partial q_x} & \dfrac{\partial^2 V}{\partial q_y^2} & \dfrac{\partial^2 V}{\partial q_y \partial q_z} & \omega_{\oplus} & 0 & 0 \\ \dfrac{\partial^2 V}{\partial q_z \partial q_x} & \dfrac{\partial^2 V}{\partial q_z \partial q_y} & \dfrac{\partial^2 V}{\partial q_z^2} & 0 & 0 & 0 \\ 0 & \omega_{\oplus} & 0 & 1 & 0 & 0 \\ -\omega_{\oplus} & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.108}$$

After multiplying the Hessian by the symplectic matrix, the $\boldsymbol{A}$ matrix is

$$\boldsymbol{A} = \begin{bmatrix} 0 & \omega_{\oplus} & 0 & 1 & 0 & 0 \\ -\omega_{\oplus} & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -\dfrac{\partial^2 V}{\partial q_x^2} & -\dfrac{\partial^2 V}{\partial q_x \partial q_y} & -\dfrac{\partial^2 V}{\partial q_x \partial q_z} & 0 & \omega_{\oplus} & 0 \\ -\dfrac{\partial^2 V}{\partial q_y \partial q_x} & -\dfrac{\partial^2 V}{\partial q_y^2} & -\dfrac{\partial^2 V}{\partial q_y \partial q_z} & -\omega_{\oplus} & 0 & 0 \\ -\dfrac{\partial^2 V}{\partial q_z \partial q_x} & -\dfrac{\partial^2 V}{\partial q_z \partial q_y} & -\dfrac{\partial^2 V}{\partial q_z^2} & 0 & 0 & 0 \end{bmatrix} \tag{5.109}$$

Solve the double partial derivatives in $\boldsymbol{A}$ with the gravitational potential for the two-body problem in Equation (5.89)

$$\boldsymbol{A} = \begin{bmatrix} 0 & \omega_{\oplus} & 0 & 1 & 0 & 0 \\ -\omega_{\oplus} & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -\dfrac{\mu}{\mathrm{r}^3} + \dfrac{3\mu x^2}{\mathrm{r}^5} & \dfrac{3\mu xy}{\mathrm{r}^5} & \dfrac{3\mu xz}{\mathrm{r}^5} & 0 & \omega_{\oplus} & 0 \\ \dfrac{3\mu xy}{\mathrm{r}^5} & -\dfrac{\mu}{\mathrm{r}^3} + \dfrac{3\mu y^2}{\mathrm{r}^5} & \dfrac{3\mu yz}{\mathrm{r}^5} & -\omega_{\oplus} & 0 & 0 \\ \dfrac{3\mu xz}{\mathrm{r}^5} & \dfrac{3\mu yz}{\mathrm{r}^5} & -\dfrac{\mu}{\mathrm{r}^3} + \dfrac{3\mu z^2}{\mathrm{r}^5} & 0 & 0 & 0 \end{bmatrix} \tag{5.110}$$

where $\mathrm{r} = \sqrt{x^2 + y^2 + z^2}$ and $\omega_{\oplus}$ is the Earth rotation rate.

Plug the osculating reference orbit position coordinates into Equation (5.110) and then calculate the local motion STM using Equation (5.104).

Numerical $\boldsymbol{A}$ Matrix from Geopotential Expansion

The only difference between the $\boldsymbol{A}$ matrix here and two-body matrix above are the second-order partial derivatives of the potential, $\partial^2 V / \partial \boldsymbol{q}^2$. Since the expansion of the geopotential is in terms of the spherical coordinates, r, ψ and λ, we can use multivariable calculus to solve for the second-order partials in terms of the generalized coordinates, q_x, q_y and q_z.

We work through the derivation of $\partial^2 V/\partial q_x^2$ and $\partial^2 V/\partial q_x \partial q_y$ in Appendix D. The other elements are derived in a similar form. The results are given here

$$\begin{aligned}\frac{\partial^2 V}{\partial q_x^2} = \frac{\partial^2 V}{\partial \mathrm{r}^2}\left(\frac{\partial \mathrm{r}}{\partial x}\right)^2 + \frac{\partial^2 V}{\partial \lambda^2}\left(\frac{\partial \lambda}{\partial x}\right)^2 + \frac{\partial^2 V}{\partial \psi^2}\left(\frac{\partial \psi}{\partial x}\right)^2 + 2\,\frac{\partial^2 V}{\partial \lambda \partial \mathrm{r}}\frac{\partial \lambda}{\partial x}\frac{\partial \mathrm{r}}{\partial x} + 2\,\frac{\partial^2 V}{\partial \psi \partial \mathrm{r}}\frac{\partial \psi}{\partial x}\frac{\partial \mathrm{r}}{\partial x} \\ + 2\,\frac{\partial^2 V}{\partial \psi \partial \lambda}\frac{\partial \psi}{\partial x}\frac{\partial \lambda}{\partial x} + \frac{\partial V}{\partial \mathrm{r}}\frac{\partial^2 \mathrm{r}}{\partial x^2} + \frac{\partial V}{\partial \lambda}\frac{\partial^2 \lambda}{\partial x^2} + \frac{\partial V}{\partial \psi}\frac{\partial^2 \psi}{\partial x^2}\end{aligned} \tag{5.111}$$

$$\begin{aligned}\frac{\partial^2 V}{\partial q_y^2} = \frac{\partial^2 V}{\partial \mathrm{r}^2}\left(\frac{\partial \mathrm{r}}{\partial y}\right)^2 + \frac{\partial^2 V}{\partial \lambda^2}\left(\frac{\partial \lambda}{\partial y}\right)^2 + \frac{\partial^2 V}{\partial \psi^2}\left(\frac{\partial \psi}{\partial y}\right)^2 + 2\,\frac{\partial^2 V}{\partial \lambda \partial \mathrm{r}}\frac{\partial \lambda}{\partial y}\frac{\partial \mathrm{r}}{\partial y} + 2\,\frac{\partial^2 V}{\partial \psi \partial \mathrm{r}}\frac{\partial \psi}{\partial y}\frac{\partial \mathrm{r}}{\partial y} \\ + 2\,\frac{\partial^2 V}{\partial \psi \partial \lambda}\frac{\partial \psi}{\partial y}\frac{\partial \lambda}{\partial y} + \frac{\partial V}{\partial \mathrm{r}}\frac{\partial^2 \mathrm{r}}{\partial y^2} + \frac{\partial V}{\partial \lambda}\frac{\partial^2 \lambda}{\partial y^2} + \frac{\partial V}{\partial \psi}\frac{\partial^2 \psi}{\partial y^2}\end{aligned} \tag{5.112}$$

$$\begin{aligned}\frac{\partial^2 V}{\partial q_z^2} = \frac{\partial^2 V}{\partial \mathrm{r}^2}\left(\frac{\partial \mathrm{r}}{\partial z}\right)^2 + \frac{\partial^2 V}{\partial \lambda^2}\left(\frac{\partial \lambda}{\partial z}\right)^2 + \frac{\partial^2 V}{\partial \psi^2}\left(\frac{\partial \psi}{\partial z}\right)^2 + 2\,\frac{\partial^2 V}{\partial \lambda \partial \mathrm{r}}\frac{\partial \lambda}{\partial z}\frac{\partial \mathrm{r}}{\partial z} + 2\,\frac{\partial^2 V}{\partial \psi \partial \mathrm{r}}\frac{\partial \psi}{\partial z}\frac{\partial \mathrm{r}}{\partial z} \\ + 2\,\frac{\partial^2 V}{\partial \psi \partial \lambda}\frac{\partial \psi}{\partial z}\frac{\partial \lambda}{\partial z} + \frac{\partial V}{\partial \mathrm{r}}\frac{\partial^2 \mathrm{r}}{\partial z^2} + \frac{\partial V}{\partial \lambda}\frac{\partial^2 \lambda}{\partial z^2} + \frac{\partial V}{\partial \psi}\frac{\partial^2 \psi}{\partial z^2}\end{aligned} \tag{5.113}$$

$$\begin{aligned}\frac{\partial^2 V}{\partial q_x \partial q_y} = \frac{\partial^2 V}{\partial \mathrm{r}^2}\frac{\partial \mathrm{r}}{\partial x}\frac{\partial \mathrm{r}}{\partial y} + \frac{\partial^2 V}{\partial \lambda \partial \mathrm{r}}\frac{\partial \lambda}{\partial x}\frac{\partial \mathrm{r}}{\partial y} + \frac{\partial^2 V}{\partial \psi \partial \mathrm{r}}\frac{\partial \psi}{\partial x}\frac{\partial \mathrm{r}}{\partial y} + \frac{\partial^2 V}{\partial \mathrm{r} \partial \lambda}\frac{\partial \mathrm{r}}{\partial x}\frac{\partial \lambda}{\partial y} + \frac{\partial^2 V}{\partial \lambda^2}\frac{\partial \lambda}{\partial x}\frac{\partial \lambda}{\partial y} \\ + \frac{\partial^2 V}{\partial \psi \partial \lambda}\frac{\partial \psi}{\partial x}\frac{\partial \lambda}{\partial y} + \frac{\partial^2 V}{\partial \mathrm{r} \partial \psi}\frac{\partial \mathrm{r}}{\partial x}\frac{\partial \psi}{\partial y} + \frac{\partial^2 V}{\partial \lambda \partial \psi}\frac{\partial \lambda}{\partial x}\frac{\partial \psi}{\partial y} + \frac{\partial^2 V}{\partial \psi^2}\frac{\partial \psi}{\partial x}\frac{\partial \psi}{\partial y} + \frac{\partial V}{\partial \mathrm{r}}\frac{\partial^2 \mathrm{r}}{\partial x \partial y} \\ + \frac{\partial V}{\partial \lambda}\frac{\partial^2 \lambda}{\partial x \partial y} + \frac{\partial V}{\partial \psi}\frac{\partial^2 \psi}{\partial x \partial y}\end{aligned} \tag{5.114}$$

$$\begin{aligned}\frac{\partial^2 V}{\partial q_x \partial q_z} = \frac{\partial^2 V}{\partial \mathrm{r}^2}\frac{\partial \mathrm{r}}{\partial x}\frac{\partial \mathrm{r}}{\partial z} + \frac{\partial^2 V}{\partial \lambda \partial \mathrm{r}}\frac{\partial \lambda}{\partial x}\frac{\partial \mathrm{r}}{\partial z} + \frac{\partial^2 V}{\partial \psi \partial \mathrm{r}}\frac{\partial \psi}{\partial x}\frac{\partial \mathrm{r}}{\partial z} + \frac{\partial^2 V}{\partial \mathrm{r} \partial \lambda}\frac{\partial \mathrm{r}}{\partial x}\frac{\partial \lambda}{\partial z} + \frac{\partial^2 V}{\partial \lambda^2}\frac{\partial \lambda}{\partial x}\frac{\partial \lambda}{\partial z} \\ + \frac{\partial^2 V}{\partial \psi \partial \lambda}\frac{\partial \psi}{\partial x}\frac{\partial \lambda}{\partial z} + \frac{\partial^2 V}{\partial \mathrm{r} \partial \psi}\frac{\partial \mathrm{r}}{\partial x}\frac{\partial \psi}{\partial z} + \frac{\partial^2 V}{\partial \lambda \partial \psi}\frac{\partial \lambda}{\partial x}\frac{\partial \psi}{\partial z} + \frac{\partial^2 V}{\partial \psi^2}\frac{\partial \psi}{\partial x}\frac{\partial \psi}{\partial z} + \frac{\partial V}{\partial \mathrm{r}}\frac{\partial^2 \mathrm{r}}{\partial x \partial z} \\ + \frac{\partial V}{\partial \lambda}\frac{\partial^2 \lambda}{\partial x \partial z} + \frac{\partial V}{\partial \psi}\frac{\partial^2 \psi}{\partial x \partial z}\end{aligned} \tag{5.115}$$

$$
\begin{aligned}
\frac{\partial^2 V}{\partial q_y \partial q_z} = \frac{\partial^2 V}{\partial \mathrm{r}^2}\frac{\partial \mathrm{r}}{\partial y}\frac{\partial \mathrm{r}}{\partial z} + \frac{\partial^2 V}{\partial \lambda \partial \mathrm{r}}\frac{\partial \lambda}{\partial y}\frac{\partial \mathrm{r}}{\partial z} + \frac{\partial^2 V}{\partial \psi \partial \mathrm{r}}\frac{\partial \psi}{\partial y}\frac{\partial \mathrm{r}}{\partial z} + \frac{\partial^2 V}{\partial \mathrm{r} \partial \lambda}\frac{\partial \mathrm{r}}{\partial y}\frac{\partial \lambda}{\partial z} + \frac{\partial^2 V}{\partial \lambda^2}\frac{\partial \lambda}{\partial y}\frac{\partial \lambda}{\partial z} \\
+ \frac{\partial^2 V}{\partial \psi \partial \lambda}\frac{\partial \psi}{\partial y}\frac{\partial \lambda}{\partial z} + \frac{\partial^2 V}{\partial \mathrm{r} \partial \psi}\frac{\partial \mathrm{r}}{\partial y}\frac{\partial \psi}{\partial z} + \frac{\partial^2 V}{\partial \lambda \partial \psi}\frac{\partial \lambda}{\partial y}\frac{\partial \psi}{\partial z} + \frac{\partial^2 V}{\partial \psi^2}\frac{\partial \psi}{\partial y}\frac{\partial \psi}{\partial z} + \frac{\partial V}{\partial \mathrm{r}}\frac{\partial^2 \mathrm{r}}{\partial y \partial z} \quad (5.116) \\
+ \frac{\partial V}{\partial \lambda}\frac{\partial^2 \lambda}{\partial y \partial z} + \frac{\partial V}{\partial \psi}\frac{\partial^2 \psi}{\partial y \partial z}
\end{aligned}
$$

where $\partial^2 V/\partial \mathrm{r}^2$, $\partial^2 V/\partial \psi^2$, $\partial^2 V/\partial \lambda^2$, $\partial^2 V/\partial \psi \partial \mathrm{r}$, $\partial^2 V/\partial \lambda \partial \mathrm{r}$, and $\partial^2 V/\partial \psi \partial \lambda$ are the second-order partials and mixed partials in spherical coordinates, $\partial V/\partial \mathrm{r}$, $\partial V/\partial \psi$, and $\partial V/\partial \lambda$ are the gravity gradients in spherical coordinates.

The geopotential gradients are found with Legendre polynomial recursions described in [174][175][176]

$$
\frac{\partial V}{\partial \mathrm{r}} = -\frac{\mu}{\mathrm{r}^2} \sum_{n=n_{\min}}^{n_{\max}} \left(\frac{R_\oplus}{\mathrm{r}}\right)^n (n+1) \sum_{m=0}^{n} \big(\bar{C}_{n,m} \cos m\lambda + \bar{S}_{n,m} \sin m\lambda\big) \bar{P}_{n,m}(\sin \psi) \qquad (5.117)
$$

$$
\frac{\partial V}{\partial \psi} = \frac{\mu}{\mathrm{r}} \sum_{n=n_{\min}}^{n_{\max}} \left(\frac{R_\oplus}{\mathrm{r}}\right)^n \sum_{m=0}^{n} \big(\bar{C}_{n,m} \cos m\lambda + \bar{S}_{n,m} \sin m\lambda\big) \frac{\mathrm{d}\bar{P}_{n,m}(\sin \psi)}{\mathrm{d}\psi} \qquad (5.118)
$$

$$
\frac{\partial V}{\partial \lambda} = \frac{\mu}{\mathrm{r}} \sum_{n=n_{\min}}^{n_{\max}} \left(\frac{R_\oplus}{\mathrm{r}}\right)^n \sum_{m=0}^{n} \big(\bar{S}_{n,m} \cos m\lambda - \bar{C}_{n,m} \sin m\lambda\big) m \bar{P}_{n,m}(\sin \psi) \qquad (5.119)
$$

where μ is the gravitational constant, r is the spherical radius, $R_\oplus$ is the Earth equatorial radius, λ is the longitude, ψ is the geocentric latitude, $\bar{C}_{n,m}$ and $\bar{S}_{n,m}$ are the fully normalized harmonics coefficients of order n and degree m, and $\bar{P}_{n,m}$ are the fully normalized Legendre polynomials of order n and degree m.

The second-order partial derivatives in spherical coordinates are

$$\begin{aligned}\frac{\partial^2 V}{\partial \mathrm{r}^2} &= \frac{\mu}{\mathrm{r}^3}\sum_{n=n_{\min}}^{n_{\max}}\left(\frac{R_\oplus}{\mathrm{r}}\right)^n (n+1)(n \\ &\qquad + 2)\sum_{m=0}^{n}\big(\bar{C}_{n,m}\cos m\lambda+\bar{S}_{n,m}\sin m\lambda\big)\bar{P}_{n,m}(\sin\psi)\end{aligned} \tag{5.120}$$

$$\frac{\partial^2 V}{\partial \mathrm{r}\partial\psi} = -\frac{\mu}{\mathrm{r}^2}\sum_{n=n_{\min}}^{n_{\max}}\left(\frac{R_\oplus}{\mathrm{r}}\right)^n (n+1)\sum_{m=0}^{n}\big(\bar{C}_{n,m}\cos m\lambda+\bar{S}_{n,m}\sin m\lambda\big)\frac{\mathrm{d}\bar{P}_{n,m}(\sin\psi)}{\mathrm{d}\psi} \tag{5.121}$$

$$\frac{\partial^2 V}{\partial \mathrm{r}\partial\lambda} = -\frac{\mu}{\mathrm{r}^2}\sum_{n=n_{\min}}^{n_{\max}}\left(\frac{R_\oplus}{\mathrm{r}}\right)^n (n+1)\sum_{m=0}^{n}\big(\bar{S}_{n,m}\cos m\lambda-\bar{C}_{n,m}\sin m\lambda\big)m\bar{P}_{n,m}(\sin\psi) \tag{5.122}$$

$$\frac{\partial^2 V}{\partial\psi^2} = \frac{\mu}{\mathrm{r}}\sum_{n=n_{\min}}^{n_{\max}}\left(\frac{R_\oplus}{\mathrm{r}}\right)^n \sum_{m=0}^{n}\big(\bar{C}_{n,m}\cos m\lambda+\bar{S}_{n,m}\sin m\lambda\big)\frac{\mathrm{d}^2\bar{P}_{n,m}(\sin\psi)}{\mathrm{d}\psi^2} \tag{5.123}$$

$$\frac{\partial^2 V}{\partial\psi\partial\lambda} = \frac{\mu}{\mathrm{r}}\sum_{n=n_{\min}}^{n_{\max}}\left(\frac{R_\oplus}{\mathrm{r}}\right)^n \sum_{m=0}^{n}\big(\bar{S}_{n,m}\cos m\lambda-\bar{C}_{n,m}\sin m\lambda\big)m\frac{\mathrm{d}\bar{P}_{n,m}(\sin\psi)}{\mathrm{d}\psi} \tag{5.124}$$

$$\frac{\partial^2 V}{\partial\lambda^2} = -\frac{\mu}{\mathrm{r}}\sum_{n=n_{\min}}^{n_{\max}}\left(\frac{R_\oplus}{\mathrm{r}}\right)^n \sum_{m=0}^{n}\big(\bar{C}_{n,m}\cos m\lambda+\bar{S}_{n,m}\sin m\lambda\big)m^2\bar{P}_{n,m}(\sin\psi) \tag{5.125}$$

The first- and second-order partials of the spherical radius are

$$\mathrm{r} = \sqrt{x^2+y^2+z^2} \tag{5.126}$$

$$\frac{\partial\mathrm{r}}{\partial x} = \frac{x}{\mathrm{r}} \tag{5.127}$$

$$\frac{\partial\mathrm{r}}{\partial y} = \frac{y}{\mathrm{r}} \tag{5.128}$$

$$\frac{\partial\mathrm{r}}{\partial z} = \frac{z}{\mathrm{r}} \tag{5.129}$$

$$\frac{\partial^2\mathrm{r}}{\partial x^2} = \frac{y^2+z^2}{\mathrm{r}^3} \tag{5.130}$$

$$\frac{\partial^2 \mathrm{r}}{\partial x \partial y} = \frac{-xy}{\mathrm{r}^3} \tag{5.131}$$

$$\frac{\partial^2 \mathrm{r}}{\partial x \partial z} = \frac{-xz}{\mathrm{r}^3} \tag{5.132}$$

$$\frac{\partial^2 \mathrm{r}}{\partial y^2} = \frac{x^2 + z^2}{\mathrm{r}^3} \tag{5.133}$$

$$\frac{\partial^2 \mathrm{r}}{\partial y \partial z} = \frac{-yz}{\mathrm{r}^3} \tag{5.134}$$

$$\frac{\partial^2 \mathrm{r}}{\partial z^2} = \frac{x^2 + y^2}{\mathrm{r}^3} \tag{5.135}$$

The first- and second-order partials of the longitude are

$$\lambda = \tan^{-1}(y/x) \tag{5.136}$$

$$\frac{\partial \lambda}{\partial x} = \frac{-y}{x^2 + y^2} \tag{5.137}$$

$$\frac{\partial \lambda}{\partial y} = \frac{x}{x^2 + y^2} \tag{5.138}$$

$$\frac{\partial \lambda}{\partial z} = 0 \tag{5.139}$$

$$\frac{\partial^2 \lambda}{\partial x^2} = \frac{2xy}{(x^2 + y^2)^2} \tag{5.140}$$

$$\frac{\partial^2 \lambda}{\partial x \partial y} = \frac{y^2 - x^2}{(x^2 + y^2)^2} \tag{5.141}$$

$$\frac{\partial^2 \lambda}{\partial x \partial z} = 0 \tag{5.142}$$

$$\frac{\partial^2 \lambda}{\partial y^2} = \frac{-2xy}{(x^2+y^2)^2} \tag{5.143}$$

$$\frac{\partial^2 \lambda}{\partial y \partial z} = 0 \tag{5.144}$$

$$\frac{\partial^2 \lambda}{\partial z^2} = 0 \tag{5.145}$$

The first- and second-order partials of the geocentric latitude are

$$\psi = \sin^{-1}(z/\mathrm{r}) \tag{5.146}$$

$$\frac{\partial \psi}{\partial x} = -\frac{xz}{\mathrm{r}^3\sqrt{1-\frac{z^2}{\mathrm{r}^2}}} \tag{5.147}$$

$$\frac{\partial \psi}{\partial y} = -\frac{yz}{\mathrm{r}^3\sqrt{1-\frac{z^2}{\mathrm{r}^2}}} \tag{5.148}$$

$$\frac{\partial \psi}{\partial z} = \frac{\frac{1}{\mathrm{r}}\left(1-\frac{z^2}{\mathrm{r}^2}\right)}{\sqrt{1-\frac{z^2}{\mathrm{r}^2}}} \tag{5.149}$$

$$\frac{\partial^2 \psi}{\partial x^2} = \frac{x^2z^3}{\mathrm{r}^7\left(1-\frac{z^2}{\mathrm{r}^2}\right)^{\frac{3}{2}}} + \frac{3x^2z}{\mathrm{r}^5\sqrt{1-\frac{z^2}{\mathrm{r}^2}}} - \frac{z}{\mathrm{r}^3\sqrt{1-\frac{z^2}{\mathrm{r}^2}}} \tag{5.150}$$

$$\frac{\partial^2 \psi}{\partial x \partial y} = \frac{xyz^3}{\mathrm{r}^7\left(1-\frac{z^2}{\mathrm{r}^2}\right)^{\frac{3}{2}}} + \frac{3xyz}{\mathrm{r}^5\sqrt{1-\frac{z^2}{\mathrm{r}^2}}} \tag{5.151}$$

$$\frac{\partial^2 \psi}{\partial x \partial z} = \frac{xz\left(\frac{2z^3}{\mathrm{r}^4}-\frac{2z}{\mathrm{r}^2}\right)}{2\mathrm{r}^3\left(1-\frac{z^2}{\mathrm{r}^2}\right)^{\frac{3}{2}}} + \frac{3xz^2}{\mathrm{r}^5\sqrt{1-\frac{z^2}{\mathrm{r}^2}}} - \frac{x}{\mathrm{r}^3\sqrt{1-\frac{z^2}{\mathrm{r}^2}}} \tag{5.152}$$

$$\frac{\partial^2 \psi}{\partial y^2} = \frac{y^2z^3}{\mathrm{r}^7\left(1-\frac{z^2}{\mathrm{r}^2}\right)^{\frac{3}{2}}} + \frac{3y^2z}{\mathrm{r}^5\sqrt{1-\frac{z^2}{\mathrm{r}^2}}} - \frac{z}{\mathrm{r}^3\sqrt{1-\frac{z^2}{\mathrm{r}^2}}} \tag{5.153}$$

$$\frac{\partial^2\psi}{\partial y\partial z}=\frac{yz\left(\frac{2z^3}{\mathrm{r}^4}-\frac{2z}{\mathrm{r}^2}\right)}{2\mathrm{r}^3\left(1-\frac{z^2}{\mathrm{r}^2}\right)^{\frac{3}{2}}}+\frac{3yz^2}{\mathrm{r}^5\sqrt{1-\frac{z^2}{\mathrm{r}^2}}}-\frac{y}{\mathrm{r}^3\sqrt{1-\frac{z^2}{\mathrm{r}^2}}} \tag{5.154}$$

$$\frac{\partial^2\psi}{\partial z^2}=\frac{\frac{3z^3}{\mathrm{r}^5}-\frac{3z}{\mathrm{r}^3}}{\sqrt{1-\frac{z^2}{\mathrm{r}^2}}}-\frac{\left(\frac{2z^3}{\mathrm{r}^4}-\frac{2z}{\mathrm{r}^2}\right)\left(\frac{1}{\mathrm{r}}-\frac{z^2}{\mathrm{r}^3}\right)}{2\left(1-\frac{z^2}{\mathrm{r}^2}\right)^{\frac{3}{2}}} \tag{5.155}$$

Evaluate each of the partials using the IOD reference orbit position coordinates in the ECEF frame at the observation times. Substitute the partial derivatives into Equation (5.109) to generate the $\boldsymbol{A}$ matrix at each time step.

5.5.4 Verification

The same four hypothesis verification cases from Section 5.3.7 are used to demonstrate the accuracy of the filter dynamics model. Recall that this included: 1) Case 06A - low differential argument of perigee and two-body dynamics; 2) Case 06B - low differential argument of perigee and EGM96 20×20 dynamics; 3) Case 08A - high differential argument of perigee and two-body dynamics; and 4) Case 08B - high differential argument of perigee and EGM96 20×20 dynamics. The low $\delta\omega$ case had a smaller error than the high $\delta\omega$ case, so this represents a reasonable error spread for testing the filter post-IOD. The hypotheses are used as the reference orbit and are numerically propagated in sequence with the true observations rather than using a torus. Doing so restricts errors generated by the torus. For Case 06A and Case 08A, the analytical two-body $\boldsymbol{A}$ matrix is used in the filter and for Case 06B and Case 08B, the numerical $\boldsymbol{A}$ matrix from the geopotential expansion is used. Since the true orbits are known, we can assess how well the filter corrects the hypotheses. No uncertainty in observations or dynamics are assumed during verification.

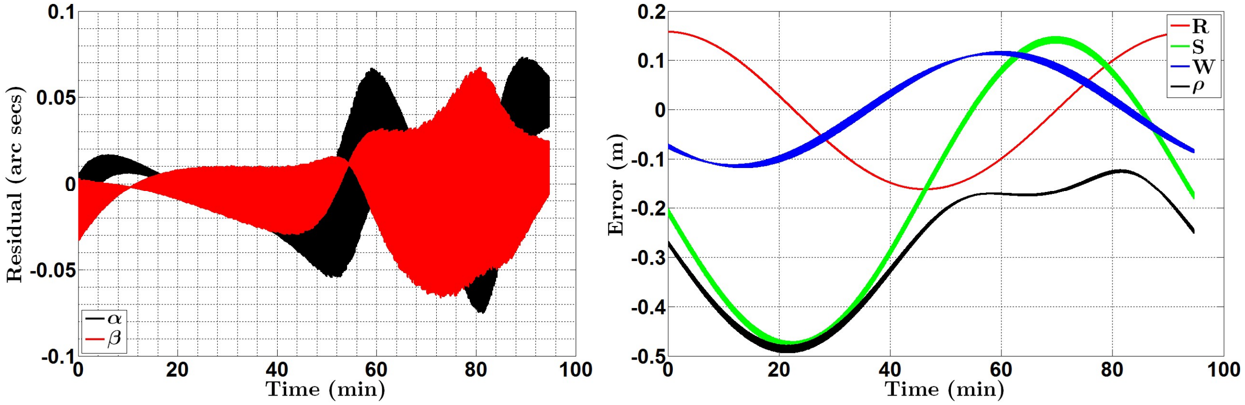
LEO, Low Differential Argument of Perigee, Two-Body (Case 06A)

The true error in the initial state vector of the hypothesis is listed in Table 8. The reference correction, ${}^{\mathrm{e}}\delta\widetilde{\boldsymbol{X}}_{\mathrm{ref}}(t_0)$, should converge toward these values if implemented correctly.

Table 8: Case 06A: IOD Hypothesis Error at Epoch & NLS Correction

ECEF	IOD Error	NLS Correction
x (m)	-292.5480051180	-292.3002844542
y (m)	105.5353403398	105.6427801884
z (m)	96.43949928977	96.42259807531
$\dot{x}$ (m/s)	-0.075237535075	-0.075021411164
$\dot{y}$ (m/s)	-0.125235554641	-0.125048483922
$\dot{z}$ (m/s)	-0.121723634505	-0.121678923869

Using a convergence criterion of $|\delta\boldsymbol{\rho}_0| < 0.1$ m, the filter converges after three iterations. The angle residuals are plotted in Figure 46(a). The absolute error in each of the LVLH coordinates and the range are shown in Figure 46(b). The results indicate that the two-body $\boldsymbol{A}$ matrix and NLS filter have been implemented correctly.



(a) Angle Residuals **(b) Position Error of Hypothesis**

Figure 46: Filter Accuracy for Case 06A. The angle residuals are plotted in (a). The black curve is for α and the red curve is for β. The range error and coordinate errors (Hill's frame) are plotted in (b). The red curve is the radial error, the green curve is the along-track error, the blue curve is the cross-track error and the black curve is the range error.

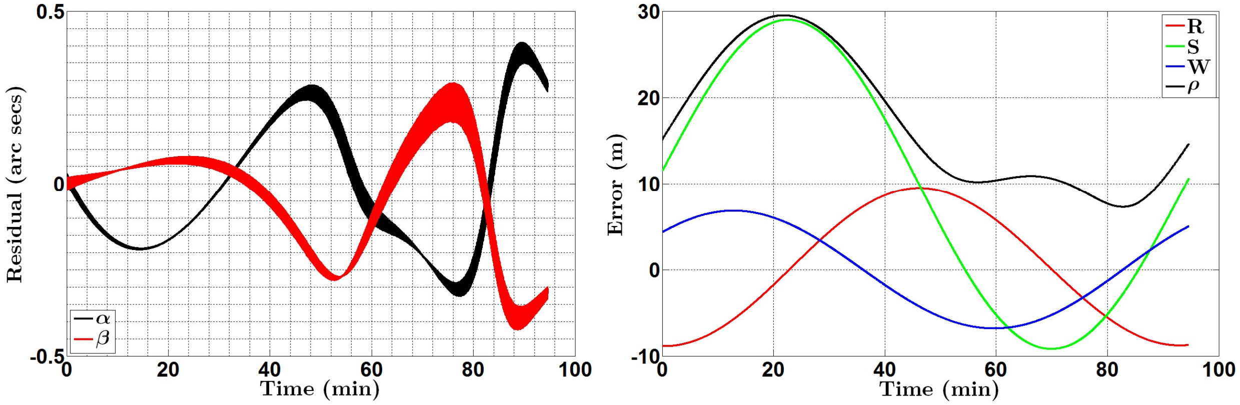
LEO, Low Differential Argument of Perigee, EGM96 20 × 20 (Case 06B)

The true error in the initial state vector of the hypothesis is listed in Table 9. The reference correction, ${}^{\mathrm{e}}\delta\widetilde{\boldsymbol{X}}_{\mathrm{ref}}(t_0)$, should converge toward these values if implemented correctly.

Table 9: Case 06B: IOD Hypothesis Error at Epoch & NLS Correction

ECEF	IOD Error	NLS Correction
x (m)	-123.7280280561	-137.4531728045
y (m)	-771.2614865881	-777.5015124360
z (m)	409.9048193729	410.8923916120
$\dot{x}$ (m/s)	-0.263253394749	-0.277545506660
$\dot{y}$ (m/s)	0.904065222081	0.893730633613
$\dot{z}$ (m/s)	-0.528897602688	-0.530271102794

Using a convergence criterion of $|\delta\boldsymbol{\rho}_0| < 0.1$ m, the filter converges after four iterations. The angle residuals are plotted in Figure 47(a). The absolute error in each of the LVLH coordinates and the range are shown in Figure 47(b). The results are not quite as good as Case 06A. A larger initial error in reference orbit may explain the estimate error, but this is a completely acceptable result that suggests the numerical $\boldsymbol{A}$ matrix and geopotential functionals have been implemented correctly. For some perspective, the range error throughout the estimated orbit is ~0.027%—well below the 1% objective.



(a) Angle Residuals (b) Position Error of Hypothesis

Figure 47: Filter Accuracy for Case 06B. The angle residuals are plotted in (a). The black curve is for α and the red curve is for β. The range error and coordinate errors (Hill's frame) are plotted in (b). The red curve is the radial error, the green curve is the along-track error, the blue curve is the cross-track error and the black curve is the range error.

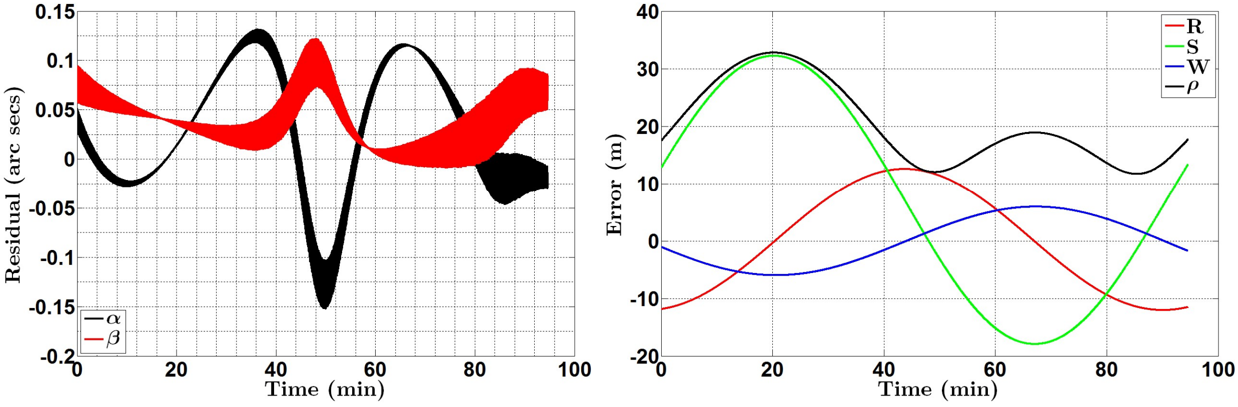
LEO, High Differential Argument of Perigee, Two-Body (Case 08A)

The true error in the initial state vector of the hypothesis is listed in Table 10. The reference correction, ${}^{\mathrm{e}}\delta\boldsymbol{X}_{\mathrm{ref}}(t_0)$, should converge toward these values if implemented correctly.

Table 10: Case 08A: IOD Hypothesis Error at Epoch & NLS Correction

ECEF	IOD Error	NLS Correction
x (m)	724.3579646337	740.3764464078
y (m)	-3.641745192908	-9.017772746998
z (m)	84.58829722076	80.11643084178
$\dot{x}$ (m/s)	0.699779014041	0.717037668591
$\dot{y}$ (m/s)	-0.145345612043	-0.146639616719
$\dot{z}$ (m/s)	-0.600395421750	-0.608796418315

Using a convergence criterion of $|\delta\boldsymbol{\rho}_0| < 0.1$ m, the filter converges after three iterations. The angle residuals are plotted in Figure 48(a). The absolute error in each of the LVLH coordinates and the range are shown in Figure 48(b). The residuals are larger than in Case 06, but this is expected due to the larger IOD errors. Any divergence of the reference orbit from the truth limits the accuracy of the final estimate since the dynamics are based upon the global and local motion of the reference orbit. Even still, it is quite remarkable to see the filter reduce the position error from the the kilometer-level to meter-level (~0.023% range error on average) in this case.



(a) Angle Residuals (b) Position Error of Hypothesis

Figure 48: Filter Accuracy for Case 08A. The angle residuals are plotted in (a). The black curve is for α and the red curve is for β. The range error and coordinate errors (Hill's frame) are plotted in (b). The red curve is the radial error, the green curve is the along-track error, the blue curve is the cross-track error and the black curve is the range error.

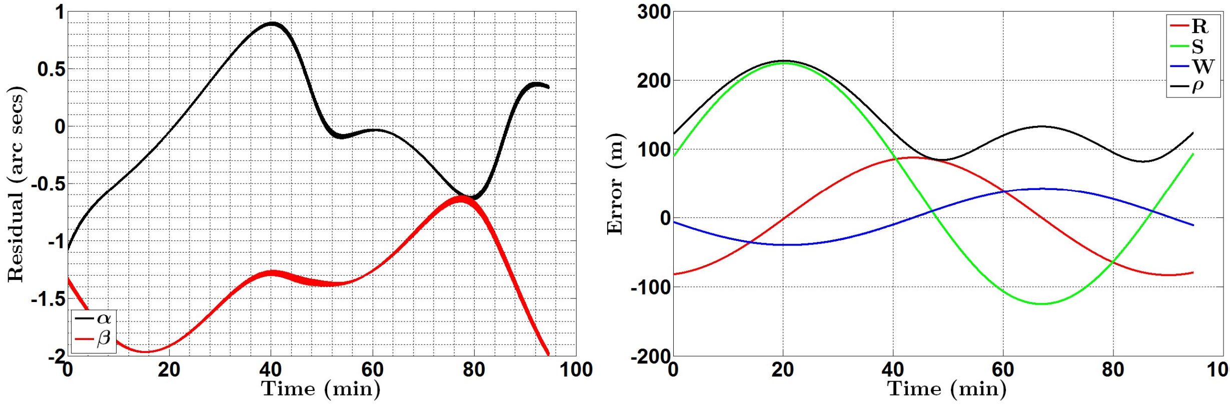
LEO, High Differential Argument of Perigee, EGM96 20 × 20 (Case 08B)

The true error in the initial state vector of the hypothesis is listed in Table 11. The reference correction, ${}^{\mathrm{e}}\delta \boldsymbol{X}_{\mathrm{ref}}(t_0)$, should converge toward these values if implemented correctly.

Table 11: Case 08B: IOD Hypothesis Error at Epoch & NLS Correction

ECEF	IOD Error	NLS Correction
x (m)	2057.679842367	2168.609454297
y (m)	-1508.958988079	-1545.961906750
z (m)	356.4651072556	326.2869915419
$\dot{x}$ (m/s)	2.461728478731	2.581121488380
$\dot{y}$ (m/s)	1.060053006694	1.049435567522
$\dot{z}$ (m/s)	-2.429334581434	-2.487631514490

Using a convergence criterion of $|\delta \boldsymbol{\rho}_0| < 0.1$ m, the filter converges after four iterations. The angle residuals are plotted in Figure 49(a). The absolute error in each of the LVLH coordinates and the range are shown in Figure 49(b). The residuals are the largest of the four tests, but this result is expected since the reference orbit has a 2.2 km range error at epoch. The final estimate of the Target's orbit has a maximum error of 228 m in range—just 0.16% of the true range. The methodology appears to be fairly robust to IOD errors from large differentials in argument of perigee.



(a) Angle Residuals

(b) Position Error of Hypothesis

Figure 49: Filter Accuracy for Case 08B. The angle residuals are plotted in (a). The black curve is for α and the red curve is for β. The range error and coordinate errors (Hill's frame) are plotted in (b). The red curve is the radial error, the green curve is the along-track error, the blue curve is the cross-track error and the black curve is the range error.

Verification Insight

In a controlled experiment where the numerically integrated IOD reference orbit was the only source of error, the batch filter performed remarkably well. In all cases, errors after differential correction were no worse than 0.16% of the true range. This is excellent compared to the 1% rule of thumb for conducting RPOs. Case 08 represents one of the most stressing cases for our eccentricity assumption with a high differential argument of perigee, yet the filter performed well with both $\boldsymbol{A}$ matrix formulations. We have high confidence that the dynamics model, measurement model and filter are implemented correctly.

5.6 Reference KAM Torus

The Hamiltonian for Earth-orbiting satellites in the expanded geopotential was previously given by Equation (5.93) as a function of the generalized coordinates, $\boldsymbol{q} = (q_1, q_2, q_3)$, and their conjugate momenta, $\boldsymbol{p} = (p_1, p_2, p_3)$. KAM theorem seeks to map this lightly perturbed Hamiltonian to one represented by new coordinates, $\boldsymbol{Q} = (Q_1, Q_2, Q_3)$, and momenta, $\boldsymbol{P} = (P_1, P_2, P_3)$, in which only the momenta appear in the new Hamiltonian, $\mathcal{K}(\boldsymbol{P})$. See [155] for further discussion of the new Hamiltonian and generating function as approximated with the Delaunay variables. The absence of the new coordinates merely implies from Hamilton-Jacobi theory that the coordinates have constant frequencies. Indeed, Earth orbiting satellites exhibit three fundamental frequencies in the ECEF frame: an anomalistic frequency, ω_1, which is just the Keplerian frequency with a correction for the nonspherical Earth; a nodal frequency, ω_2, which is a combination of the Earth's rotation rate and the regression rate of the line of nodes; and an apsidal frequency, ω_3, which is the rotation rate of the line of apsides due to Earth's oblateness.

The three new momenta are integrals of motion which mandates that the solution lie on a three-dimensional manifold which is topologically equivalent to a 3-torus [177]. In other words, the new momenta resemble the actions from three sets of action-angle pairs and the new coordinates are the angles describing the position on each of the three action circuits.

An easy homeomorphism from the native Cartesian space to the phase space of a torus does not currently exist, but the common quasi-periodic nature of the native coordinates and the torus angles makes an N-tuple Fourier series a possible solution for mapping between both spaces. There are generally two approaches used for constructing the torus Fourier series: 1) trajectory-following techniques that perform a Fourier decomposition of data from long numerically integrated orbits, and 2) iterative techniques that find successively better approximations of the series using Hamilton's equations [178]. Binney and Spergel [179][180] pioneered the first approach for non-integrable galactic dynamics and Wiesel spearheaded the first known efforts for earth orbits [153][154]. Wiesel's two-pass trajectory-following approach is applied here; fundamental frequencies are identified on the first pass and the series coefficients are extracted on the second pass.

The aim is to construct the native coordinates in the time domain from a finite Fourier series

$$\boldsymbol{q}(t) = \sum_k \boldsymbol{D}_k(\boldsymbol{I})\exp(i(l_k\omega_1 + m_k\omega_2 + n_k\omega_3)t) \tag{5.156}$$

where $\boldsymbol{D}_k$ are the complex series coefficients associated with the discrete frequencies $\omega_k = (l_k\omega_1 + m_k\omega_2 + n_k\omega_3)$, and (l_k, m_k, n_k) are integers for combining the fundamental frequencies. The more conventional real form gives the Fourier series as:

$$\boldsymbol{q}(t) = \boldsymbol{C}_0 + \sum_k \boldsymbol{C}_k \cos(l_k Q_1 + m_k Q_2 + n_k Q_3) + \boldsymbol{S}_k \cos(l_k Q_1 + m_k Q_2 + n_k Q_3) \tag{5.157}$$

$$\boldsymbol{Q}(t) = \boldsymbol{\omega}(t - t_0) + \boldsymbol{Q}(t_0) \tag{5.158}$$

where $\boldsymbol{Q}$ are the torus angle variables incremented linearly in time, $\boldsymbol{\omega}$ is the set of fundamental frequencies found through the first pass through the power spectrum, $\boldsymbol{C}_k$ and $\boldsymbol{S}_k$ are the Fourier coefficients or the amplitudes of the Fourier transform at each integer combination of the fundamental frequencies. The combination of (l_k, m_k, n_k) uses an incrementing scheme that avoids repeated frequency combinations in the numerical routine.

A time derivative gives the velocities

$$\begin{aligned}\dot{\boldsymbol{q}}(t) = \sum_k (l_k\omega_1 + m_k\omega_2 + n_k\omega_3)[&\boldsymbol{S}_k \cos(l_k Q_1 + m_k Q_2 + n_k Q_3) \\ &- \boldsymbol{C}_k \cos(l_k Q_1 + m_k Q_2 + n_k Q_3)]\end{aligned} \tag{5.159}$$

Figure 50 depicts an analog of the torus coordinates in the ECEF reference frame where Q_1 and Q_3 combine to define the satellite's position in the orbital plane and Q_2 designates the line of nodes.

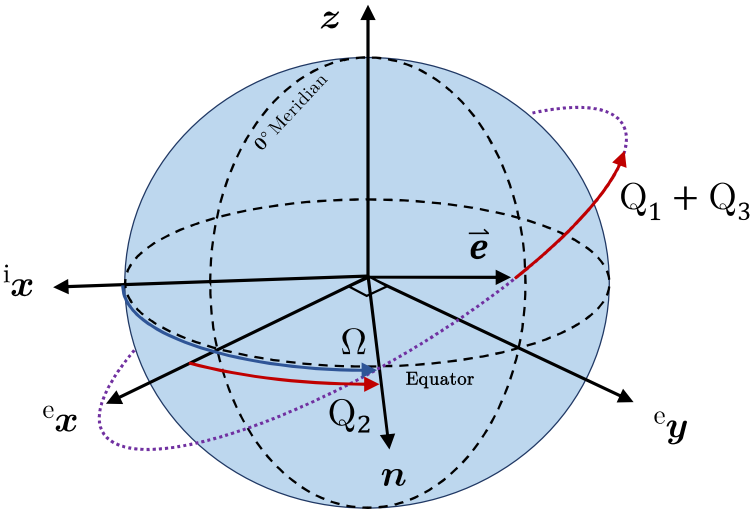


Figure 50: Torus Angle Analog in the ECEF Frame. The three torus coordinates are loosely contrasted with Keplerian elements since the Fourier coefficients can be transformed to coincide with any $Q(t_0)$. Q_1 and Q_3 combine to define the satellite's position in the orbital plane and Q_2 designates the line of nodes.

One may ponder the exact relationship between the Fourier series coefficients and the action-angle variables that describe the satellite position on the torus. From classical mechanics, the constant actions define the shape of invariant tori, while the angles are the coordinates on the tori. Mathematically, the canonical angle coordinates are those given in Equation (5.158), but their conjugate action momenta do not appear explicitly in the Fourier series. The momenta represent the dimensions in phase space that are directed away from the torus surface and are only implicitly present through the series coefficients. Nevertheless, they can still be calculated explicitly from the Poincaré integral invariants [153]

$$P_i = \frac{1}{2\pi} \oint_{\Gamma_i} \boldsymbol{p} \cdot \mathrm{d}\boldsymbol{q} = \frac{1}{2\pi} \int_0^{2\pi} \boldsymbol{p} \cdot \frac{\partial \boldsymbol{q}}{\partial Q_i} \mathrm{d}Q_i \tag{5.160}$$

where Q_i are the angle coordinates, $\boldsymbol{q}$ and $\boldsymbol{p}$ are the native coordinates and momenta, and Γ_i is a fundamental contour about the torus. As the system oscillates around the torus, the time derivative of the coordinates is equivalent to the fundamental frequencies. Since the action momenta are constant on a torus, their time derivative is simply zero, which infers the system's Hamiltonian function is only a function of action momenta. This makes sense because the Hamiltonian is conserved. As such, the Hamiltonian equations of motion are resolved as:

$$\frac{dQ_i}{dt} = \frac{\partial \mathcal{K}(\boldsymbol{P})}{\partial P_i} = \omega_i(\boldsymbol{P}) \tag{5.161}$$

$$\frac{dP_i}{dt} = -\frac{\partial \mathcal{K}(\boldsymbol{P})}{\partial Q_i} = 0 \tag{5.162}$$

From Equation (5.161) it is apparent that the frequencies in the orbital motion are a function of the action momenta. Thus, the influence of the action momenta on the Fourier series is apparent in the Fourier coefficients as amplitudes of the frequencies and their combinations in time.

5.6.1 Trajectory Following Fourier Analysis

A spectral analysis of a numerical integration of the reference orbit allows us to identify the fundamental frequencies and decompose the orbit into a Fourier series. A variation of Jacques Laskar's Numerical Algorithm of the Fundamental Frequency (NAFF) [181][182] is the machinery of choice for finding the fundamental frequencies and coefficients to construct the torus Fourier series. The NAFF is a technique that approximates the truncated, continuous Fourier transform (TCFT) so that prominent spectral lines are identified without the destructive effects of aliasing and leakage that will be discussed shortly.

A one-year numerical integration (6 months forward and 6 months backward from epoch) in the full geopotential is generally acceptable. The transformation of the numerically integrated coordinate data, $f(t)$, from the time domain to the frequency domain is done using a finite Fourier transform over the symmetric time interval $[-T, T]$ where T is 6 months. The typical Fourier transform of a function over an infinite time span is performed by

$$\mathcal{F}(\nu) = \int_{-\infty}^{\infty} f(t) e^{-2\pi\nu i t} \mathrm{d}t \tag{5.163}$$

where ν is the cycle frequency. The Fourier transform assumes infinite periodicity in the signal, but over an arbitrary, finite time interval, it is doubtful that the signal endpoints are of the same value. The consequence of this discontinuity is a phenomenon known as spectral leakage in the Fourier transform. To inhibit this, Laskar multiplies the signal by a Hanning window function, χ_p, that forces the signal to start and end at zero amplitude

$$\chi_p\left(\frac{t}{T}\right) = \frac{2^p (p!)^2}{(2p)!} \left(1 + \cos\left(\pi \frac{t}{T}\right)\right)^p \tag{5.164}$$

where p is the order of the cosine function and T is the frequency interval. The window function is shown graphically for various values of p in Figure 51(a) on the normalized frequency interval $[-1, 1]$.

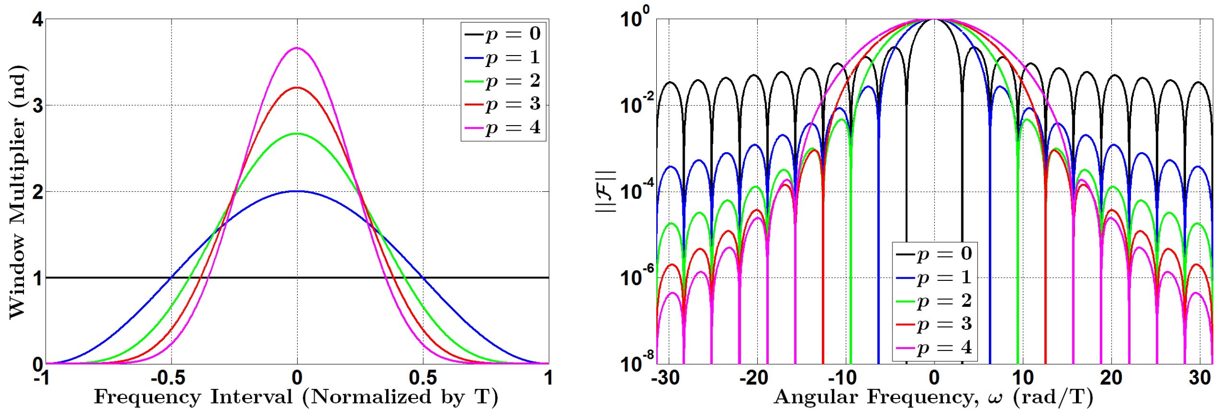
The finite Fourier transform with the window function over the timespan $[-T, T]$ is

$$\begin{aligned}\mathcal{F}(\nu) &= \frac{1}{2T}\int_{-T}^{T} f(t)e^{-2\pi\nu i t}\chi_p\left(\frac{t}{T}\right)\mathrm{d}t \\ &= \frac{1}{2T}\int_{0}^{T} f(t)e^{-2\pi\nu i t}\chi_p\left(\frac{t}{T}\right)\mathrm{d}t - \frac{1}{2T}\int_{0}^{-T} f(t)e^{-2\pi\nu i t}\chi_p\left(\frac{t}{T}\right)\mathrm{d}t\end{aligned} \tag{5.165}$$

The domain of the Fourier transform can be expressed as cycle frequency, ν, or angular frequency, $\omega = 2\pi\nu$, but the latter will typically be used for the torus.

Choosing an appropriate window power is essential to dissecting spectral content from the transformed data. The effect of increasing the window power is demonstrated by applying a Fourier transform on a single spectral line of unit amplitude over the timespan $[-1, 1]$ and shown in Figure 51(b)

$$\mathcal{F}(\omega) = \frac{1}{2}\int_{-1}^{1} e^{\omega i t}\chi_p(t)\mathrm{d}t = \frac{(-1)^p\pi^{2p}(p!)^2\sin\omega}{\omega(\omega^2-\pi^2)\cdots(\omega^2-p^2\pi^2)} \tag{5.166}$$



(a) Hanning Window of Order p

(b) Sidelobe Effects from $\mathcal{F}_{p=0}(\omega)$ to $\mathcal{F}_{p=4}(\omega)$

Figure 51: Hanning Window Characteristics. Laskar's method uses a Hanning window to prevent spectral leakage by forcing the signal to start and end at zero amplitude as in (a). The window function has an effect on discrete spectral lines causing the main lobe to widen and side lobes to fall off more rapidly as p increases (b).

In this example, the signal is a gate function and has a single spectral line at $\omega_0 = 0$, but Figure 51(b) shows side lobe oscillations of $\omega_T = \pi/T$ which appear from the cosine term in Equation (5.164). As p increases, the main lobe of the ω_0 spectral line broadens and the side lobes around it fall off more rapidly. The amplitude of the spectral line remains the same with increasing p, so the advantage of higher order window functions is an accelerated convergence upon a discrete frequency. One disadvantage occurs in signals containing integer combinations of frequency sets in which one frequency is significantly smaller than the other frequencies. When this occurs, such as the case of cascading harmonics from the small apsidal frequency, the higher order window functions can "swallow" nearby spectral content. This is known as spectral shadowing.

Ultimately, the choice of the Hanning window power is contingent upon the signal characteristics, but Laskar shows for KAM solutions that the accuracy of the frequency analysis (order $1/T^{2p+2}$) generally improves by increasing p. He suggests using the highest possible value of p until precision begins to decrease. A Hanning window of order $p = 2$ is used in this work to avoid spectral shadowing.

Once the reference orbit has been transformed into the frequency domain, the next step is to acquire the fundamental frequencies. The process for finding these frequencies is iterative. Begin with an approximate set using Lagrange's planetary equations

$$\omega_1 = \sqrt{\frac{\mu}{a^3}} - \frac{3\sqrt{\mu} J_2 R_\oplus^2}{2a^{7/2}(1-e^2)^{3/2}} \left(\frac{3}{2}\sin^2(i) - 1\right) \tag{5.167}$$

$$\omega_2 = \omega_\oplus - \frac{3\sqrt{\mu}J_2 R_\oplus^2}{2a^{7/2}(1-e^2)^2}\cos i \tag{5.168}$$

$$\omega_3 = -\frac{3\sqrt{\mu}J_2 R_\oplus^2}{2a^{7/2}(1-e^2)^2}\left(\frac{5}{2}\sin^2(i) - 2\right) \tag{5.169}$$

Recall that discrete peaks in the spectrum appear as integer combinations of the fundamental frequencies: $\omega_k = (l_k\omega_1 + m_k\omega_2 + n_k\omega_3)$. The approximate frequencies from Equations (5.167) - (5.169) are not good enough to precisely identify the spectral peaks in each signal (x, y, z), but they get close enough to find the real peaks in small segments of the frequency spectrum $(\omega_k +/{-\Delta\omega})$ using an FFT. Only four discrete peaks are targeted in this step

$$\omega_{(1,0,1)} = \omega_1 + \omega_3 \tag{5.170}$$

$$\omega_{(1,-1,1)} = \omega_1 - \omega_2 + \omega_3 \tag{5.171}$$

$$\omega_{(2,0,2)} = 2\omega_1 + 2\omega_3 \tag{5.172}$$

$$\omega_{(2,0,1)} = 2\omega_1 + \omega_3 \tag{5.173}$$

where $\omega_{(l,m,n)} = (l\omega_1 + m\omega_2 + n\omega_3)$.

Since the FFT is coarsely sampled, a Newton-Raphson (NR) algorithm [153] seeks the maximum spectral power, $P = |\mathscr{F}(\omega)|^2$, near the peaks found from the FFT. After NR converges, the frequency is recorded and the peak is subtracted from the spectral data in each coordinate. The improved frequencies are

$$\omega_1 = {}^{z}\omega_{(2,0,1)} - {}^{z}\omega_{(1,0,1)} \tag{5.174}$$

$$\omega_2 = -\left(\frac{\left({}^{x}\omega_{(1,-1,1)} + {}^{y}\omega_{(1,-1,1)}\right)}{2} - {}^{z}\omega_{(1,0,1)}\right) \tag{5.175}$$

$$\omega_3 = {}^{z}\omega_{(2,0,2)} - {}^{z}\omega_{(2,0,1)} \tag{5.176}$$

where the left superscript indicates which extracted signal frequency is used.

A similar process is used to refine the frequencies further, but this time using 36 peaks and smaller FFT segments prior to NR. To simplify the notation for identifying all 36 peaks, we simply define the integers (l, m, n). The peaks are extracted in the following order in an attempt to minimize spectral shadowing: (1,0,1); (1,1,1); (1,2,1); (1,3,1); (1,4,1); (1,5,1); $(1,-1,1)$; $(1,-2,1)$; $(1,-3,1)$; $(1,-4,1)$; $(1,-5,1)$; (2,0,2); (2,1,2); (2,2,2); (2,3,2); (2,4,2); (2,5,2); $(2,-1,2)$; $(2,-2,2)$; $(2,-3,2)$; $(2,-4,2)$; $(2,-5,2)$; (3,0,3); (3,1,3); (3,2,3); (3,3,3); (3,4,3); (3,5,3); $(3,-1,3)$; $(3,-2,3)$; $(3,-3,3)$; $(3,-4,3)$; $(3,-5,3)$; (2,0,1); (3,0,2); (3,0,1). The frequencies from the three signals are averaged for each peak and a least squares fit of the 36 integer combinations provides an improved approximation of the fundamental frequencies.

In one last attempt to refine the frequencies, we use the improved frequencies to extract 12 peaks in the following order: (1,0,1); (1,1,1); $(1,-1,1)$; $(2,0,2)^*$; (2,0,1); $(2,1,2)^*$; (2,1,1); $(2,-1,2)^*$; $(2,-1,1)$. The three integer combinations indicated with an asterisk are only extracted in an attempt to reduce spectral shadowing if they have larger peaks than the peaks succeeding them. The final fundamental set is rendered from a least squares fit of 10 selected signal frequencies: ${}^z\omega_{(1,0,1)}$, ${}^x\omega_{(1,1,1)}$, ${}^y\omega_{(1,1,1)}$, ${}^x\omega_{(1,-1,1)}$, ${}^y\omega_{(1,-1,1)}$, ${}^z\omega_{(2,0,1)}$, ${}^x\omega_{(2,1,1)}$, ${}^y\omega_{(2,1,1)}$, ${}^x\omega_{(2,-1,1)}$, ${}^y\omega_{(2,-1,1)}$.

With the estimated fundamental frequencies, the coefficients of the Fourier series are extracted by evaluating the real and imaginary parts of the Fourier transform

$$C_0 = \Re\mathcal{F}(0) \tag{5.177}$$

$$C_k = 2\Re\mathcal{F}(\omega_k) \tag{5.178}$$

$$S_k = -2\Im\mathcal{F}(\omega_k) \tag{5.179}$$

$\mathcal{F}(\omega_k)$ is the complex Fourier transform at the composite frequency ω_k, or in the case of the constant term, C_0, at $\omega = 0$.

Generally, two extraction passes are needed to correct coefficients that experience spectral shadowing on the first pass.

Chapter 6

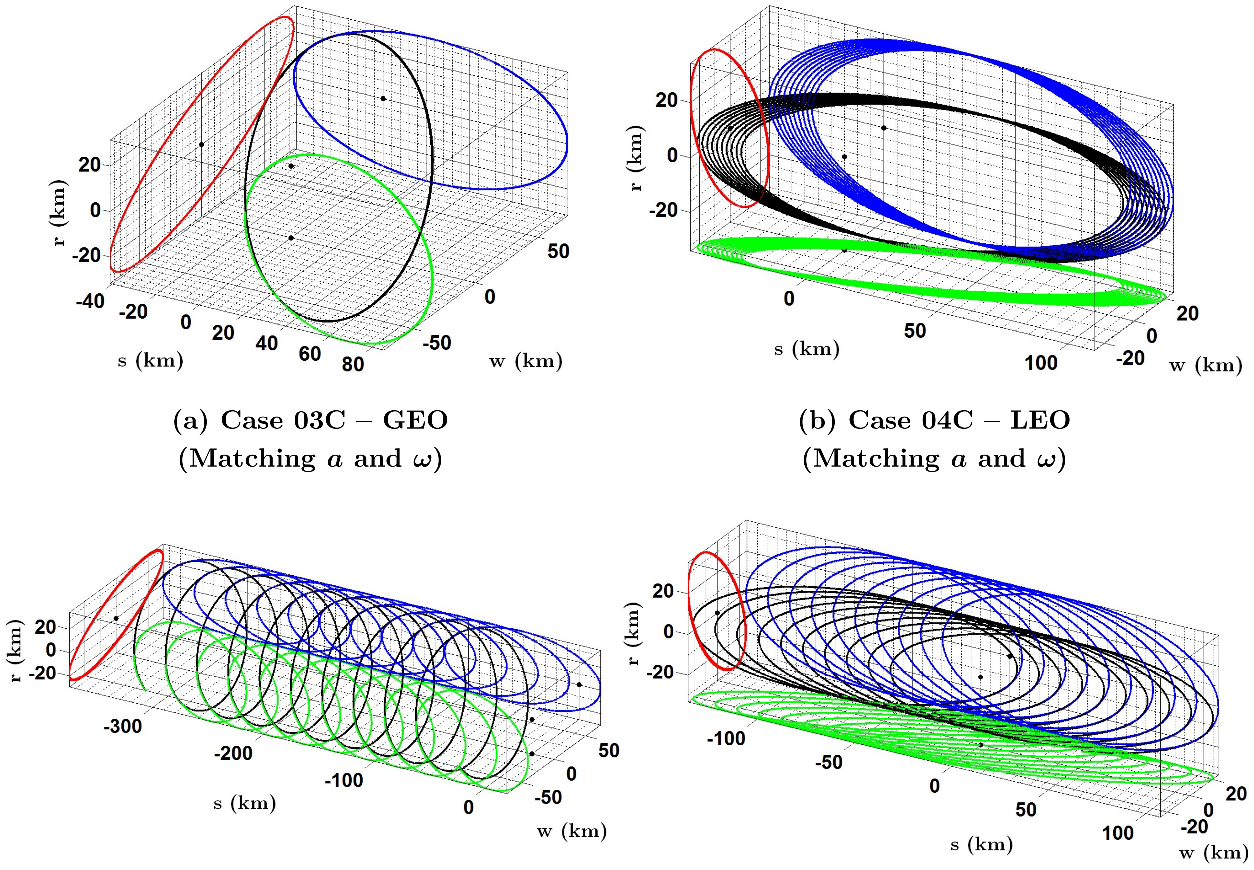
Case Study Results

An infinite number of NMC orbit scenarios could be used to evaluate our IOD and estimation process, but a subset are formulated from general characteristics of circumnavigation orbits. The existence of an NMC trajectory means $\delta a \ll a$ and station-keeping operations necessitate that $\delta a \approx 0$. Additionally, δe, δi, $\delta \Omega$ and δu are quite small within the optical navigation capability gap or else the extent of motion will exceed the Bingham threshold where classical angles-only methods become tractable. $\delta \omega$ and $\delta \nu$ can be large, but are traded to keep δu small ($\delta \Omega$ can also be traded in δu to a lesser extent). These expectations merit three basic scenarios: 1) the Target is station-keeping ($\delta a \approx 0$) with an idealized matching argument of perigee ($\delta \omega \approx 0$) and low differentials in the other orbital elements; 2) low differentials in all orbital elements; and 3) large differentials in $\delta \omega$ and $\delta \nu$ while all other differentials are small. For each of these scenarios, one is performed at GEO and another at LEO. None of the scenario parameters is treated as an assumption in the research code; there is no prior knowledge of any of this to aid the algorithm.

Table 12 lists the test cases and the associated differential orbital elements. Figure 52 shows what each of the trajectories looks like with 10 orbits in the full geopotential for physical intuition. Only the *first orbit* is used to estimate the Target's state, though. A substantial amount of drift occurs in all but the energy matching cases (Case 03C and Case 04C), so it is understandable why δa must remain small to perform persitent proximity operations.

Table 12: Outline of Case Study Differential Orbital Elements

Case	Orbit	δa (km)	δe (nd)	δi (°)	$\delta\Omega$ (°)	$\delta\omega$ (°)	$\delta\nu$ (°)
03C	GEO	0.0	0.00075	0.1	-0.03	0.000	0.060
04C	LEO	0.0	0.00500	0.2	0.10	0.000	0.300
05C	GEO	3.0	0.00075	0.1	-0.03	0.100	-0.120
06C	LEO	1.0	0.00500	0.2	0.10	-0.200	0.500
07C	GEO	3.0	0.00075	0.1	-0.03	295.0	65.00
08C	LEO	1.0	0.00500	0.2	0.10	130.4	-130.0

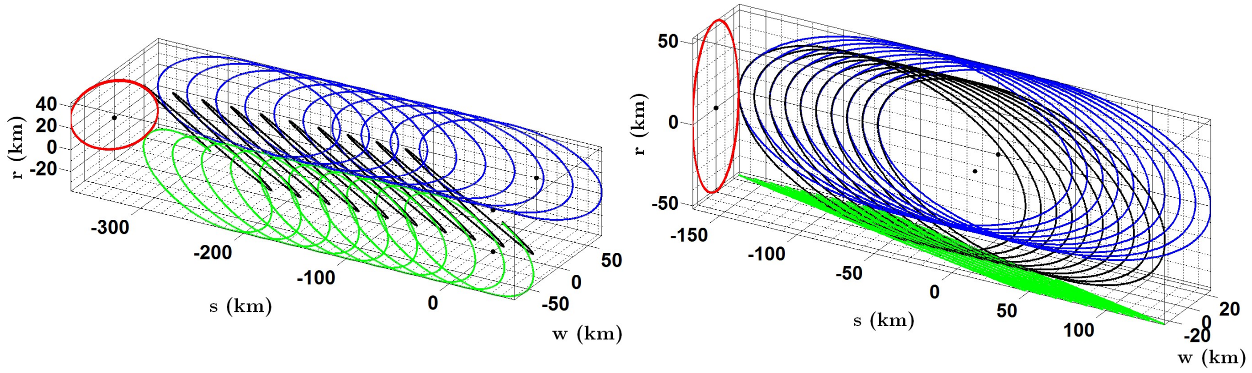


(a) Case 03C – GEO (Matching a and ω)

(b) Case 04C – LEO (Matching a and ω)

(c) Case 05C – GEO (Small Differentials)

(d) Case 06C – LEO (Small Differentials)



(e) Case 07C – GEO (Large $\delta\omega$)

(f) Case 08C – LEO (Large $\delta\omega$)

Figure 52: Test Case Depictions. (a) and (b) are idealized station-keeping cases at GEO and LEO, respectively. The LEO case exhibits some uncorrected natural drift due to the Earth's oblateness. (c) and (d) are small differential element cases at GEO and LEO, respectively. (e) and (f) have large differentials in argument of perigee while the other differential elements are small. The black curve is the actual trajectory. The red curve is a projection in the rw-plane. The green curve is a projection in the sw-plane. The blue curve is a projection in the rs-plane.

Implementation Notes

Truth data is generated with numerically integrated orbits in the full geopotential (20 × 20) with a sample rate of 10 Hz at LEO and 1 Hz at GEO. All other perturbations are ignored. No prior knowledge of the Target is assumed and zero-mean measurement noise is added to the angles with $\sigma = 10$ arc sec; otherwise, perfect knowledge of the Surveyor's inertial state is assumed for generating the $\boldsymbol{R}^{hi}$ rotation matrix. Errors in the reference KAM torus (typically less than 10 m RMS in the coordinates) introduce noise in the filter dynamics. GEO cases use numerically integrated reference trajectories instead of the KAM torus due to challenges fitting a torus to resonant orbits.

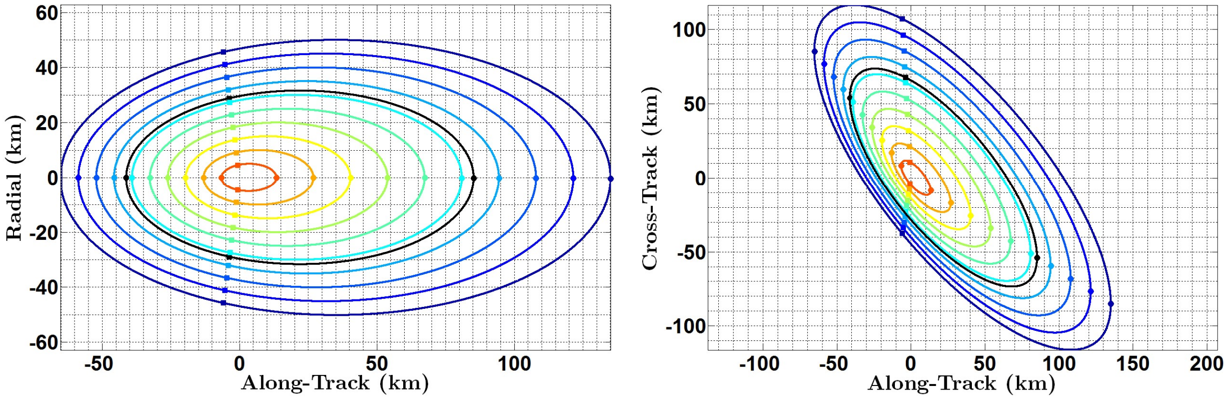
6.1 Case 03C – GEO, Matching Orbital Energy, Matching Perigee

This case demonstrates a station-keeping NMC trajectory at GEO, which requires an energy matching condition ($\delta a \approx 0$). Out-of-plane motion is induced from $\delta i = 0.1°$ and $\delta\Omega = -0.03°$. δi also causes an along-track drift, albeit very small due to the high altitude. The trajectory is offset in the along-track direction by the combination of $\delta\nu = 0.06°$ and $\delta\Omega$. The argument of perigee is coincident, so it does not contribute to the along-track shift in this case. The scale of the relative trajectory is dominated by $\delta e = 0.00075$ given the conditions of δa and $\delta\omega$. Although the differential elements seem quite small, it does not take very much at this altitude to distort the relative trajectory within the ranges of angles-only navigation. Table 13 indicates the COEs for the Surveyor, Target and uniformly sampled admissible hypotheses. It also indicates the magnitude of $\boldsymbol{\rho}_1$ for a sense of the relative trajectory's scale. Note that the semi-major axis and argument of perigee are the same for all hypotheses. This is a result of the logic that forces $\delta a = 0$ when $\Delta\alpha_T < 2.5°$ and the argument of perigee search in $\delta\omega = 10°$ increments about ω_s.

Table 13: Case 03C Orbital Elements for Truth & Hypotheses

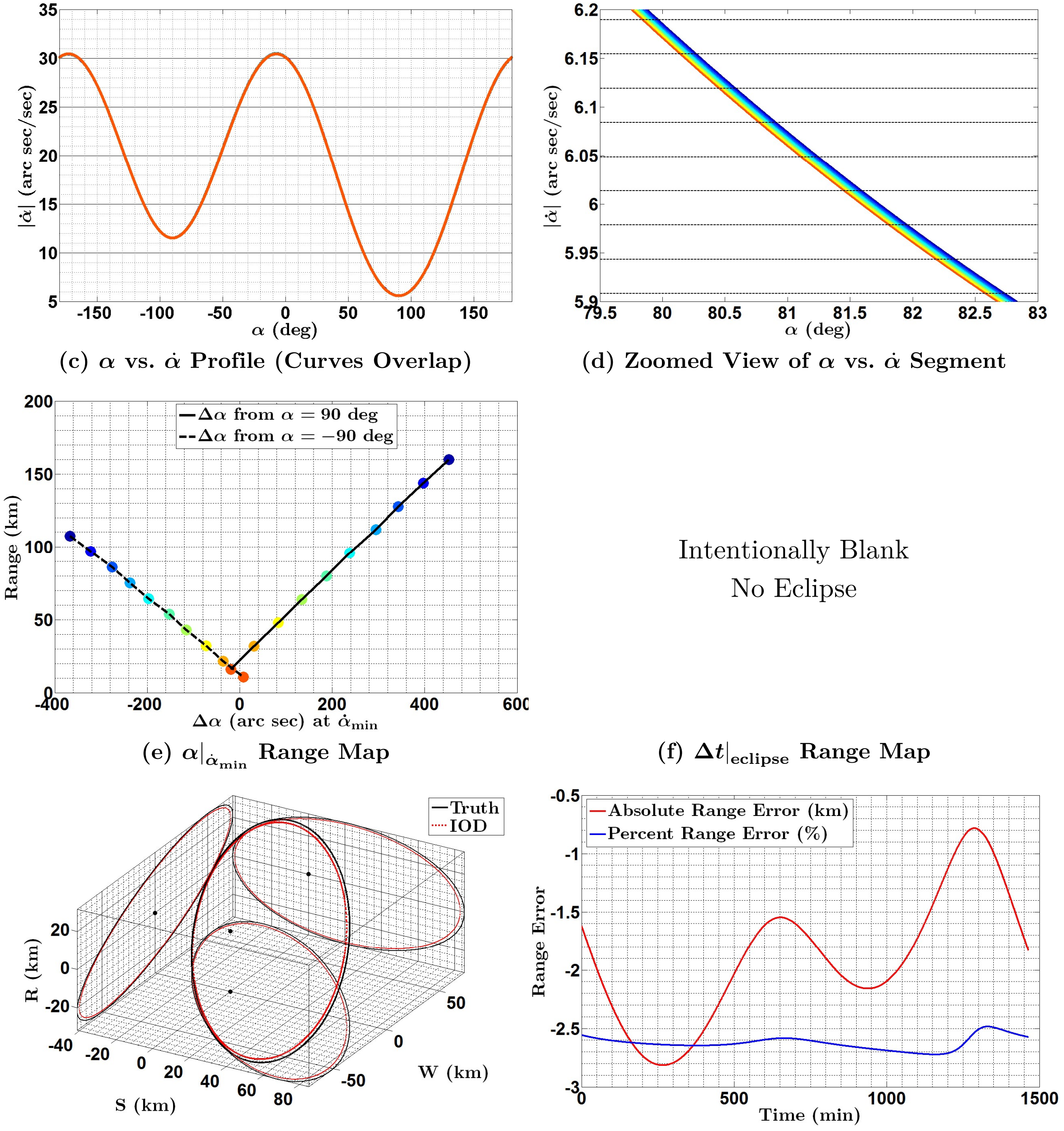
	$\lvert\boldsymbol{\rho}_1\rvert$ (km)	$\boldsymbol{a}$ (km)	$\boldsymbol{e}$ (nd)	$\boldsymbol{i}$ (°)	$\boldsymbol{\Omega}$ (°)	$\boldsymbol{\omega}$ (°)	$\boldsymbol{\nu}$ (°)
Surv	-	42166.15	0.000304	0.1929	87.10659	222.8015	0.000000
Targ	63.1942	42166.15	0.001054	0.2929	87.07659	222.8015	0.060000
H1	100.010	42166.15	0.001492	0.351008	87.06457	222.8015	0.089695
H2	90.0018	42166.15	0.001373	0.335188	87.06087	222.8015	0.088595
H3	79.9935	42166.15	0.001254	0.31937	87.05824	222.8015	0.086435
H4	69.9914	42166.15	0.001135	0.303554	87.05682	222.8015	0.083075
H5	59.9881	42166.15	0.001016	0.287741	87.05682	222.8015	0.078285
H6	49.9882	42166.15	0.000897	0.271929	87.05850	222.8015	0.071835
H7	39.9890	42166.15	0.000779	0.256119	87.06217	222.8015	0.063405
H8	29.9904	42166.15	0.000660	0.240311	87.06820	222.8015	0.052605
H9	19.9949	42166.15	0.000541	0.224506	87.07712	222.8015	0.038945
H10	9.9980	42166.15	0.000423	0.208702	87.08957	222.8015	0.021755

The IOD results are summarized in a sequence of eight plots shown in Figure 53. The hypotheses show good scalability agreement with the truth as indicated by the trajectory projections in Figure 53(a) and Figure 53(b). When $\delta a \approx 0$, the skewness factor can be used to identify the region of the orbit with the best range observability. Since the trajectory skews right (skewness factor is $d = 0.6534$), range observability is better in the $+\alpha$ domain surrounding the global $\dot{\alpha}_{\text{min}}$. The better observability is justified by the larger $\Delta\alpha$ spread in this region as shown in the $\alpha|_{\dot{\alpha}_{\text{min}}}$ range map in Figure 53(e). Range resolvability in the $+\alpha$ domain ($\text{m} = 0.3049$) is only slightly worse than in the $-\alpha$ domain ($\text{m} = 0.2579$), so we sacrifice range resolution for better observability. A set of 200 $\dot{\alpha}$ waypoints surrounding the global $\dot{\alpha}_{\text{min}}$ are used to map range and scale the IOD COEs. Several of them are shown in Figure 53(d) as indicated by the dashed lines. There is no eclipse at GEO during the time of this scenario, so photometry is not used for scaling the COEs. The IOD solution is within 2.7% of the true range as shown in Figure 53(h). The accuracy of the IOD solution is also conveyed by the differential orbital elements in Table 14.



(a) RS-plane Hypotheses Projection

(b) SW-plane Hypotheses Projection



(c) α vs. $\dot{\alpha}$ Profile (Curves Overlap)

(d) Zoomed View of α vs. $\dot{\alpha}$ Segment

(e) $\alpha|_{\dot{\alpha}_{\min}}$ Range Map

(f) $\boldsymbol{\Delta t}|_{\text{eclipse}}$ Range Map

(g) Trajectories & Projections

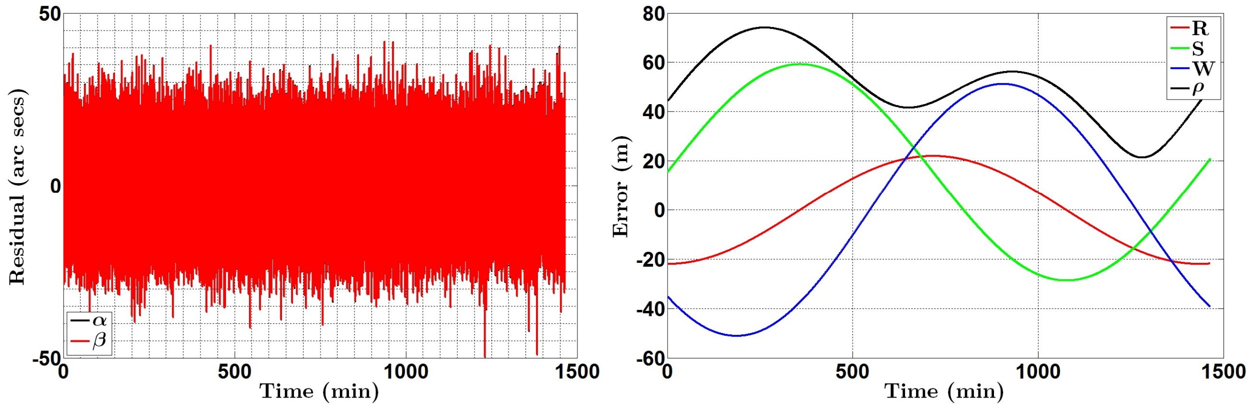
(h) Range Error

Figure 53: Case 03C Initial Orbit Determination Results. A set of 10 admissible hypothesis trajectories are projected in the rs-plane in (a) and the sw-plane in (b) along with the truth (the black curve). The location of minimum angular rate is indicated by the dots and the location of the maximum angular rate is indicated by the squares. The α vs. $\dot{\alpha}$ curves are plotted in (c), but practically overlap. A zoomed view of a small segment is

shown in (d) where the dashed lines are $\dot{\alpha}$ waypoints. The $\alpha|_{\dot{\alpha}_{\min}}$ range map is shown in (e) and the eclipse range map is shown in (f). The colored dots correspond to the colors used to represent the hypotheses in (a) and (b). The IOD solution (dashed red curve) is plotted with the truth (solid black curve) in (g) with their projections in all three directions to gain perspective of the in-plane and out-of-plane motion. The black dots indicate the location of the Surveyor for the actual and projected trajectories. The range error is plotted in (h) as an absolute error (red curve) and as a percentage of the truth (blue curve).

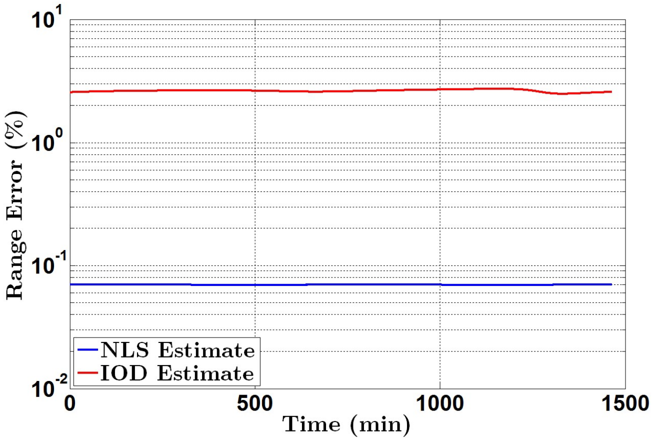
Estimation with the Numerical $\boldsymbol{A}$ Matrix & Integrated Reference Orbit

The NLS filter results are shown in Figure 54 using the numerical $\boldsymbol{A}$ matrix for local motion dynamics and a numerically integrated IOD reference orbit (recall that the torus could not be generated for this case). The root mean square (RMS) of the angle residuals are 10.02 arc sec in both angles indicating an excellent fit given the uncertainty in the angles ($\sigma_{\alpha,\beta} = 10$ arc sec). The accuracy of the estimate in Hill's frame is shown in Figure 54(b) where the range error peaks at 74 m. The percent range error holds relatively steady at ~0.07% of the true range, which is substantially better than the IOD solution and the 1% rule of thumb for RPOs. The accuracy of the NLS solution is also conveyed by the differential orbital elements in Table 14.



(a) Angle Residuals

(b) Coordinate & Range Errors



(c) Percent Range Error

Figure 54: Case 03C NLS Results (Numerical $\boldsymbol{A}$ Matrix & Integrated Ref. Orbit). The angle residuals are shown in (a) where α is indicated by the black curve (typically obscured by β) and β is indicated by the red curve. The relative position coordinate and range errors are shown in (b) where the radial direction is indicated in red, the along-track direction is indicated in green, the cross-track direction is indicated in blue and the range is indicated in black. Finally, the percent range error is presented in (c). The red curve designates the IOD solution and the blue curve designates the NLS solution.

Table 14: Case 03C Differential Orbital Elements for Truth, IOD & NLS

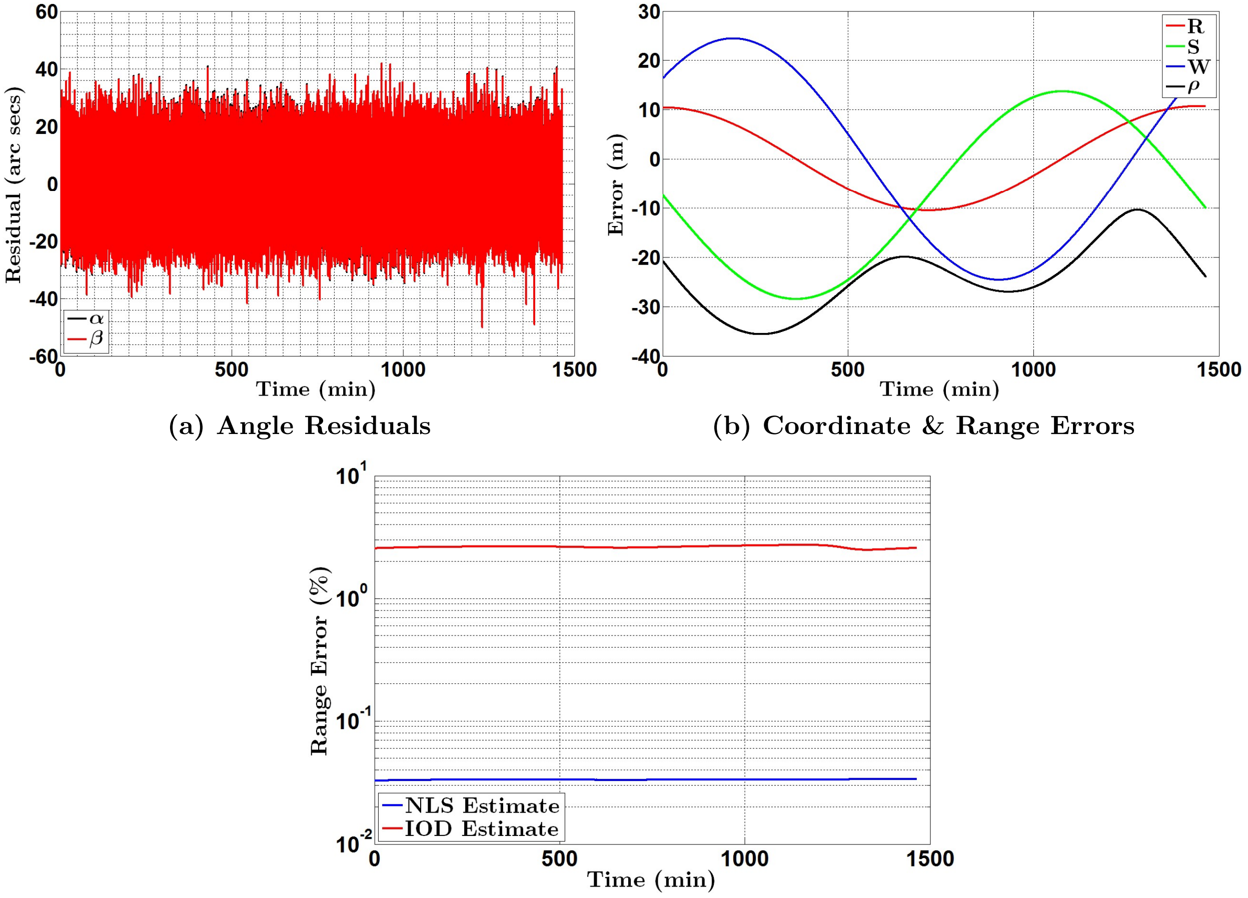
	δa (km)	δe (nd)	δi (°)	$\delta\Omega$ (°)	$\delta\omega$ (°)	$\delta\nu$ (°)
Truth	0.00000	0.00075	0.10000	-0.03000	0.00000	0.06000
IOD	0.00000	0.00073	0.09731	-0.05154	0.00000	0.08080
NLS†	-0.00235	0.00075	0.10006	-0.09802	-6.75E-2	0.19996
NLS‡	-0.00188	0.00075	0.09997	-0.03015	6.06E-5	0.06427

† Numerical $\boldsymbol{A}$ matrix & numerically integrated reference orbit

‡ Two-Body $\boldsymbol{A}$ matrix & numerically integrated reference orbit

Estimation with the Two-Body $\boldsymbol{A}$ matrix & Integrated Reference Orbit

The analytical two-body $\boldsymbol{A}$ matrix was also tested against the numerical $\boldsymbol{A}$ matrix to assess whether the added complexity and computational cost is warranted. The results are shown in Figure 55. The RMS of the angle residuals have not changed—10.02 arc sec in both angles. The accuracy of the estimate in Hill's frame is shown in Figure 55(b) where the range error peaks at -36 m. The percent range error holds relatively steady at ~0.033% of the true range, which is half of the error using the numerical $\boldsymbol{A}$ matrix. These results suggest that the expanded geopotential is overkill in the $\boldsymbol{A}$ matrix. The reference orbit already contains the global effects of the full geopotential including J_2 and the sectoral terms that cause a longitudinal drift at GEO. The local motion seems to be well modelled with two-body equations of variation for this GEO case. We should be careful not to extrapolate these results. Figure 52(a) shows virtually no along-track drift throughout the duration of 10 orbits, so it is no surprise that the Keplerian $\boldsymbol{A}$ matrix is sufficient.



(a) Angle Residuals

(b) Coordinate & Range Errors

(c) Percent Range Error

Figure 55: Case 03C NLS Results (Two-Body A Matrix & Integrated Ref. Orbit). The angle residuals are shown in (a) where α is indicated by the black curve (typically obscured by β) and β is indicated by the red curve. The relative position coordinate and range errors are shown in (b) where the radial direction is indicated in red, the along-track direction is indicated in green, the cross-track direction is indicated in blue and the range is indicated in black. Finally, the percent range error is presented in (c). The red curve designates the IOD solution and the blue curve designates the NLS solution.

6.2 Case 04C – LEO, Matching Orbital Energy, Matching Perigee

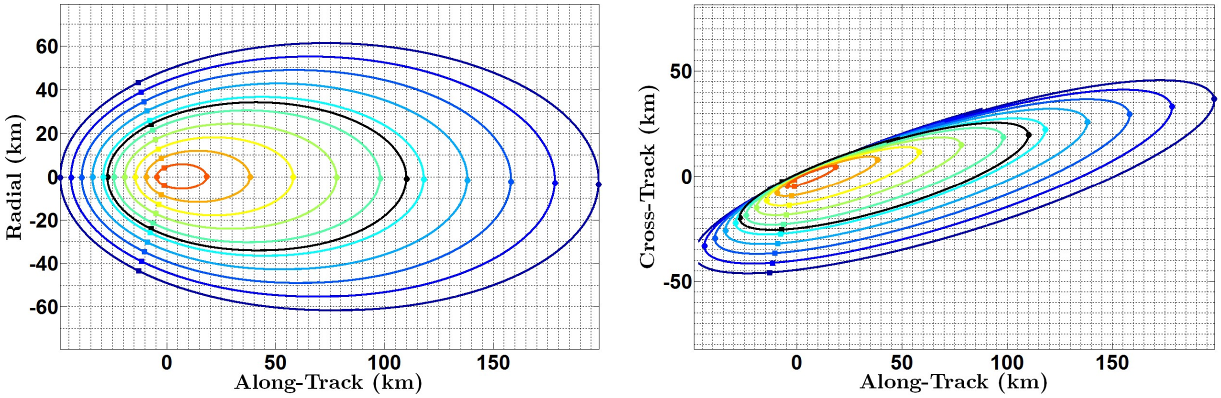
This case demonstrates a persistent NMC trajectory at LEO, which requires an energy matching condition ($\delta a \approx 0$). Out-of-plane motion is induced from $\delta i = 0.2°$ and $\delta\Omega = 0.1°$. δi also causes an along-track drift. The trajectory is shifted in the along-track direction by $\delta\nu = 0.3°$ and $\delta\Omega$. The argument of perigee is coincident, so it does not contribute to the along-track shift in this case. The scale of the relative trajectory is dominated by $\delta e = 0.005$ given the conditions of δa and $\delta\omega$. Table 15 indicates the COEs for the Surveyor, Target and uniformly sampled admissible hypotheses. It also indicates the magnitude of $\boldsymbol{\rho}_1$ for a sense of the relative trajectory's scale.

Table 15: Case 04C Orbital Elements for Truth & Hypotheses

	$\|\boldsymbol{\rho}_1\|$ (km)	$\boldsymbol{a}$ (km)	$\boldsymbol{e}$ (nd)	$\boldsymbol{i}$ (°)	$\boldsymbol{\Omega}$ (°)	$\boldsymbol{\omega}$ (°)	$\boldsymbol{\nu}$ (°)
Surv	-	6789.086	0.001601	51.75242	72.49305	60.69588	0.000000
Targ	55.77359	6789.086	0.006601	51.95242	72.59305	60.69588	0.300000
H1	99.97613	6789.086	0.010609	52.10731	72.68847	60.69588	0.536085
H2	89.86799	6789.086	0.009695	52.07176	72.66943	60.69588	0.480095
H3	79.76535	6789.086	0.008783	52.03618	72.65059	60.69588	0.424355
H4	69.66590	6789.086	0.007872	52.00058	72.63194	60.69588	0.368835
H5	59.57155	6789.086	0.006963	51.96495	72.6135	60.69588	0.313555
H6	49.48004	6789.086	0.006054	51.92929	72.59526	60.69588	0.258485
H7	39.39336	6789.086	0.005148	51.89362	72.57722	60.69588	0.203645

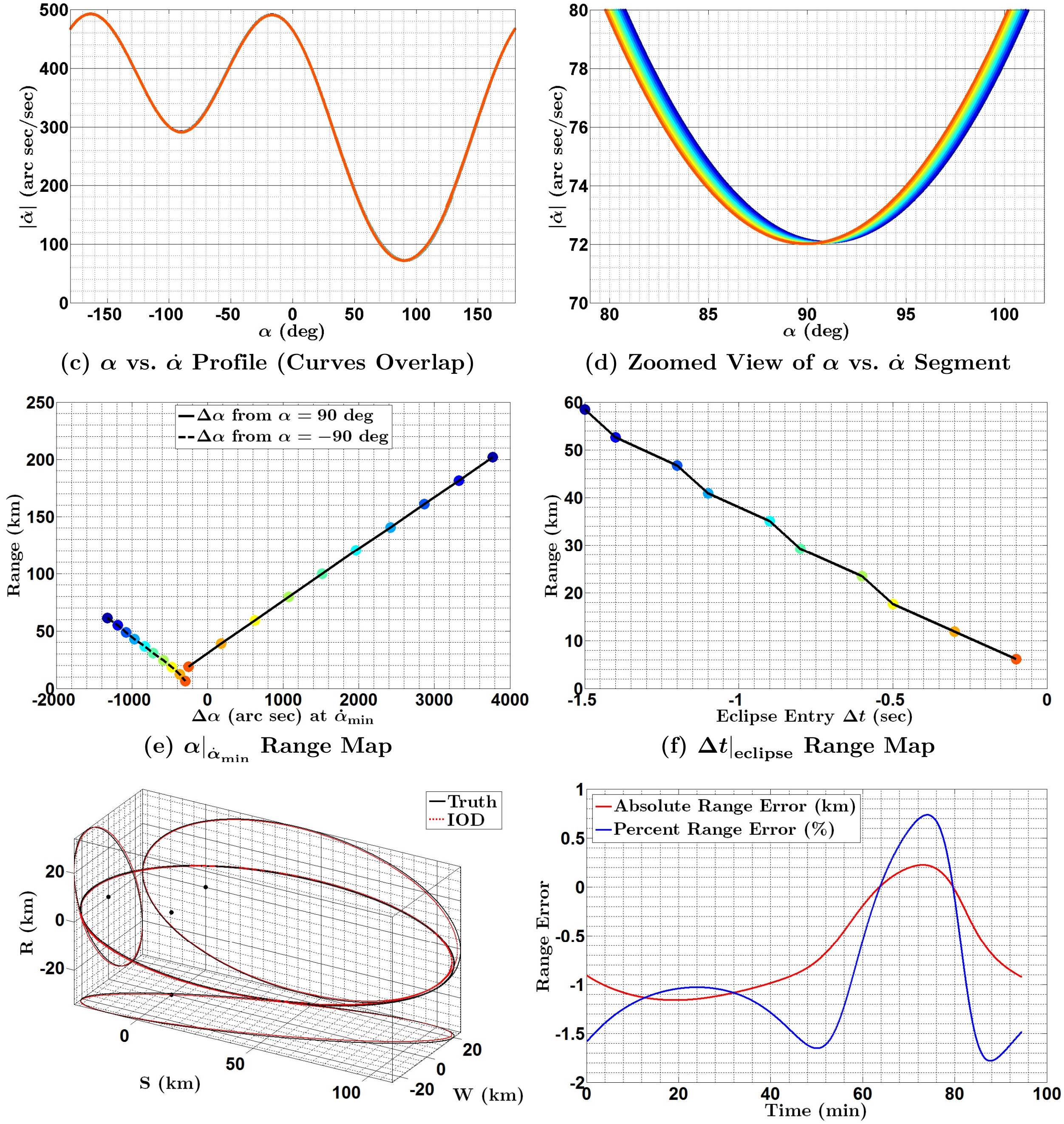
H8	29.31031	6789.086	0.004242	51.85791	72.55939	60.69588	0.149015
H9	19.23198	6789.086	0.003338	51.82218	72.54176	60.69588	0.094595
H10	9.162617	6789.086	0.002435	51.78643	72.52434	60.69588	0.040375

The IOD results are summarized in a sequence of eight plots shown in Figure 56. The hypotheses show good scalability agreement with the truth as indicated by the trajectory projections in Figure 56(a) and Figure 56(b). Since the trajectory skews right (skewness factor is $d = 0.3966$), range observability is better in the $+\alpha$ domain surrounding the global $\dot{\alpha}_{\min}$. The selection is justified by the larger $\Delta\alpha$ spread in this region as shown in the $\alpha|_{\dot{\alpha}_{\min}}$ range map in Figure 56(e). Range resolvability in the $+\alpha$ domain ($\mathrm{m} = 0.0455$) is comparable to the $-\alpha$ domain ($\mathrm{m} = 0.0531$). A set of 200 $\dot{\alpha}$ waypoints surrounding the global $\dot{\alpha}_{\min}$ are used to map range and scale the IOD COEs. This is made possible by the good structural coherency in the α vs. $\dot{\alpha}$ curves shown in Figure 56(d). The eclipse time differential is too short to be of any use for scaling as shown in Figure 56(f), so only the $\alpha|_{\dot{\alpha}}$ range maps are used. The IOD solution is within 1.8% of the true range as shown in Figure 56(h). The accuracy of the IOD solution is also conveyed by the differential orbital elements in Table 16.



(a) RS-plane Hypotheses Projection

(b) SW-plane Hypotheses Projection



(c) α vs. $\dot{\alpha}$ Profile (Curves Overlap)

(d) Zoomed View of α vs. $\dot{\alpha}$ Segment

(e) $\alpha|_{\dot{\alpha}_{\text{min}}}$ Range Map

(f) $\Delta t|_{\text{eclipse}}$ Range Map

(g) Trajectories & Projections

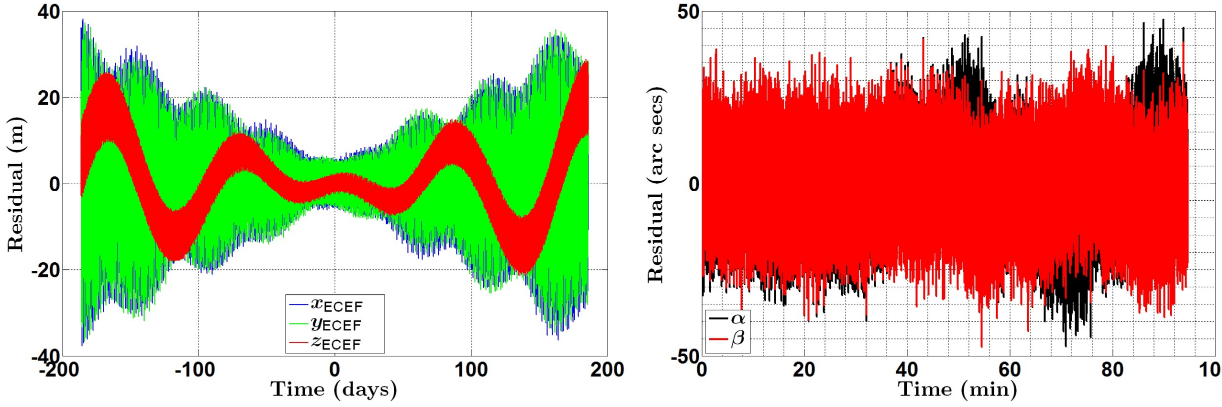
(h) Range Error

Figure 56: Case 04C Initial Orbit Determination Results. A set of 10 admissible hypothesis trajectories are projected in the rs-plane in (a) and the sw-plane in (b) along with the truth (the black curve). The location of minimum angular rate is indicated by the dots and the location of the maximum angular rate is indicated by the squares. The α vs. $\dot{\alpha}$ curves are plotted in (c), but practically overlap. A zoomed view of a small segment near

$\dot{\alpha}_{\min}$ is shown in (d). The $\alpha|_{\dot{\alpha}_{\min}}$ range map is shown in (e) and the eclipse range map is shown in (f). The colored dots correspond to the colors used to represent the hypotheses in (a) and (b). The IOD solution (dashed red curve) is plotted with the truth (solid black curve) in (g) with their projections in all three directions to gain perspective of the in-plane and out-of-plane motion. The black dots indicate the location of the Surveyor for the actual and projected trajectories. The range error is plotted in (h) as an absolute error (red curve) and as a percentage of the truth (blue curve).

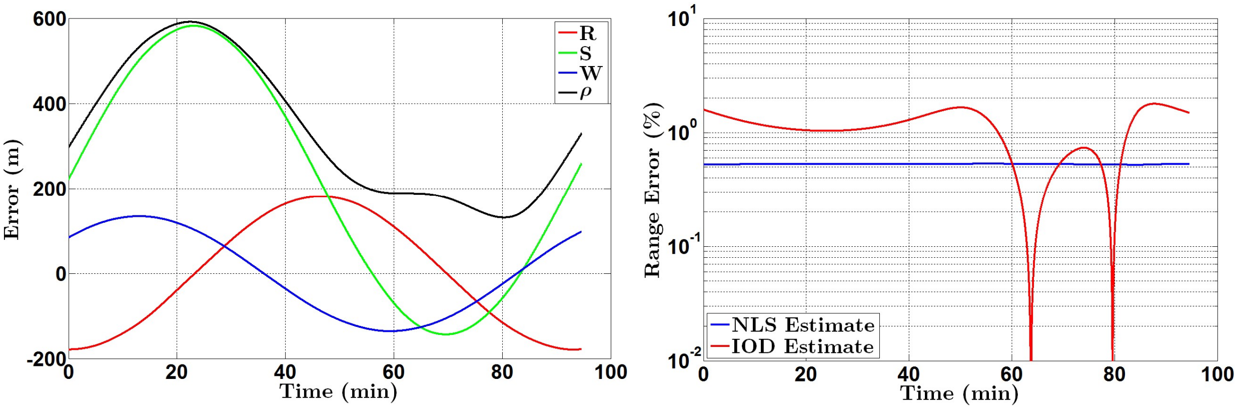
Estimation with the Numerical $\boldsymbol{A}$ Matrix & Torus Reference Orbit

The NLS filter results are shown in Figure 57 using the numerical $\boldsymbol{A}$ matrix for local motion dynamics and an IOD reference torus. The RMS of the angle residuals are 11.25 and 10.37 arc sec in α and β, respectively. The RMS of the torus native coordinates are 9.19, 9.19 and 8.37 m in x, y and z, respectively. This is a decent torus fit, but it does affect the accuracy of the estimate as process noise. The accuracy of the estimate in Hill's frame is shown in Figure 57(c) where the range error peaks at 591 m. The percent range error holds relatively steady at ~0.53% of the true range, which is better than the IOD solution and the 1% rule of thumb for RPOs. The accuracy of the NLS solution is also conveyed by the differential orbital elements in Table 16.



(a) Torus Accuracy

(b) Angle Residuals



(c) Coordinate & Range Errors

(d) Percent Range Error

Figure 57: Case 04C NLS Results (Numerical *A* Matrix & Torus Ref. Orbit). The ECEF coordinate residuals from the torus orbit fitting process are shown in (a) where x, y and z are indicated by blue, green and red, respectively. The angle residuals are shown in (b) where α is indicated by the black curve (typically obscured by β) and β is indicated by the red curve. The relative position coordinate and range errors are shown in (c) where the radial direction is indicated in red, the along-track direction is indicated in green, the cross-track direction is indicated in blue and the range is indicated in black. Finally, the percent range error is presented in (d). The red curve designates the IOD solution and the blue curve designates the NLS solution.

Table 16: Case 04C Differential Orbital Elements for Truth, IOD & NLS

	δa (km)	δe (nd)	δi (°)	$\delta\Omega$ (°)	$\delta\omega$ (°)	$\delta\nu$ (°)
Truth	0.00000	0.00500	0.20000	0.10000	0.00000	0.30000
IOD	0.00000	0.00494	0.19604	0.11199	0.00000	0.28804
NLS*	-0.29060	0.00498	0.19603	0.09958	-0.73820	1.05027
NLS†	-0.00081	0.00500	0.20009	0.10004	-0.00057	0.30728
NLS⋆	0.00917	0.00501	0.20052	0.10031	-0.01370	0.32104
NLS‡	-0.00232	0.00499	0.19964	0.09985	-0.00665	0.31269

* Numerical *A* matrix & torus reference orbit

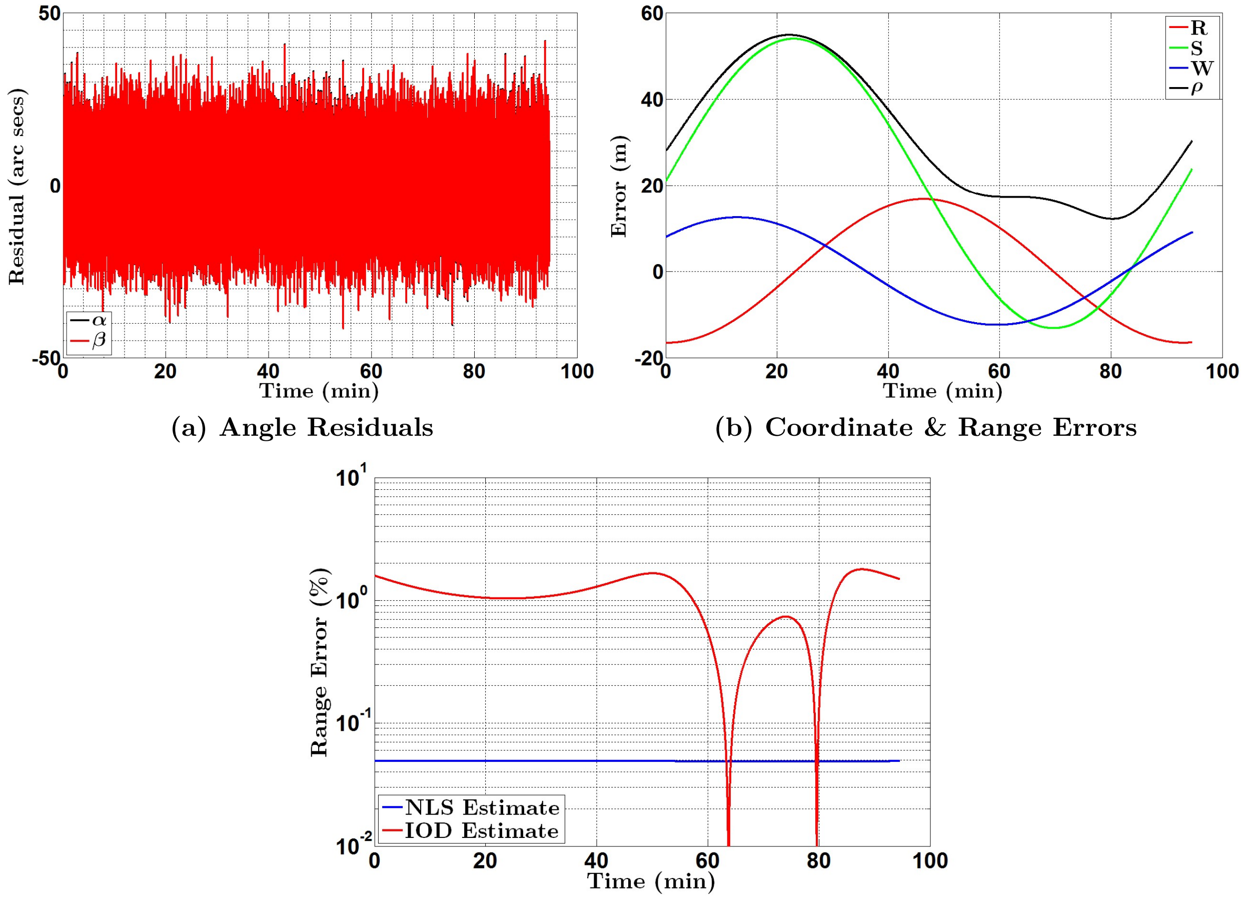
† Numerical *A* matrix & numerically integrated reference orbit

⋆ Two-Body *A* matrix & torus reference orbit

‡ Two-Body *A* matrix & numerically integrated reference orbit

Estimation with the Numerical $\boldsymbol{A}$ Matrix & Integrated Reference Orbit

The previous result is not as good as the Case 03C GEO results that used a numerically integrated reference orbit, so how much of an improvement can be made with a numerically integrated reference orbit in this LEO case? The NLS filter results are shown in Figure 58 using the numerical $\boldsymbol{A}$ matrix for local motion dynamics and a numerically integrated reference orbit. The RMS of the angle residuals are 10.02 and 10.03 arc sec in α and β, respectively. The accuracy of the estimate in Hill's frame is shown in Figure 58(b) where the range error peaks at 55 m. The percent range error holds relatively steady at ~0.05% of the true range, which is better than the last trial with the torus by an order of magnitude.



(a) Angle Residuals

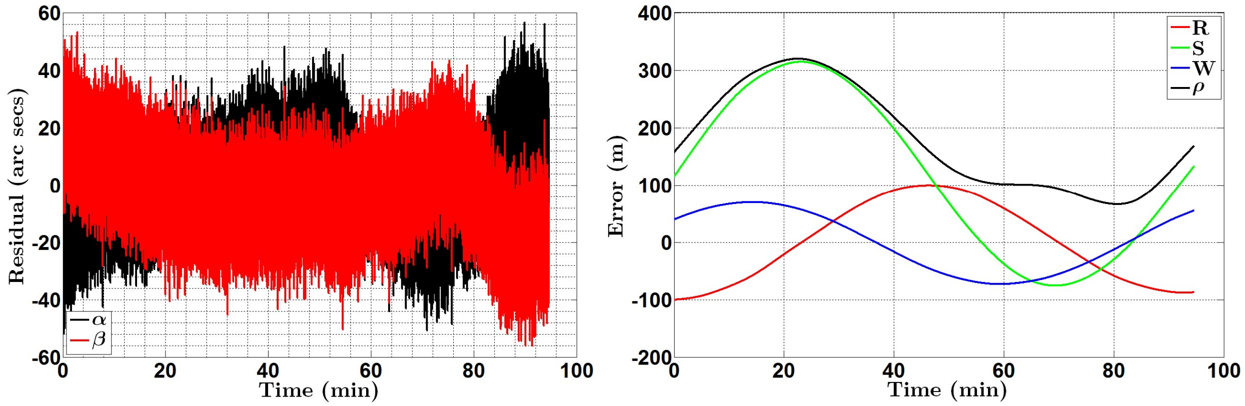
(b) Coordinate & Range Errors

(c) Percent Range Error

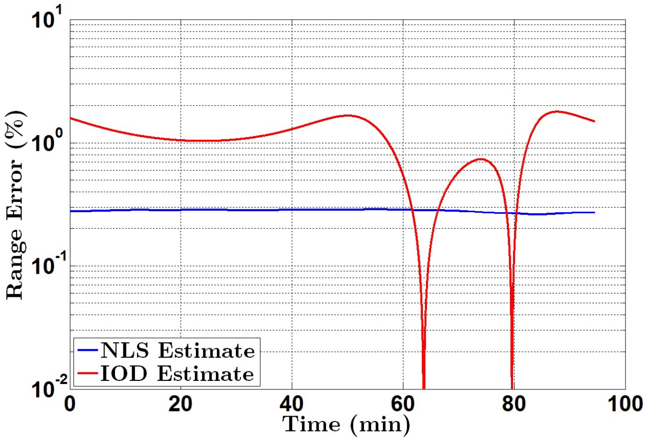
Figure 58: Case 04C NLS Results (Numerical *A* Matrix & Integrated Ref. Orbit). The angle residuals are shown in (a) where α is indicated by the black curve (obscured by β) and β is indicated by the red curve. The relative position coordinate and range errors are shown in (b) where the radial direction is indicated in red, the along-track direction is indicated in green, the cross-track direction is indicated in blue and the range is indicated in black. Finally, the percent range error is presented in (c). The red curve designates the IOD solution and the blue curve designates the NLS solution.

Estimation with the Two-Body *A* matrix & Torus Reference Orbit

Another variation in the dynamics model uses the analytical two-body $\boldsymbol{A}$ matrix for local motion dynamics and an IOD reference torus. The results are shown in Figure 59 where we have left out the torus construction residual since it has not changed. The RMS of the angle residuals are 13.61 and 13.87 arc sec in α and β, respectively. The accuracy of the estimate in Hill's frame is shown in Figure 59(b) where the range error peaks at 320 m. The percent range error holds relatively steady at ~0.28% of the true range which is about half of the error using the numerical $\boldsymbol{A}$ matrix. Once again, these results suggest that the local motion is well modelled with the two-body equations of variation.



(a) Angle Residuals **(b) Coordinate & Range Errors**

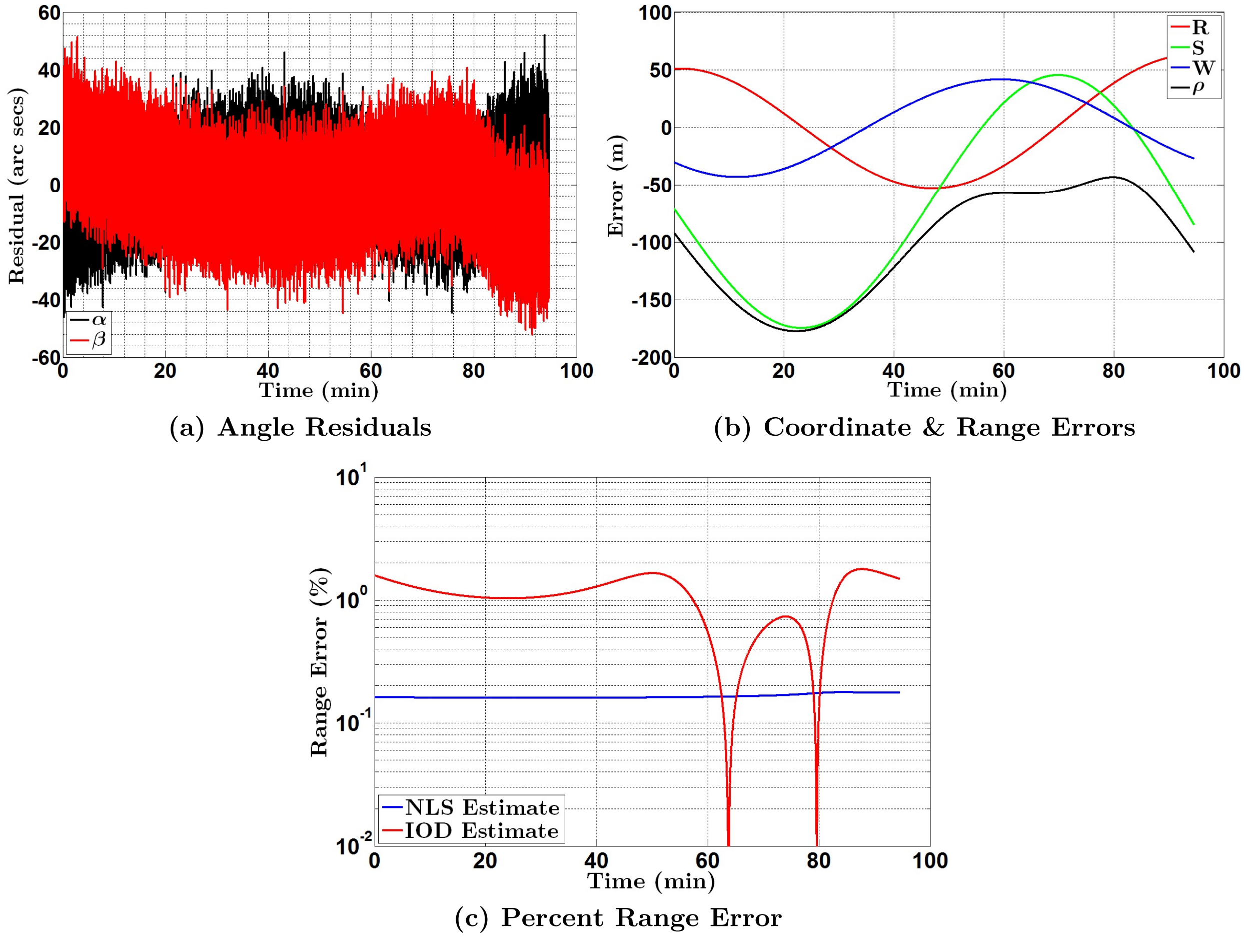


(c) Percent Range Error

Figure 59: Case 04C NLS Results (Two-Body A Matrix & Torus Ref. Orbit). The angle residuals are shown in (a) where α is indicated by the black curve and β is indicated by the red curve. The relative position coordinate and range errors are shown in (b) where the radial direction is indicated in red, the along-track direction is indicated in green, the cross-track direction is indicated in blue and the range is indicated in black. Finally, the percent range error is presented in (c). The red curve designates the IOD solution and the blue curve designates the NLS solution.

Estimation with the Two-Body A matrix & Integrated Reference Orbit

One final test is performed using the analytical two-body $\boldsymbol{A}$ matrix and a numerically integrated reference orbit. The results are shown in Figure 60. The RMS of the angle residuals are 11.17 and 12.57 arc sec in α and β, respectively. The accuracy of the estimate in Hill's frame is shown in Figure 59(b) where the range error peaks at -178 m. The percent range error holds relatively steady at ~0.16% of the true range, which is not quite as good as the numerical $\boldsymbol{A}$ matrix with the integrated reference orbit.



(a) Angle Residuals

(b) Coordinate & Range Errors

(c) Percent Range Error

Figure 60: Case 04C NLS Results (Two-Body A Matrix & Integrated Ref. Orbit). The angle residuals are shown in (a) where α is indicated by the black curve and β is indicated by the red curve. The relative position coordinate and range errors are shown in (b) where the radial direction is indicated in red, the along-track direction is indicated in green, the cross-track direction is indicated in blue and the range is indicated in black. Finally, the percent range error is presented in (c). The red curve designates the IOD solution and the blue curve designates the NLS solution.

6.3 Case 05C – GEO, Small Differential Elements

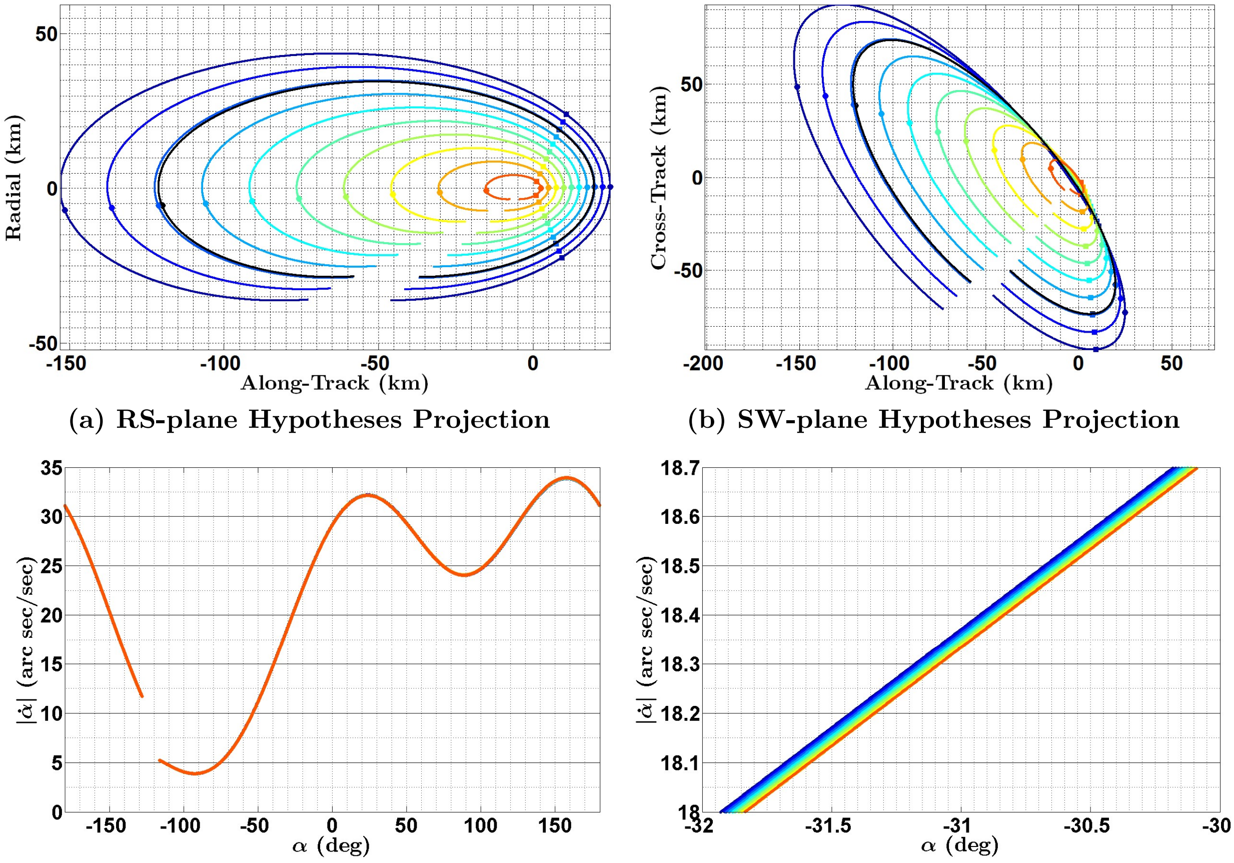
This case demonstrates a drifting NMC trajectory at GEO with small differentials in all orbital elements. A moderate amount of drifting is caused by $\delta a = 3$ km. Out-of-plane motion is induced from $\delta i = 0.1°$ and $\delta\Omega = -0.03°$ and the trajectory is shifted in the negative along-track direction by the combination of $\delta\omega = 0.1°$, $\delta\nu = -0.12°$ and $\delta\Omega$. The scale of the relative trajectory is dominated by $\delta e = 0.00075$ with very minor influence from δa and $\delta\omega$. Table 17 indicates the COEs for the Surveyor, Target and uniformly sampled admissible hypotheses. It also indicates the magnitude of $\boldsymbol{\rho}_1$ for a sense of the relative trajectory's scale.

Table 17: Case 05C Orbital Elements for Truth & Hypotheses

	$\|\boldsymbol{\rho}_1\|$ (km)	$\boldsymbol{a}$ (km)	$\boldsymbol{e}$ (nd)	$\boldsymbol{i}$ (°)	$\boldsymbol{\Omega}$ (°)	$\boldsymbol{\omega}$ (°)	$\boldsymbol{\nu}$ (°)
Surv	-	42166.15	0.000304	0.192900	87.10659	222.8015	0.000000
Targ	79.2197	42169.15	0.001054	0.292900	87.07659	222.9015	-0.120000
H1	100.236	42169.93	0.001251	0.318772	87.00900	222.8015	0.034775
H2	90.2057	42169.55	0.001157	0.306161	87.01213	222.8015	0.037945
H3	80.1723	42169.17	0.001062	0.293581	87.02098	222.8015	0.035375
H4	70.1468	42168.79	0.000967	0.280974	87.02506	222.8015	0.037585
H5	60.1216	42168.41	0.000872	0.268385	87.03301	222.8015	0.035905
H6	50.0920	42168.04	0.000777	0.255816	87.04542	222.8015	0.029775
H7	40.0728	42167.66	0.000683	0.243212	87.05106	222.8015	0.030415
H8	30.0542	42167.28	0.000588	0.230626	87.06139	222.8015	0.026365
H9	20.0336	42166.9	0.000493	0.21806	87.0772	222.8015	0.016825
H10	10.0141	42166.53	0.000399	0.205486	87.09247	222.8015	0.007845

The IOD results are summarized in a sequence of eight plots shown in Figure 61. The hypotheses show good scalability agreement with the truth as indicated by the trajectory projections in Figure 61(a) and Figure 61(b). Our instinct from the last two cases is that range observability is better in the $-\alpha$ domain surrounding the global $\dot{\alpha}_{\min}$ since the trajectory skews left (skewness factor is $d = 1.7209$), but that is not the case here. The differential semi-major axis distorts the angle offsets so that there is better range resolvability and observability in the $+\alpha$ domain in this case. The $\alpha|_{\dot{\alpha}_{\min}}$ range map in Figure 61(e) shows a larger $\Delta\alpha$ spread in this region. Range resolvability in the $+\alpha$

domain ($\mathrm{m} = 0.0937$) is three times better than the $-\alpha$ domain ($\mathrm{m} = 0.3075$). The structural coherency of the α vs $\dot{\alpha}$ arcs surrounding the local $\dot{\alpha}_{\text{min}}$ in the $+\alpha$ domain is poor due to close proximity to the $\dot{\alpha}_{\text{max}}$ points indicated in Figure 61(a). As a result, only the local $\dot{\alpha}_{\text{min}}$ is used to map range and scale the IOD COEs. It was discovered during post-processing that there is good structural coherency in other parts of the orbit as shown in Figure 61(d). Proteus is presently designed to look for coherency on arcs near $\dot{\alpha}_{\text{min}}$, so future work should explore how differential elements change the coherent structure in the α vs $\dot{\alpha}$ arcs for exploitability. There is no eclipse at GEO during the time of this scenario, so photometry is not used for scaling the COEs. The IOD solution is within 2.1% of the true range as shown in Figure 61(h). The accuracy of the IOD solution is also conveyed by the differential orbital elements in Table 18.

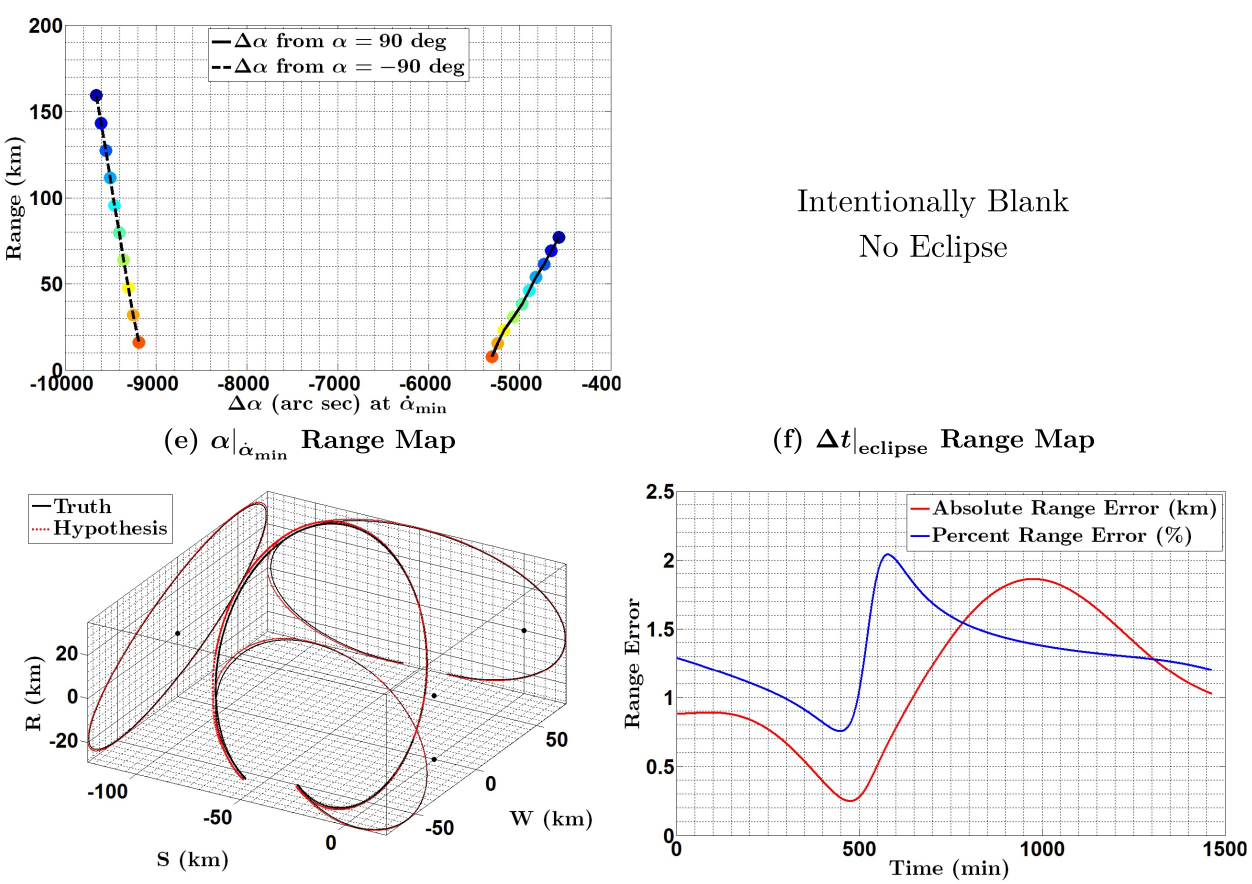


(a) RS-plane Hypotheses Projection

(b) SW-plane Hypotheses Projection

(c) α vs. $\dot{\alpha}$ Profile (Curves Overlap)

(d) Zoomed View of α vs. $\dot{\alpha}$ Segment



(e) $\alpha|_{\dot{\alpha}_{\min}}$ Range Map

(f) $\Delta t|_{\text{eclipse}}$ Range Map

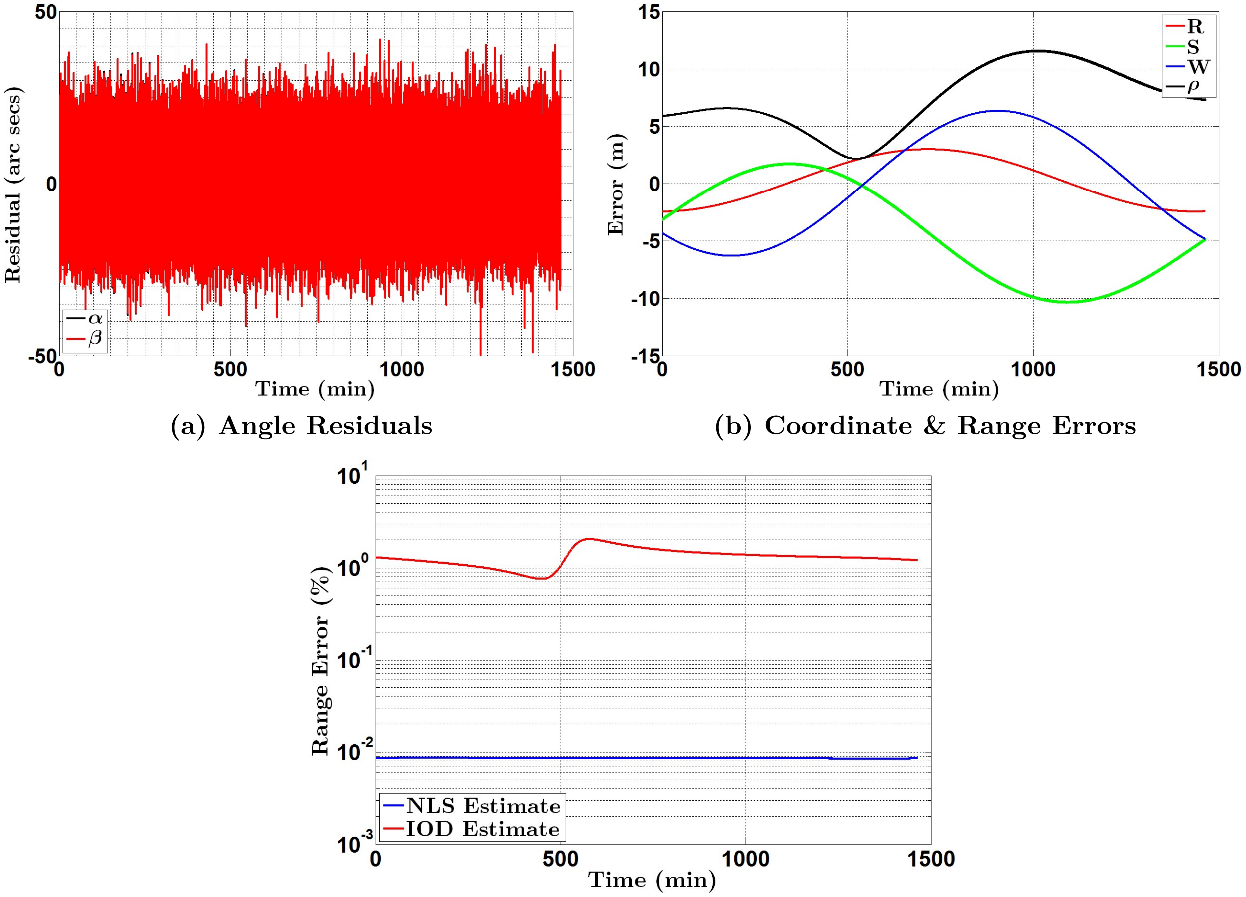
(g) Trajectories & Projections

(h) Range Error

Figure 61: Case 05C Initial Orbit Determination Results. A set of 10 admissible hypothesis trajectories are projected in the rs-plane in (a) and the sw-plane in (b) along with the truth (the black curve). The location of minimum angular rate is indicated by the dots and the location of the maximum angular rate is indicated by the squares. The α vs. $\dot{\alpha}$ curves are plotted in (c), but practically overlap. A zoomed view of a small segment exhibiting structural coherency is shown in (d). The $\alpha|_{\dot{\alpha}_{\min}}$ range map is shown in (e) and the eclipse range map is shown in (f). The colored dots correspond to the colors used to represent the hypotheses in (a) and (b). The IOD solution (dashed red curve) is plotted with the truth (solid black curve) in (g) with their projections in all three directions to gain perspective of the in-plane and out-of-plane motion. The black dots indicate the location of the Surveyor for the actual and projected trajectories. The range error is plotted in (h) as an absolute error (red curve) and as a percentage of the truth (blue curve).

Estimation with the Numerical *A* Matrix & Integrated Reference Orbit

The NLS filter results are shown in Figure 62 using the numerical $\boldsymbol{A}$ matrix for local motion dynamics and a numerically integrated IOD reference orbit. The RMS of the angle residuals are 10.02 arc sec in both angles indicating an excellent fit given the uncertainty in the angles ($\sigma_{\alpha,\beta} = 10$ arc sec). The accuracy of the estimate in Hill's frame is shown in Figure 62(b) where the range error peaks at 12 m. The percent range error holds relatively steady at ~0.0085% of the true range, which is substantially better than the IOD solution and the 1% rule of thumb for RPOs. The accuracy of the NLS solution is also conveyed by the differential orbital elements in Table 18.



(a) Angle Residuals **(b) Coordinate & Range Errors**

(c) Percent Range Error

Figure 62: Case 05C NLS Results (Numerical *A* Matrix & Integrated Ref. Orbit). The angle residuals are shown in (a) where α is indicated by the black curve (typically obscured by β) and β is indicated by the red curve. The relative position coordinate and range errors are shown in (b) where the radial direction is indicated in red, the along-track direction is indicated in green, the cross-track direction is indicated in blue and the range is indicated in black. Finally, the percent range error is presented in (c). The red curve designates the IOD solution and the blue curve designates the NLS solution.

Table 18: Case 05C Differential Orbital Elements for Truth, IOD & NLS

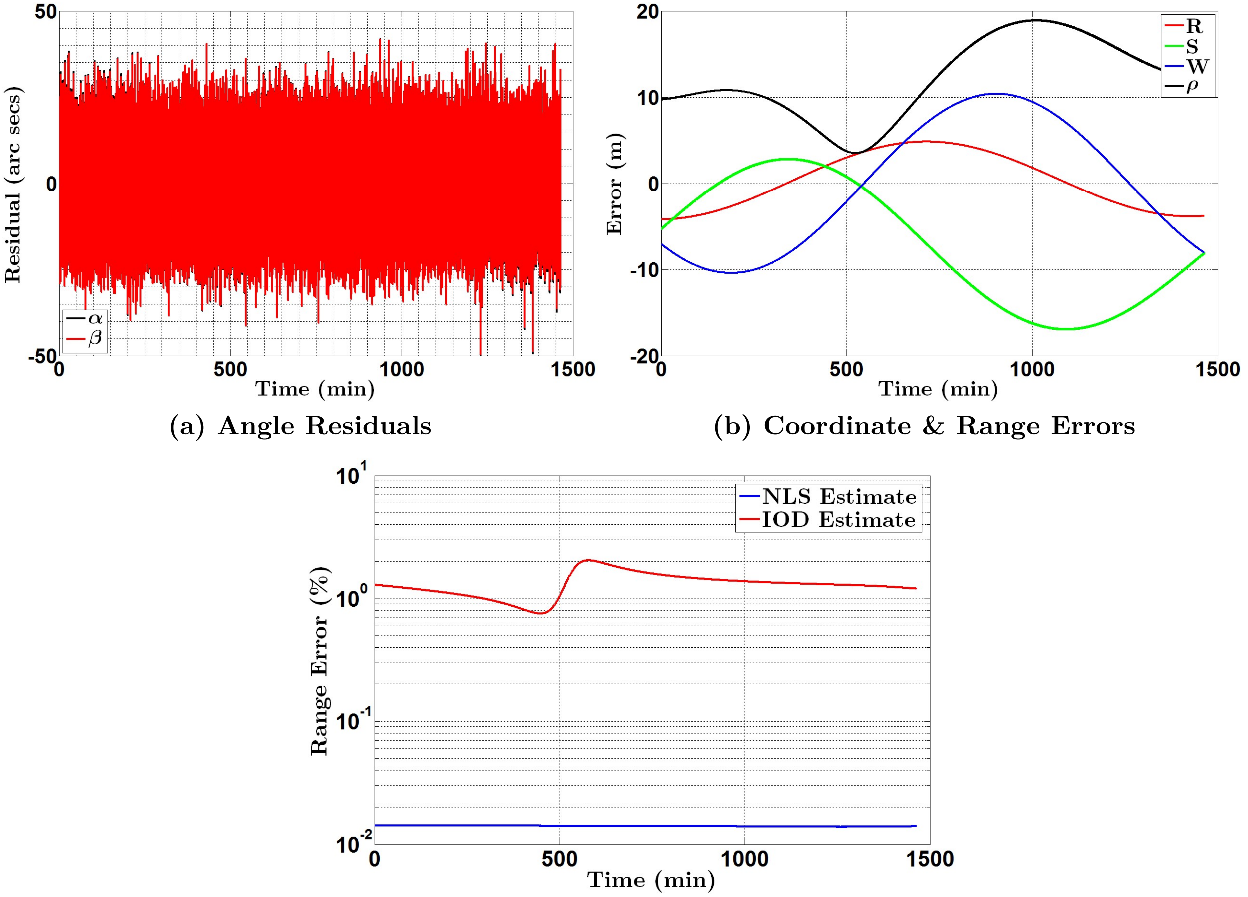
	$\boldsymbol{\delta a}$ **(km)**	$\boldsymbol{\delta e}$ **(nd)**	$\boldsymbol{\delta i}$ **(°)**	$\boldsymbol{\delta \Omega}$ **(°)**	$\boldsymbol{\delta \omega}$ **(°)**	$\boldsymbol{\delta \nu}$ **(°)**
Truth	3.00000	0.00075	0.10000	-0.03000	0.10000	-0.12000
IOD	3.03628	0.00076	0.10111	-0.08787	0.00000	0.03742
NLS†	2.99854	0.00075	0.10001	-0.03001	0.10015	-0.11596
NLS‡	2.99860	0.00075	0.10001	-0.02997	0.09994	-0.11578

† Numerical ***A*** matrix & numerically integrated reference orbit

‡ Two-Body ***A*** matrix & numerically integrated reference orbit

Estimation with the Two-Body *A* matrix & Integrated Reference Orbit

The analytical two-body ***A*** matrix was also tested against the numerical ***A*** matrix to assess whether the added complexity and computational cost is warranted. The results are shown in Figure 63. The RMS of the angle residuals have not changed—10.02 arc sec in both angles. The accuracy of the estimate in Hill's frame is shown in Figure 63(b) where the range error peaks at 19 m. The percent range error holds relatively steady at ~0.014% of the true range, which is a little worse than the error using the numerical ***A*** matrix. Once again, the local motion is well modelled with two-body equations of variation at GEO.



(a) Angle Residuals

(b) Coordinate & Range Errors

(c) Percent Range Error

Figure 63: Case 05C NLS Results (Two-Body A Matrix & Integrated Ref. Orbit). The angle residuals are shown in (a) where α is indicated by the black curve (typically obscured by β) and β is indicated by the red curve. The relative position coordinate and range errors are shown in (b) where the radial direction is indicated in red, the along-track direction is indicated in green, the cross-track direction is indicated in blue and the range is indicated in black. Finally, the percent range error is presented in (c). The red curve designates the IOD solution and the blue curve designates the NLS solution.

6.4 Case 06C – LEO, Small Differential Elements

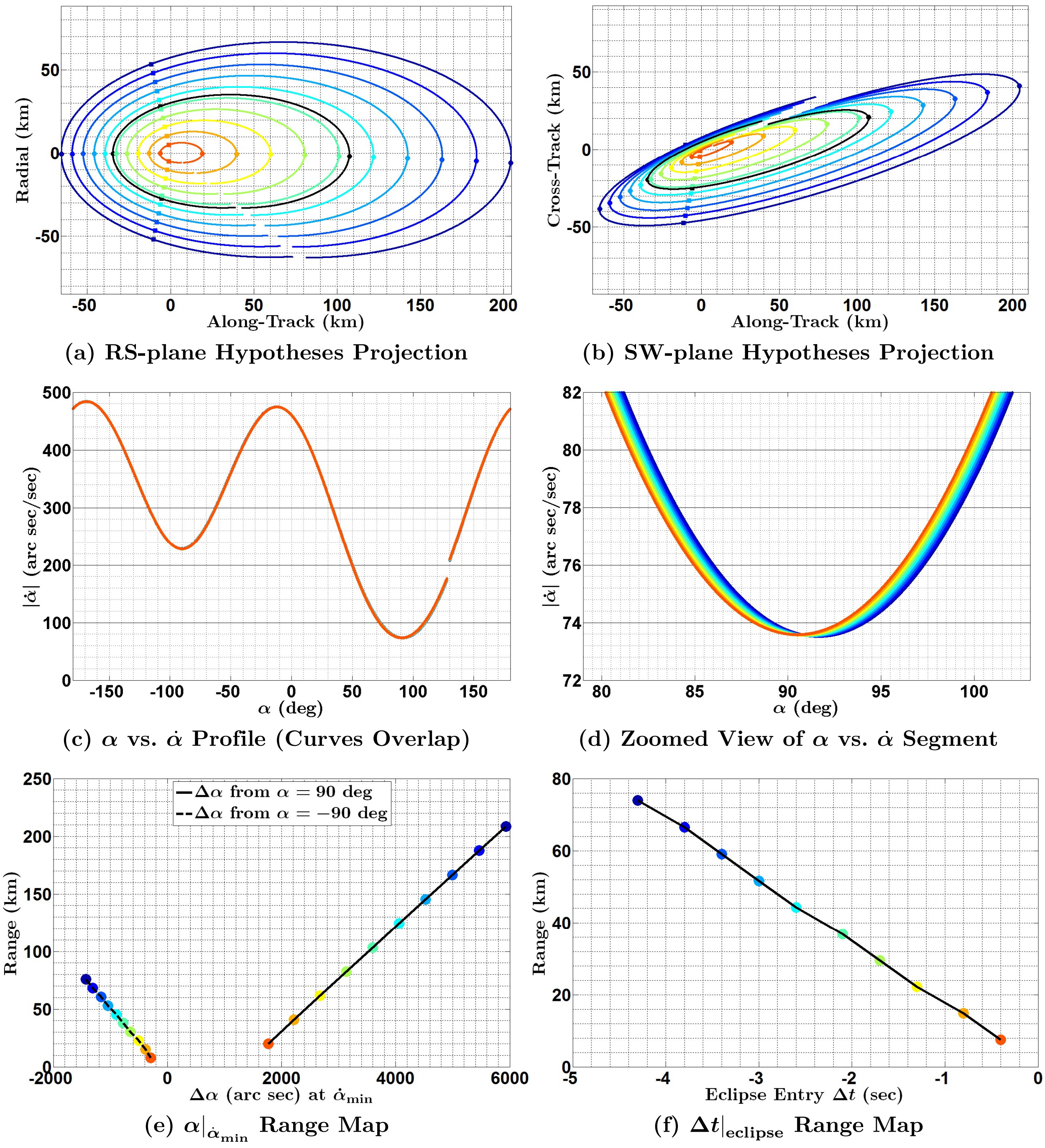
This case demonstrates a drifting NMC trajectory at LEO with small differentials in all orbital elements. A moderate amount of drifting is caused by $\delta a = 1$ km. Out-of-plane motion is induced from $\delta i = 0.2°$ and $\delta\Omega = 0.1°$ and the trajectory is shifted in the along-track direction by the combination of $\delta\omega = -0.2°$, $\delta\nu = 0.5°$ and $\delta\Omega$. The scale of the relative trajectory is dominated by $\delta e = 0.005$ with very minor influence from δa and $\delta\omega$. Table 19 indicates the COEs for the Surveyor, Target and uniformly sampled admissible hypotheses. It also indicates the magnitude of $\boldsymbol{\rho}_1$ for a sense of the relative trajectory's scale.

Table 19: Case 06C Orbital Elements for Truth & Hypotheses

	$\lvert\boldsymbol{\rho}_1\rvert$ (km)	$\boldsymbol{a}$ (km)	$\boldsymbol{e}$ (nd)	$\boldsymbol{i}$ (°)	$\boldsymbol{\Omega}$ (°)	$\boldsymbol{\omega}$ (°)	$\boldsymbol{\nu}$ (°)
Surv	-	6789.086	0.001601	51.75242	72.49305	60.69588	0.000000
Targ	52.87975	6790.086	0.006601	51.95242	72.59305	60.49588	0.500000
H1	100.5349	6791.130	0.011087	52.12857	72.70172	60.69588	0.566485
H2	90.38421	6790.921	0.010128	52.09082	72.68103	60.69588	0.507515
H3	80.23687	6790.712	0.009170	52.05306	72.66061	60.69588	0.448775
H4	70.09503	6790.505	0.008214	52.01530	72.64045	60.69588	0.390295
H5	59.95854	6790.298	0.007258	51.97753	72.62056	60.69588	0.332065
H6	49.82531	6790.092	0.006303	51.93977	72.60092	60.69588	0.274055
H7	39.69639	6789.887	0.005349	51.90200	72.58155	60.69588	0.216275
H8	29.57244	6789.683	0.004396	51.86423	72.56245	60.69588	0.158725
H9	19.45288	6789.479	0.003444	51.82646	72.54362	60.69588	0.101385
H10	9.341857	6789.276	0.002494	51.78869	72.52506	60.69588	0.044245

The IOD results are summarized in a sequence of eight plots shown in Figure 64. The hypotheses seem to have good scalability agreement with the truth as indicated by the trajectory projections in Figure 64(a) and Figure 64(b). The trajectory skews right (skewness factor is $d = 0.4877$) and offers better range observability in the $+\alpha$ domain surrounding the global $\dot{\alpha}_{\text{min}}$. The selection is justified by the larger $\Delta\alpha$ spread in this region as shown in the $\alpha|_{\dot{\alpha}_{\text{min}}}$ range map in Figure 64(e). Range resolvability in the $+\alpha$ domain ($\text{m} = 0.0453$) is comparable to the $-\alpha$ domain ($\text{m} = 0.0599$). A set of 200 $\dot{\alpha}$ waypoints surrounding the global $\dot{\alpha}_{\text{min}}$ are used to map range and scale the IOD COEs.

The eclipse time differential is too short to be of any use for scaling as shown in Figure 64(f), so only the $\alpha|_{\dot{\alpha}}$ range maps are used. The IOD solution is within 9.3% of the true range as shown in Figure 64(h). The accuracy of the IOD solution is also conveyed by the differential orbital elements in Table 20.



(a) RS-plane Hypotheses Projection

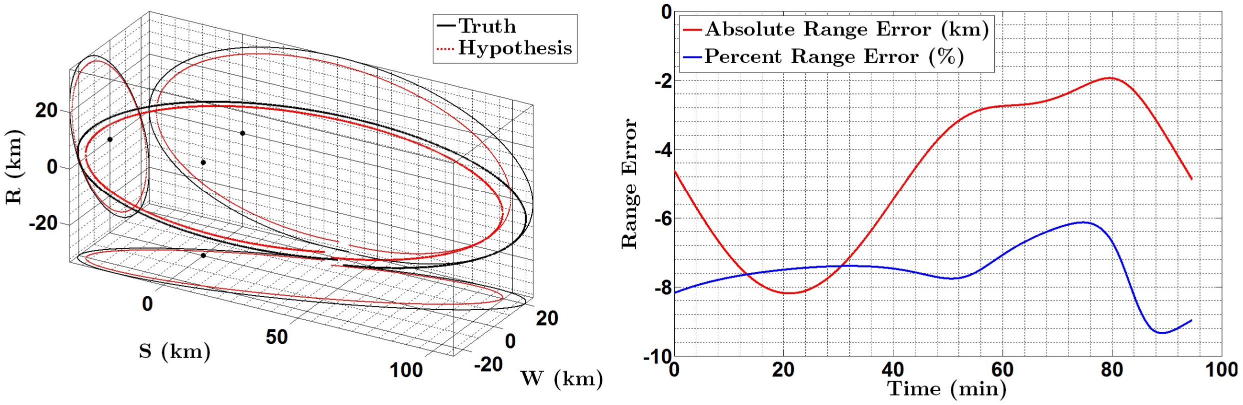
(b) SW-plane Hypotheses Projection

(c) α vs. $\dot{\alpha}$ Profile (Curves Overlap)

(d) Zoomed View of α vs. $\dot{\alpha}$ Segment

(e) $\alpha|_{\dot{\alpha}_{\min}}$ Range Map

(f) $\Delta t|_{\text{eclipse}}$ Range Map



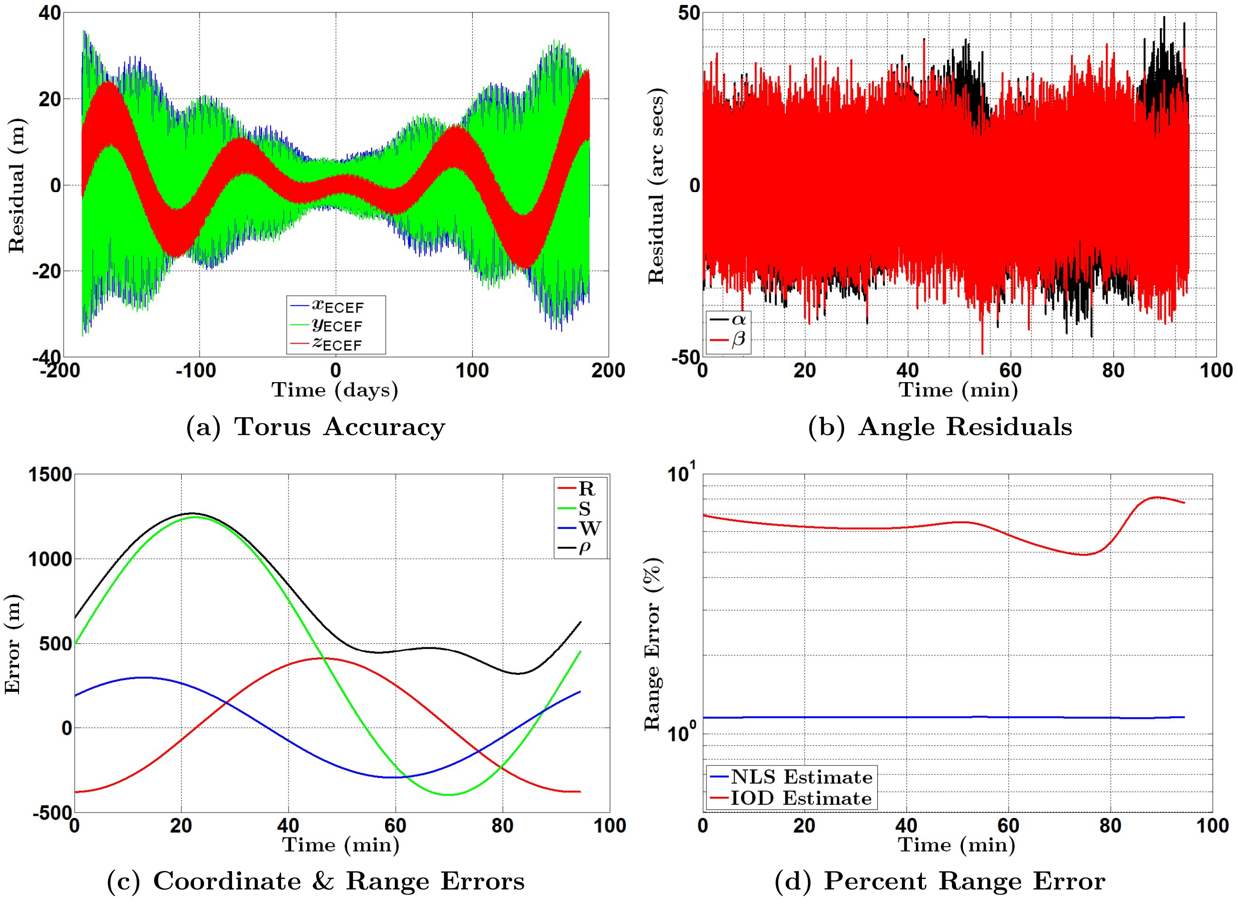
(g) Trajectories & Projections

(h) Range Error

Figure 64: Case 06C Initial Orbit Determination Results. A set of 10 admissible hypothesis trajectories are projected in the rs-plane in (a) and the sw-plane in (b) along with the truth (the black curve). The location of minimum angular rate is indicated by the dots and the location of the maximum angular rate is indicated by the squares. The α vs. $\dot{\alpha}$ curves are plotted in (c), but practically overlap. A zoomed view of a small segment near $\dot{\alpha}_{\min}$ is shown in (d). The $\alpha|_{\dot{\alpha}_{\min}}$ range map is shown in (e) and the eclipse range map is shown in (f). The colored dots correspond to the colors used to represent the hypotheses in (a) and (b). The IOD solution (dashed red curve) is plotted with the truth (solid black curve) in (g) with their projections in all three directions to gain perspective of the in-plane and out-of-plane motion. The black dots indicate the location of the Surveyor for the actual and projected trajectories. The range error is plotted in (h) as an absolute error (red curve) and as a percentage of the truth (blue curve).

Estimation with the Numerical $\boldsymbol{A}$ Matrix & Torus Reference Orbit

The NLS filter results are shown in Figure 65 using the numerical $\boldsymbol{A}$ matrix for local motion dynamics and an IOD reference torus. The RMS of the angle residuals are 10.98 and 10.43 arc sec in α and β, respectively. The RMS of the torus native coordinates are 8.59, 8.59 and 7.76 m in x, y and z, respectively. This is a decent torus fit, but it does affect the accuracy of the estimate as process noise. The accuracy of the estimate in Hill's frame is shown in Figure 65(c) where the range error peaks at 1.265 km. The percent range error holds relatively steady at ~1.15% of the true range, which is a marked improvement to the IOD solution and close to the 1% rule of thumb for RPOs. The accuracy of the NLS solution is also conveyed by the differential orbital elements in Table 20.



(a) Torus Accuracy

(b) Angle Residuals

(c) Coordinate & Range Errors

(d) Percent Range Error

Figure 65: Case 06C NLS Results (Numerical A Matrix & Torus Ref. Orbit). The ECEF coordinate residuals from the torus orbit fitting process are shown in (a) where x, y and z are indicated by blue, green and red, respectively. The angle residuals are shown in (b) where α is indicated by the black curve (typically obscured by β) and β is indicated by the red curve. The relative position coordinate and range errors are shown in (c) where the radial direction is indicated in red, the along-track direction is indicated in green, the cross-track direction is indicated in blue and the range is indicated in black. Finally, the percent range error is presented in (d). The red curve designates the IOD solution and the blue curve designates the NLS solution.

Table 20: Case 06C Differential Orbital Elements for Truth, IOD & NLS

	δa (km)	δe (nd)	δi (°)	$\delta\Omega$ (°)	$\delta\omega$ (°)	$\delta\nu$ (°)
Truth	1.00000	0.00500	0.20000	0.10000	-0.20000	0.50000
IOD	0.99980	0.00467	0.18610	0.10723	0.00000	0.27213
NLS*	1.02403	0.00506	0.20230	0.10116	-0.20966	0.51967
NLS†	1.00775	0.00503	0.20121	0.10060	-0.20285	0.51124
NLS⋆	1.04106	0.00508	0.20306	0.10167	-0.22731	0.53851
NLS‡	1.02367	0.00505	0.20197	0.10110	-0.21972	0.52931

* Numerical $\boldsymbol{A}$ matrix & torus reference orbit

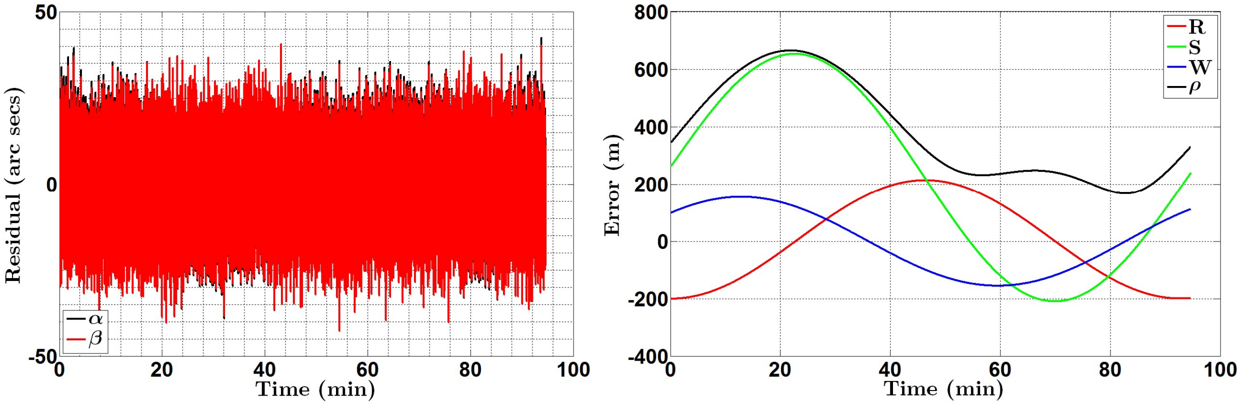
† Numerical $\boldsymbol{A}$ matrix & numerically integrated reference orbit

⋆ Two-Body $\boldsymbol{A}$ matrix & torus reference orbit

‡ Two-Body $\boldsymbol{A}$ matrix & numerically integrated reference orbit

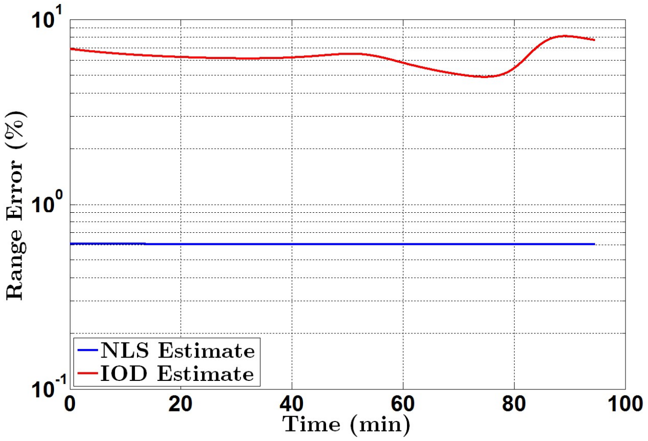
Estimation with the Numerical $\boldsymbol{A}$ Matrix & Integrated Reference Orbit

The NLS filter results are shown in Figure 66 using the numerical $\boldsymbol{A}$ matrix for local motion dynamics and a numerically integrated reference orbit. The RMS of the angle residuals are 10.04 and 10.06 arc sec in α and β, respectively. The accuracy of the estimate in Hill's frame is shown in Figure 66(b) where the range error peaks at 665 m. The percent range error holds relatively steady at ~0.61% of the true range, which is better than the last trial with the torus.



(a) Angle Residuals

(b) Coordinate & Range Errors

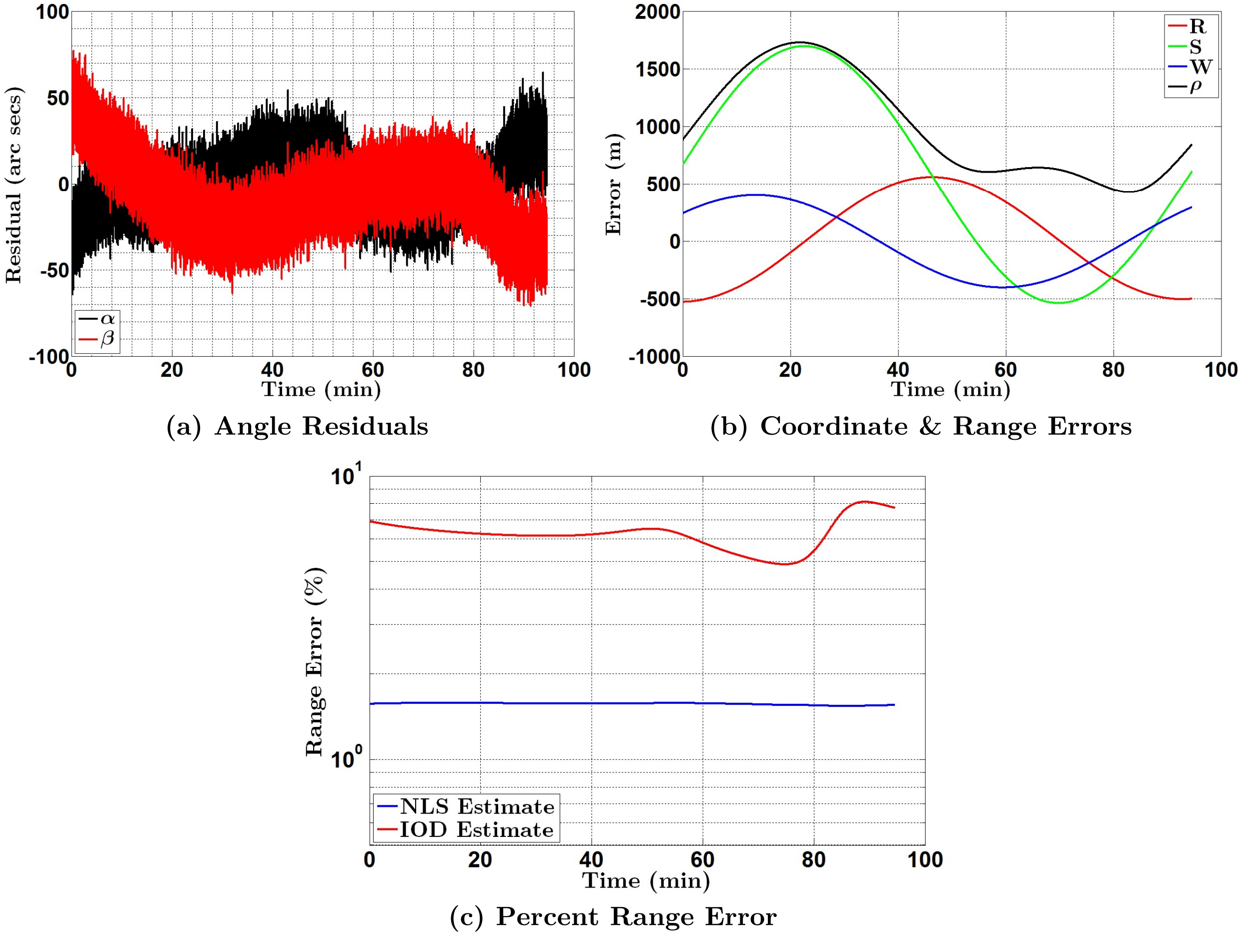


(c) Percent Range Error

Figure 66: Case 06C NLS Results (Numerical $\boldsymbol{A}$ Matrix & Integrated Ref. Orbit). The angle residuals are shown in (a) where α is indicated by the black curve (obscured by β) and β is indicated by the red curve. The relative position coordinate and range errors are shown in (b) where the radial direction is indicated in red, the along-track direction is indicated in green, the cross-track direction is indicated in blue and the range is indicated in black. Finally, the percent range error is presented in (c). The red curve designates the IOD solution and the blue curve designates the NLS solution.

Estimation with the Two-Body $\boldsymbol{A}$ Matrix & Torus Reference Orbit

Results for the analytical two-body $\boldsymbol{A}$ matrix with the IOD reference torus are shown in Figure 67 where we have left out the torus construction residual since it has not changed. The RMS of the angle residuals are 15.80 and 21.66 arc sec in α and β, respectively. The accuracy of the estimate in Hill's frame is shown in Figure 67(b) where the range error peaks at 1.727 km. The percent range error holds relatively steady at ~1.56% of the true range which is about half of the error using the numerical $\boldsymbol{A}$ matrix. Not surprisingly, this is the worst result of the four dynamics model instantiations since the torus and Keplerian $\boldsymbol{A}$ matrix represent variations on the true dynamics.



(a) Angle Residuals

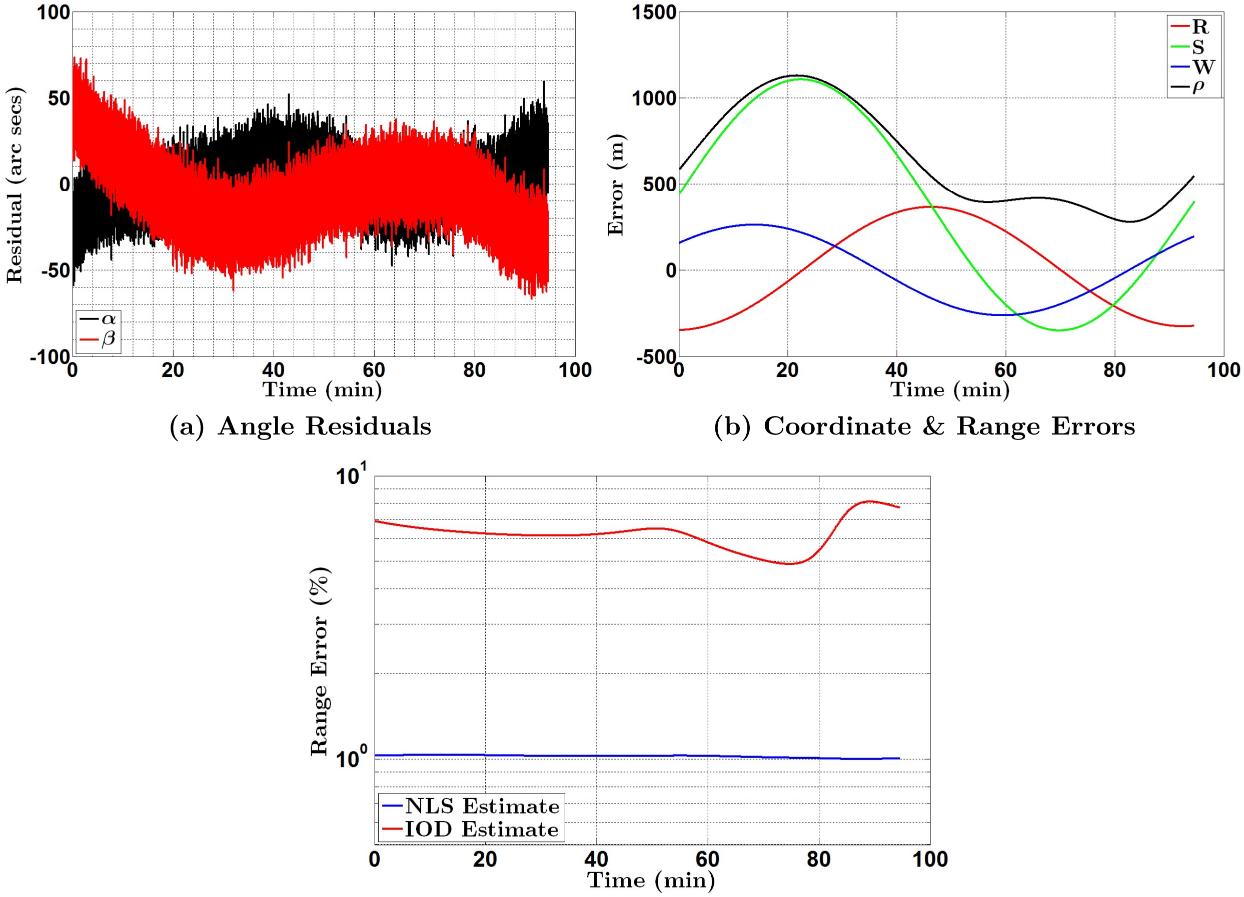
(b) Coordinate & Range Errors

(c) Percent Range Error

Figure 67: Case 06C NLS Results (Two-Body A Matrix & Torus Ref. Orbit). The angle residuals are shown in (a) where α is indicated by the black curve and β is indicated by the red curve. The relative position coordinate and range errors are shown in (b) where the radial direction is indicated in red, the along-track direction is indicated in green, the cross-track direction is indicated in blue and the range is indicated in black. Finally, the percent range error is presented in (c). The red curve designates the IOD solution and the blue curve designates the NLS solution.

Estimation with the Two-Body A Matrix & Integrated Reference Orbit

One final test is performed using the analytical two-body $\boldsymbol{A}$ matrix and a numerically integrated reference orbit. The results are shown in Figure 68. The RMS of the angle residuals are 13.08 and 20.07 arc sec in α and β, respectively. The accuracy of the estimate in Hill's frame is shown in Figure 68(b) where the range error peaks at 1.128 km. The percent range error holds relatively steady at ~1.03% of the true range.



(a) Angle Residuals

(b) Coordinate & Range Errors

(c) Percent Range Error

Figure 68: Case 06C NLS Results (Two-Body A Matrix & Integrated Ref. Orbit). The angle residuals are shown in (a) where α is indicated by the black curve and β is indicated by the red curve. The relative position coordinate and range errors are shown in (b) where the radial direction is indicated in red, the along-track direction is indicated in green, the cross-track direction is indicated in blue and the range is indicated in black. Finally, the percent range error is presented in (c). The red curve designates the IOD solution and the blue curve designates the NLS solution.

6.5 Case 07C – GEO, Large Differential Argument of Perigee

This case demonstrates a drifting NMC trajectory at GEO with a large differential argument of perigee and small differentials in the other orbital elements. We have intentionally selected $\delta\omega$ to ensure that the IOD approximation of argument of perigee will be in error by at least 5°. Under ideal circumstances, this is the largest error encountered with a coarse 10° increment heuristic search described in Section 5.3.2. Without keeping the reader in suspense, the argument of perigee search performs as well as it can by settling on 152.8015° which is 5° in error after two iterations, just as planned. The question we seek to answer is if the IOD process can handle several degrees error in argument of perigee. As a control to this experiment, we seed a parallel run (**Case 07D**) with the exact argument of perigee. The results are presented together for a sharper contrast.

In this case, the Target experiences a moderate amount of drifting caused by $\delta a = 3$ km. Out-of-plane motion is induced from $\delta i = 0.1°$ and $\delta\Omega = -0.03°$ and the trajectory is shifted in the negative along-track direction by the combination of $\delta\omega = 295°$, $\delta\nu = 65°$ and $\delta\Omega$. The scale of the relative trajectory is dominated by $\delta e = 0.00075$ and $\delta\omega$ with minor influence from δa. Table 21 indicates the COEs for the Surveyor, Target and uniformly sampled admissible hypotheses for the run with 5° error in argument of perigee and Table 22 indicates the COEs for the control run with 0° error in argument of perigee. The tables also indicate the magnitude of $\boldsymbol{\rho}_1$ for a sense of the relative trajectory's scale.

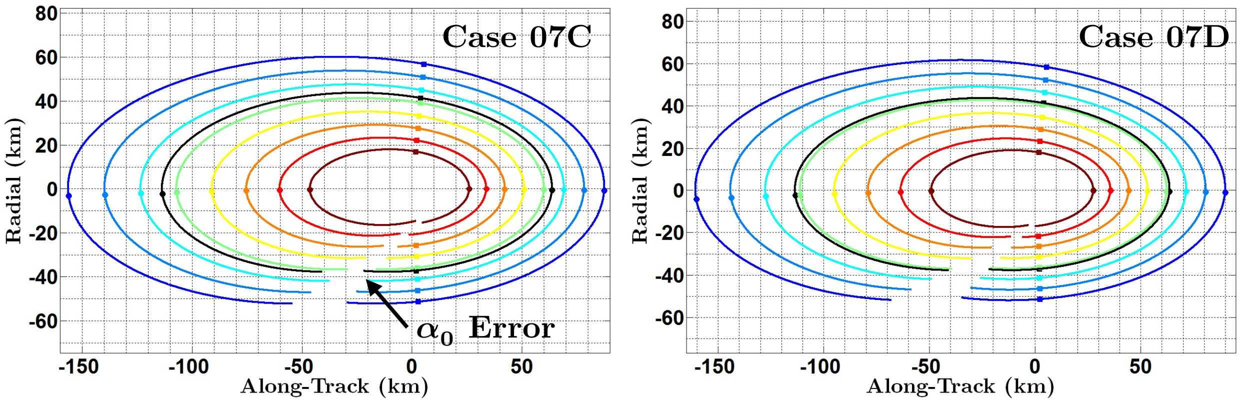
Table 21: Case 07C Orbital Elements for Truth & Hypotheses

	$\lvert\boldsymbol{\rho}_1\rvert$ (km)	$\boldsymbol{a}$ (km)	$\boldsymbol{e}$ (nd)	$\boldsymbol{i}$ (°)	$\boldsymbol{\Omega}$ (°)	$\boldsymbol{\omega}$ (°)	$\boldsymbol{\nu}$ (°)
Surv	-	42166.15	0.000304	0.192900	87.10659	222.8015	270.000
Targ	68.2192	42169.15	0.001054	0.292900	87.07659	157.8015	335.000
H1	94.8646	42170.17	0.001405	0.331191	87.17212	152.8015	-20.1058
H2	86.6081	42169.59	0.001266	0.317925	87.39592	152.8015	-20.3237
H3	78.4926	42169.03	0.001127	0.304764	87.65627	152.8015	-20.5781
H4	70.5701	42168.49	0.000988	0.291757	87.96020	152.8015	-20.8758
H5	62.9246	42167.97	0.000848	0.278963	88.32784	152.8015	-21.2371
H6	55.7029	42167.50	0.000707	0.266498	88.78146	152.8015	-21.6839
H7	49.1793	42167.11	0.000565	0.254566	89.36442	152.8015	-22.2596
H8	43.9287	42166.85	0.000419	0.243613	90.16044	152.8015	-23.0476
H9	41.4851	42166.90	0.000266	0.234992	91.31342	152.8015	-24.1913
H10	47.8848	42167.92	0.000089	0.236771	92.63464	152.8015	-25.5014

Table 22: Case 07D Orbital Elements for Truth & Hypotheses

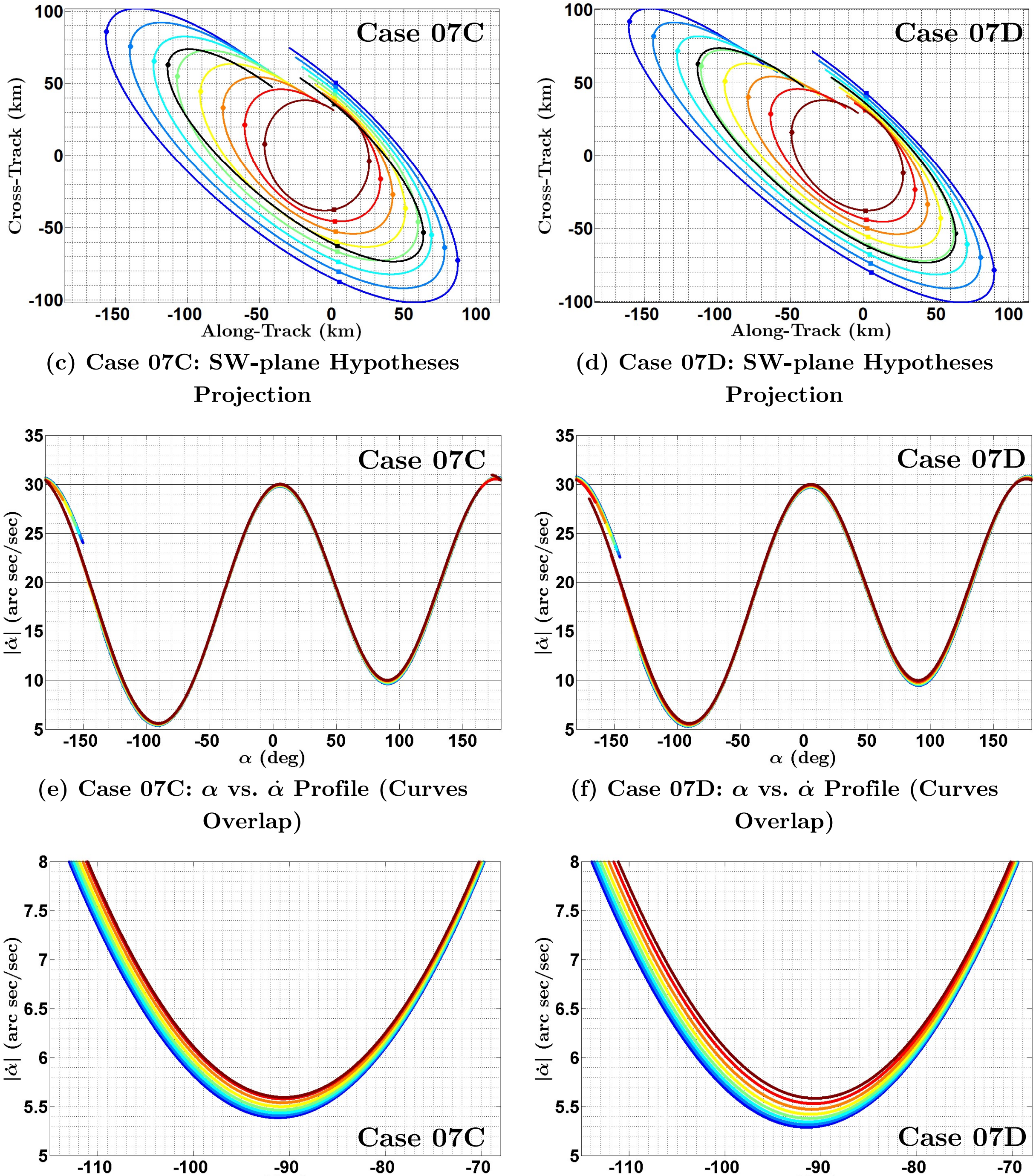
	$\lvert\boldsymbol{\rho}_1\rvert$ (km)	$\boldsymbol{a}$ (km)	$\boldsymbol{e}$ (nd)	$\boldsymbol{i}$ (°)	$\boldsymbol{\Omega}$ (°)	$\boldsymbol{\omega}$ (°)	$\boldsymbol{\nu}$ (°)
Surv	-	42166.15	0.000304	0.192900	87.10659	222.8015	270.000
Targ	68.2192	42169.15	0.001054	0.292900	87.07659	157.8015	335.000
H1	90.4899	42171.05	0.001443	0.329743	86.34160	157.8015	-24.2826
H2	82.5461	42170.37	0.001307	0.316741	86.55616	157.8015	-24.4911
H3	74.7329	42169.70	0.001171	0.303846	86.80529	157.8015	-24.7341
H4	67.1031	42169.05	0.001034	0.291101	87.09915	157.8015	-25.0216
H5	59.7337	42168.43	0.000897	0.278569	87.45260	157.8015	-25.3685
H6	52.7620	42167.84	0.000759	0.266353	87.89235	157.8015	-25.8013
H7	46.4161	42167.32	0.000619	0.254636	88.45601	157.8015	-26.3576
H8	41.1687	42166.92	0.000477	0.243776	89.21905	157.8015	-27.1125
H9							
H10							

The IOD results are summarized in Figure 69 with Case 07C plots on the left and Case 07D on the right. The Case 07C hypotheses do *not* show good scalability agreement with the truth as indicated by the misalignment of α_0 in Figure 69(a) and discordant OOP motion in Figure 69(c). Conversely, Case 07D hypotheses do show good scalability agreement with the truth. The $\alpha|_{\dot{\alpha}_{\min}}$ range maps in Figure 69(i) and (j) show a larger $\Delta\alpha$ spread in the $-\alpha$ domain surrounding the global $\dot{\alpha}_{\min}$ so there is better range observability in this region. In this case, the direction of enhanced range observability agrees with the leftward skewness (skewness factor is $d = 1.2811$). Range resolvability in the $-\alpha$ domain (m = 0.0560 for Case 07C and m = 0.0588 for Case 07D) is comparable to the $+\alpha$ domain (m = 0.0785 for Case 07C and m = 0.0415 for Case 07D) so there is no reason to consider changing sides. Both runs have good structural coherency in the α vs $\dot{\alpha}$ arcs surrounding the global $\dot{\alpha}_{\min}$, so 200 $\dot{\alpha}$ waypoints surrounding the global $\dot{\alpha}_{\min}$ are used to map range and scale the IOD COEs. Despite their coherency, the 5° bias in argument of perigee causes a shift in the curves for Case 07C compared to Case 07D as shown in Figure 69(g) and (h). There is no eclipse at GEO during the time of this scenario, so photometry is not used for scaling the COEs. At its worst, the Case 07C IOD solution has a range error of 34% and the Case 07D solutions has a range error of 1.7% as shown in Figure 69(o) and (p). The accuracy of the IOD solution is also conveyed by the differential orbital elements in Table 23.



(a) Case 07C: RS-plane Hypotheses Projection

(b) Case 07D: RS-plane Hypotheses Projection



(c) Case 07C: SW-plane Hypotheses Projection

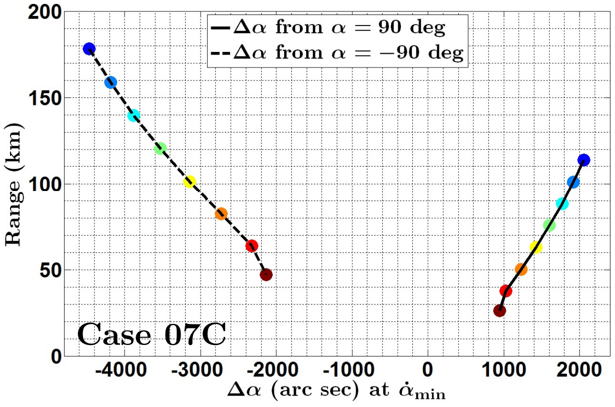
(d) Case 07D: SW-plane Hypotheses Projection

(e) Case 07C: α vs. $\dot{\alpha}$ Profile (Curves Overlap)

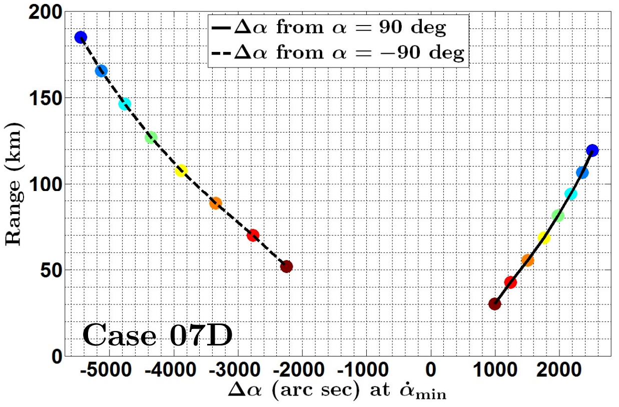
(f) Case 07D: α vs. $\dot{\alpha}$ Profile (Curves Overlap)

(g) Case 07C: Zoomed View of α vs. $\dot{\alpha}$ Segment

(h) Case 07D: Zoomed View of α vs. $\dot{\alpha}$ Segment



(i) Case 07C: $\alpha|_{\dot{\alpha}_{\min}}$ Range Map



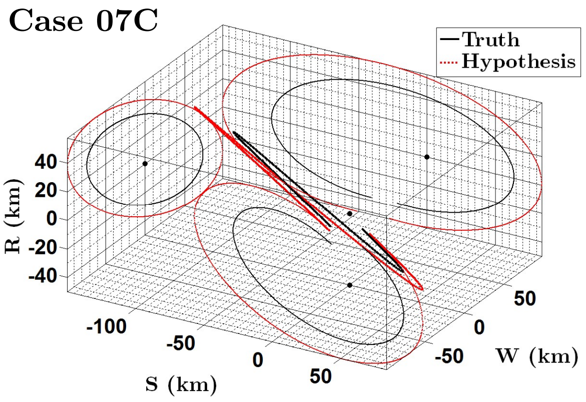
(j) Case 07D: $\alpha|_{\dot{\alpha}_{\min}}$ Range Map

Intentionally Blank
No Eclipse

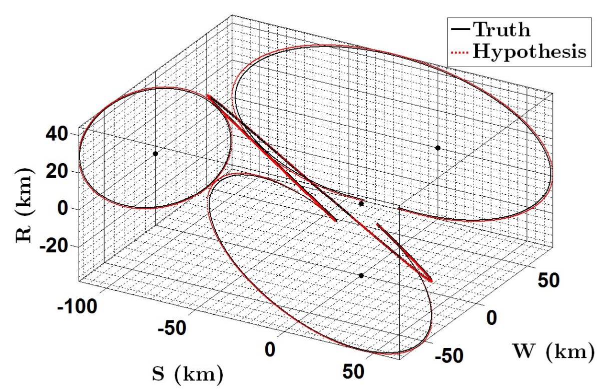
Intentionally Blank
No Eclipse

(k) Case 07C: $\Delta t|_{\text{eclipse}}$ Range Map

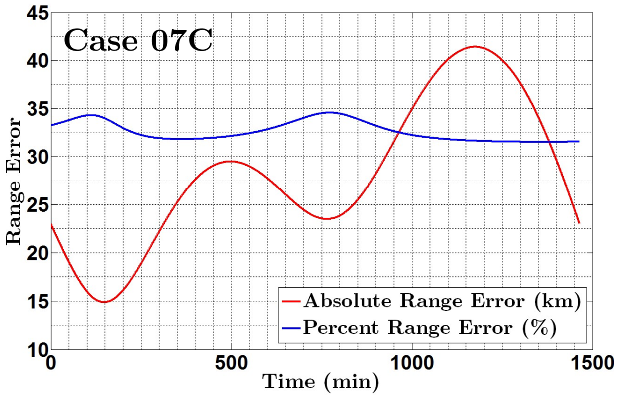
(l) Case 07D: $\Delta t|_{\text{eclipse}}$ Range Map



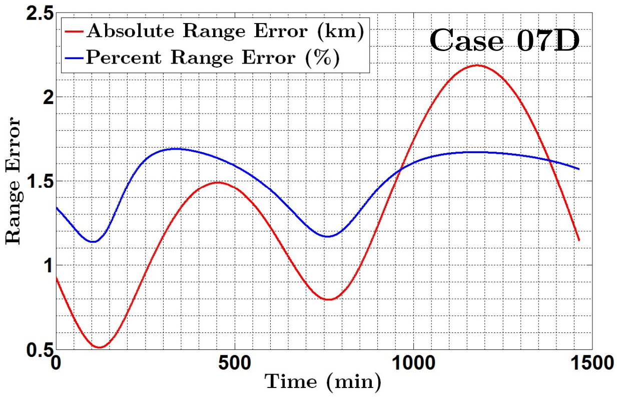
(m) Case 07C: Trajectories & Projections



(n) Case 07D: Trajectories & Projections



(o) Case 07C: Range Error



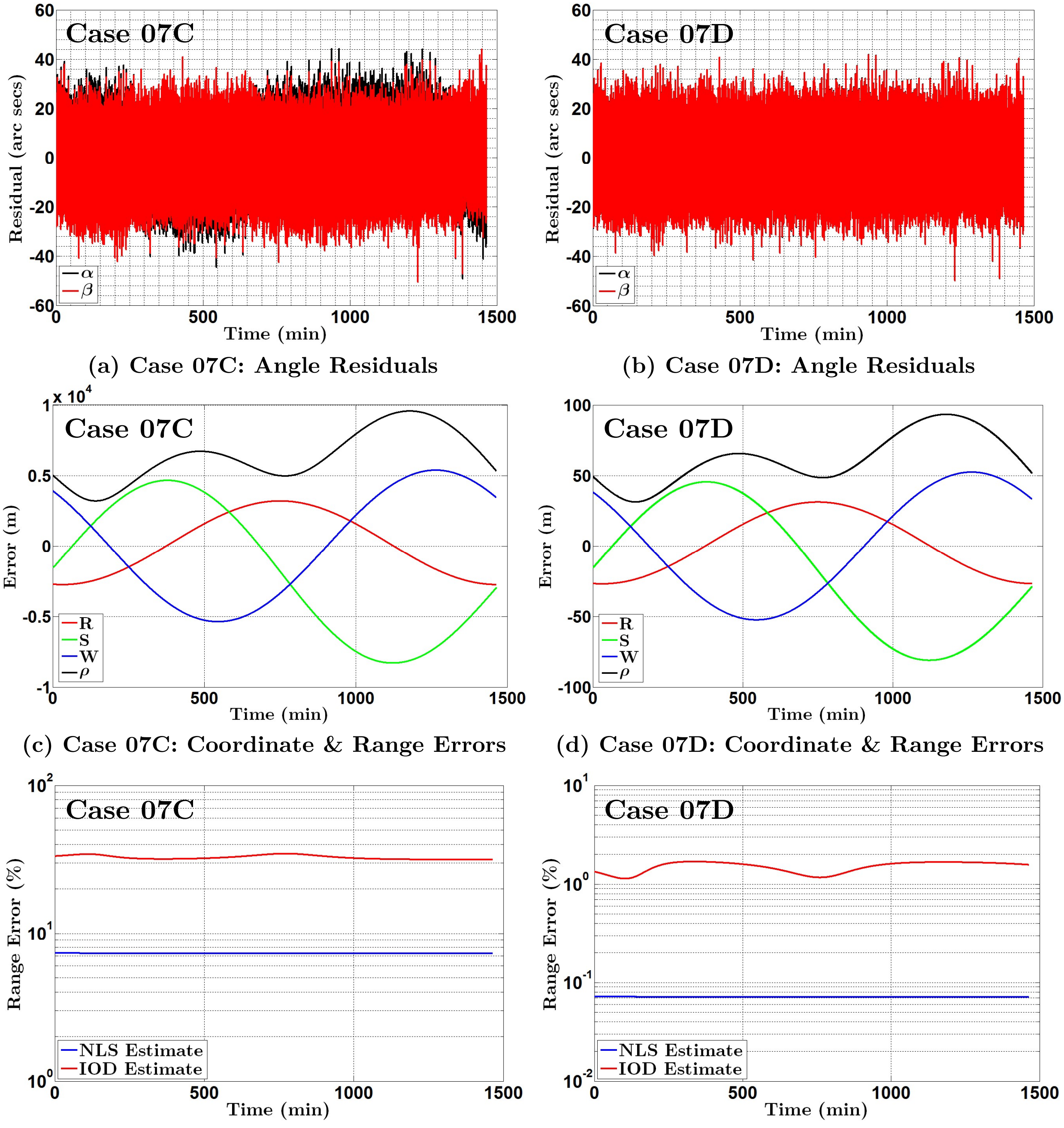
(p) Case 07D: Range Error

Figure 69: Case 07 Initial Orbit Determination Results. A set of eight admissible hypothesis trajectories are projected in the rs-plane in (a) for Case 07C and in (b) for Case 07D along with the truth (the black curve). The same is done for the sw-plane in (c) for Case 07C and (d) for Case 07D. The location of minimum angular rate is indicated by the dots and the location of the maximum angular rate is indicated by the squares. The α vs. $\dot{\alpha}$ curves are plotted in (e) and (f) for Case 07C and Case 07D, respectively. A zoomed view of a small segment exhibiting structural coherency is shown in (g) and (h). The $\alpha|_{\dot{\alpha}_{\min}}$ range map is shown in (i) and (j). The colored dots correspond to the colors used to represent the hypotheses in (a) and (b). The IOD solution (dashed red curve) is plotted with the truth (solid black curve) in (m) and (n) with their projections in all three directions to gain perspective of the in-plane and out-of-plane motion. The black dots indicate the location of the Surveyor for the actual and projected trajectories. The range error is plotted in (o) and (p) as an absolute error (red curve) and as a percentage of the truth (blue curve).

These results are solid evidence that the IOD technique is dependent upon good argument of perigee approximation for high differential arguments of perigee. The previous cases performed reasonably well with low differentials and errors in the argument of perigee that were a fraction of a degree. Our technique can handle higher fidelity searches to improve the argument of perigee. Rather than searching in 10° increments, the code should be modified to search in 1° increments next, but it comes with 10 × the computational cost.

Estimation with the Two-Body $\boldsymbol{A}$ Matrix & Integrated Reference Orbit

Given the large error in the Case 07C IOD solution, there is little to be gained from highly accurate local motion dynamics, so the analytical two-body $\boldsymbol{A}$ matrix is used with the batch filter. The results are shown in Figure 70. Once again, we show the Case 07D results in parallel. The RMS of the angle residuals for Case 07C are 10.33 and 10.21 arc sec in α and β, respectively, and 10.02 and 10.02 arc sec for Case 07D. The accuracy of the estimate in Hill's frame is shown in Figure 70(c) and (d) where the range error peaks at 9.56 km for Case 07C and 93 m for Case 07D. The percent range error holds relatively steady at ~7.3% of the true range for Case 07C and ~0.071% for Case 07D.



(a) Case 07C: Angle Residuals

(b) Case 07D: Angle Residuals

(c) Case 07C: Coordinate & Range Errors

(d) Case 07D: Coordinate & Range Errors

(e) Case 07C: Percent Range Error

(f) Case 07D: Percent Range Error

Figure 70: Case 07 NLS Results (Two-Body A Matrix & Integrated Ref. Orbit). The angle residuals are shown in (a) and (b) for Case 07C and Case

07D, respectively, where α is indicated by the black curve and β is indicated by the red curve. The relative position coordinate and range errors are shown in (c) and (d) where the radial direction is indicated in red, the along-track direction is indicated in green, the cross-track direction is indicated in blue and the range is indicated in black. Finally, the percent range error is presented in (e) and (f). The red curve designates the IOD solution and the blue curve designates the NLS solution.

Table 23: Case 07 Differential Orbital Elements for Truth, IOD & NLS

	δa (km)	δe (nd)	δi (°)	$\delta\Omega$ (°)	$\delta\omega$ (°)	$\delta\nu$ (°)
Truth	3.00000	0.00075	0.10000	-0.03000	-65.0000	65.0000
07C IOD	3.85035	0.00104	0.13336	0.24035	-70.0000	69.7220
07D IOD	3.05911	0.00076	0.10140	-0.10772	-65.0000	65.0769
07C NLS‡	3.22519	0.00081	0.10730	-0.03037	-66.0260	66.0284
07D NLS‡	3.00044	0.00075	0.10007	-0.02999	-65.0097	65.0139

‡ Two-Body $\boldsymbol{A}$ matrix & numerically integrated reference orbit

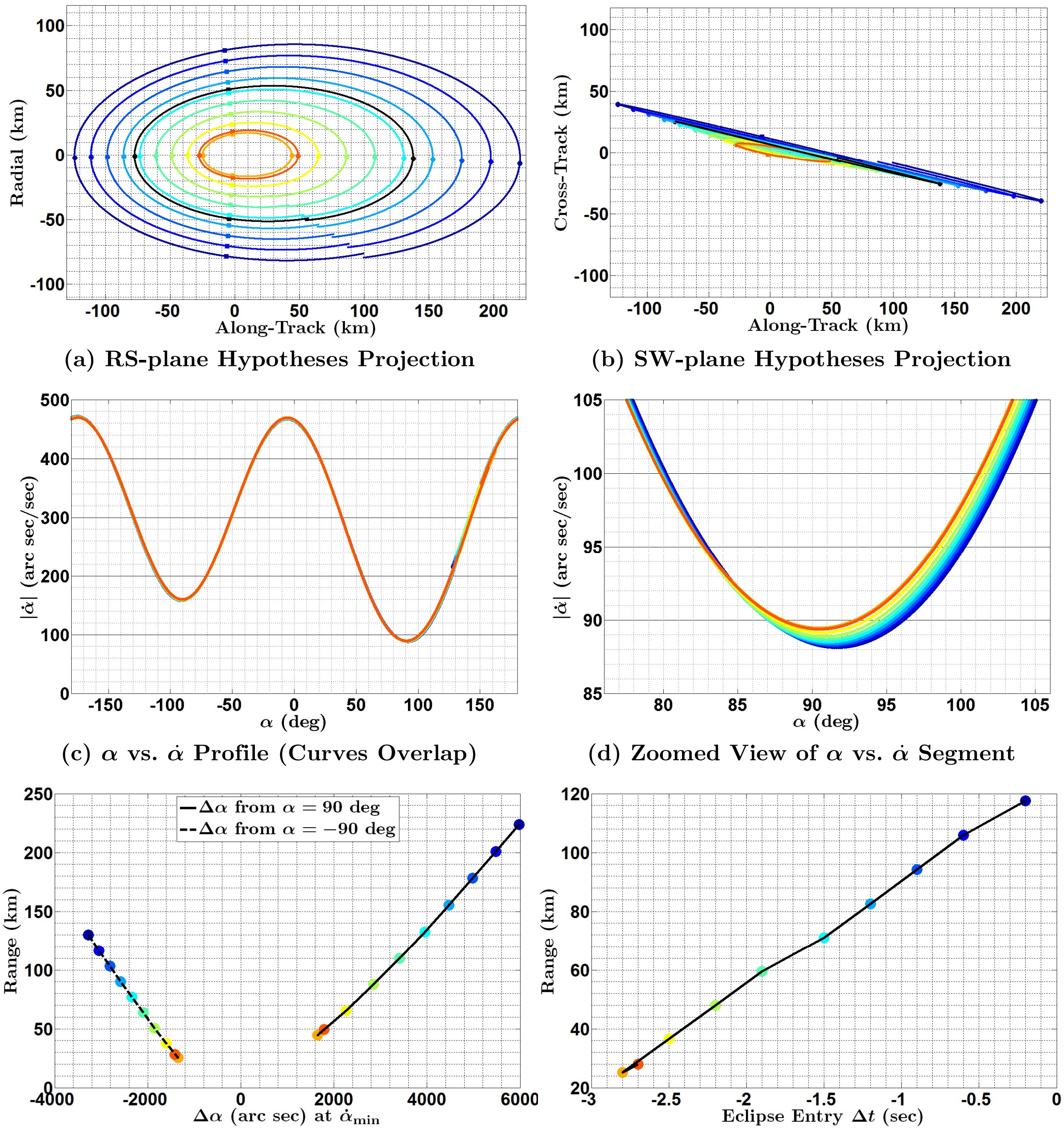
6.6 Case 08C – LEO, Large Differential Argument of Perigee

This case demonstrates a drifting NMC trajectory at LEO with a large differential argument of perigee and small differentials in the other orbital elements. A moderate amount of drifting is caused by $\delta a = 1$ km. Out-of-plane motion is induced from $\delta i = 0.2°$ and $\delta\Omega = 0.1°$ and the trajectory is shifted in the along-track direction by the combination of $\delta\omega = 130.4°$, $\delta\nu = -130.0°$ and $\delta\Omega$. The scale of the relative trajectory is dominated by $\delta e = 0.005$ and $\delta\omega$ with minor influence from δa. Table 24 indicates the COEs for the Surveyor, Target and uniformly sampled admissible hypotheses. It also indicates the magnitude of $\boldsymbol{\rho}_1$ for a sense of the relative trajectory's scale. Note that $\boldsymbol{\rho}_1$ hypotheses were initiated with magnitudes of $100, 90, \cdots, 10$, so the values listed in Table 24 indicate a scale variation from what was initially stipulated. The Thiel-Sen estimator eliminates H1, H5, H6, H9 and H10.

Table 24: Case 08C Orbital Elements for Truth & Hypotheses

	$\|\rho_1\|$ (km)	a (km)	e (nd)	i (°)	Ω (°)	ω (°)	ν (°)
Surv	-	6789.086	0.001601	51.75242	72.49305	60.69588	150.0000
Targ	61.19090	6790.086	0.006601	51.95242	72.59305	191.0959	20.00000
H1	92.73028	6790.976	0.011227	52.06603	72.63360	190.6959	20.75806
H2	84.00525	6790.747	0.009968	52.03357	72.61712	190.6959	20.66533
H3	75.32311	6790.520	0.008710	52.00113	72.60069	190.6959	20.57314
H4	66.69677	6790.296	0.007452	51.96869	72.58432	190.6959	20.48146
H5	58.15519	6790.075	0.006194	51.93629	72.56800	190.6959	20.39041
H6	49.74044	6789.859	0.004935	51.90393	72.55176	190.6959	20.30005
H7	41.53517	6789.651	0.003676	51.87165	72.53562	190.6959	20.21059
H8	33.71309	6789.457	0.002415	51.83957	72.51964	190.6959	20.12254
H9	26.69981	6789.297	0.001150	51.80801	72.50401	190.6959	20.03742
H10	28.30968	6789.331	0.001462	51.81570	72.50781	190.6959	20.05795

The IOD results are summarized in a sequence of eight plots shown in Figure 71. The hypotheses have reasonably good scalability agreement with the truth as indicated by the trajectory projections in Figure 71(a) and Figure 71(b), but the cross-track motion for the shorter ranges seem to diverge as was noted above. The trajectory skews right (skewness factor is $d = 0.7173$) and offers better range observability in the $+\alpha$ domain surrounding the global $\dot{\alpha}_{\min}$. The selection is justified by the larger $\Delta\alpha$ spread in this region as shown in the $\alpha|_{\dot{\alpha}_{\min}}$ range map in Figure 71(e). Range resolvability in the $+\alpha$ domain ($\mathrm{m} = 0.0414$) is comparable to the $-\alpha$ domain ($\mathrm{m} = 0.0542$). A set of 200 $\dot{\alpha}$ waypoints surrounding the global $\dot{\alpha}_{\min}$ are used to map range and scale the IOD COEs. The eclipse time differential is too short to be of any use for scaling as shown in Figure 71(f), so only the $\alpha|_{\dot{\alpha}}$ range maps are used. The IOD solution is within 1.6% of the true range as shown in Figure 71(h). The accuracy of the IOD solution is also conveyed by the differential orbital elements in Table 25.



(a) RS-plane Hypotheses Projection

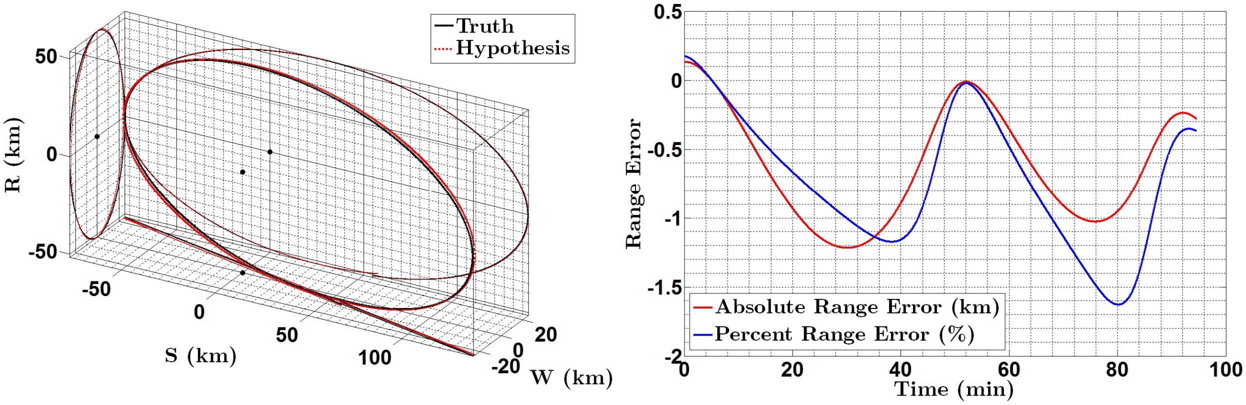
(b) SW-plane Hypotheses Projection

(c) α vs. $\dot{\alpha}$ Profile (Curves Overlap)

(d) Zoomed View of α vs. $\dot{\alpha}$ Segment

(e) $\alpha|_{\dot{\alpha}_{\min}}$ Range Map

(f) $\Delta t|_{\text{eclipse}}$ Range Map



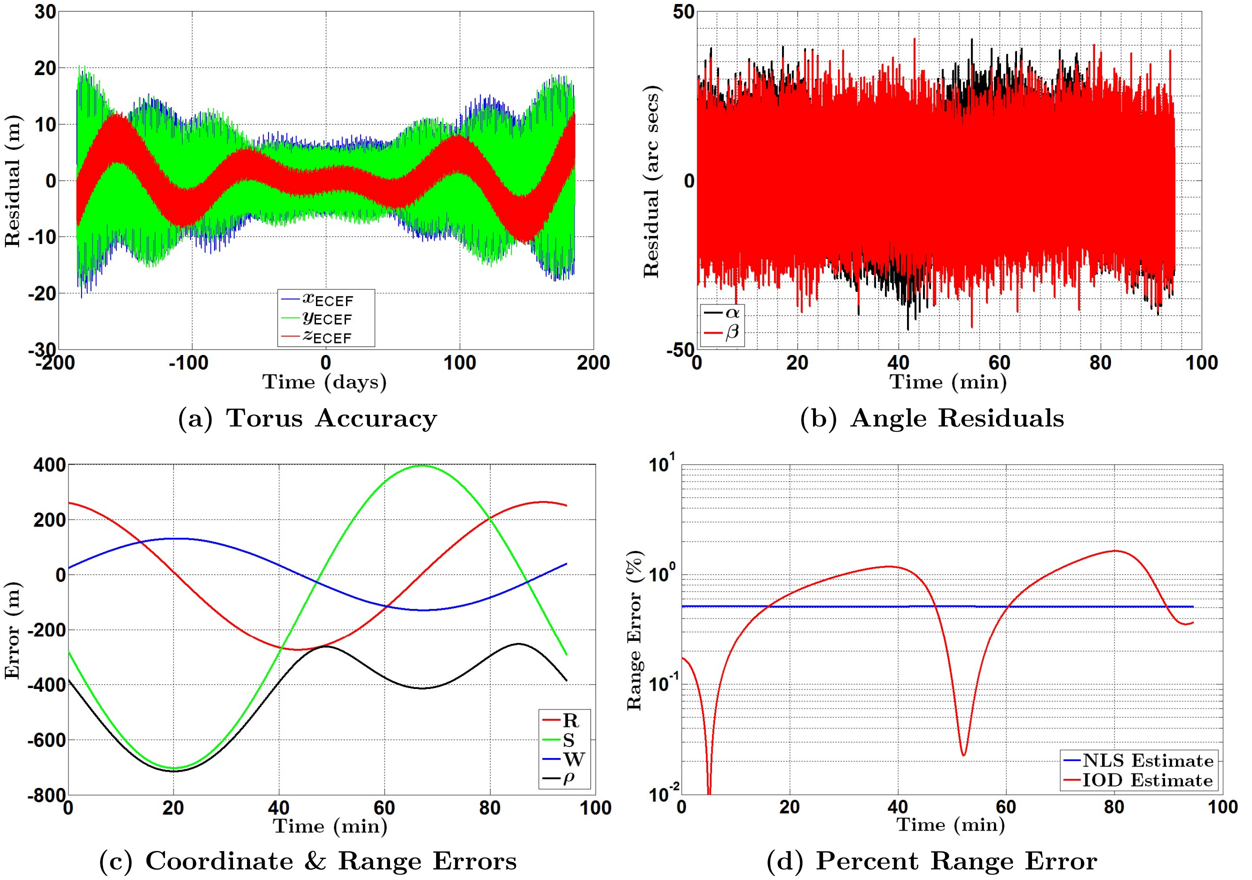
(g) **Trajectories & Projections**

(h) **Range Error**

Figure 71: Case 08C Initial Orbit Determination Results. A set of 10 admissible hypothesis trajectories are projected in the rs-plane in (a) and the sw-plane in (b) along with the truth (the black curve). The location of minimum angular rate is indicated by the dots and the location of the maximum angular rate is indicated by the squares. The α vs. $\dot{\alpha}$ curves are plotted in (c), but practically overlap. A zoomed view of a small segment near $\dot{\alpha}_{\min}$ is shown in (d). The $\alpha|_{\dot{\alpha}_{\min}}$ range map is shown in (e) and the eclipse range map is shown in (f). The colored dots correspond to the colors used to represent the hypotheses in (a) and (b). The IOD solution (dashed red curve) is plotted with the truth (solid black curve) in (g) with their projections in all three directions to gain perspective of the in-plane and out-of-plane motion. The black dots indicate the location of the Surveyor for the actual and projected trajectories. The range error is plotted in (h) as an absolute error (red curve) and as a percentage of the truth (blue curve).

Estimation with the Numerical $\boldsymbol{A}$ Matrix & Torus Reference Orbit

The NLS filter results are shown in Figure 72 using the numerical $\boldsymbol{A}$ matrix for local motion dynamics and an IOD reference torus. The RMS of the angle residuals are 10.60 and 10.16 arc sec in α and β, respectively. The RMS of the torus native coordinates are 4.77, 4.78 and 3.53 m in x, y and z, respectively. This is a decent torus fit, but it does affect the accuracy of the estimate as process noise. The accuracy of the estimate in Hill's frame is shown in Figure 72(c) where the range error peaks at -715 m. The percent range error holds relatively steady at ~0.51% of the true range, which is a meager improvement to the IOD solution, but still better than the 1% rule of thumb for RPOs. The accuracy of the NLS solution is also conveyed by the differential orbital elements in Table 25.



(a) Torus Accuracy

(b) Angle Residuals

(c) Coordinate & Range Errors

(d) Percent Range Error

Figure 72: Case 08C NLS Results (Numerical *A* Matrix & Torus Ref. Orbit). The ECEF coordinate residuals from the torus orbit fitting process are shown in (a) where x, y and z are indicated by blue, green and red, respectively. The angle residuals are shown in (b) where α is indicated by the black curve (typically obscured by β) and β is indicated by the red curve. The relative position coordinate and range errors are shown in (c) where the radial direction is indicated in red, the along-track direction is indicated in green, the cross-track direction is indicated in blue and the range is indicated in black. Finally, the percent range error is presented in (d). The red curve designates the IOD solution and the blue curve designates the NLS solution.

Table 25: Case 08C Differential Orbital Elements for Truth, IOD & NLS

	δa (km)	δe (nd)	δi (°)	$\delta\Omega$ (°)	$\delta\omega$ (°)	$\delta\nu$ (°)
Truth	1.00000	0.00500	0.20000	0.10000	130.400	-130.000
IOD	1.04821	0.00495	0.19320	0.07963	130.000	-129.583
NLS*	0.99402	0.00496	0.19896	0.09948	130.340	-129.936
NLS†	0.99960	0.00500	0.20006	0.10003	130.404	-129.997
NLS⋆	1.00494	0.00499	0.19972	0.09985	130.388	-129.982
NLS‡	1.01202	0.00503	0.20083	0.10041	130.452	-130.044

* Numerical ***A*** matrix & torus reference orbit

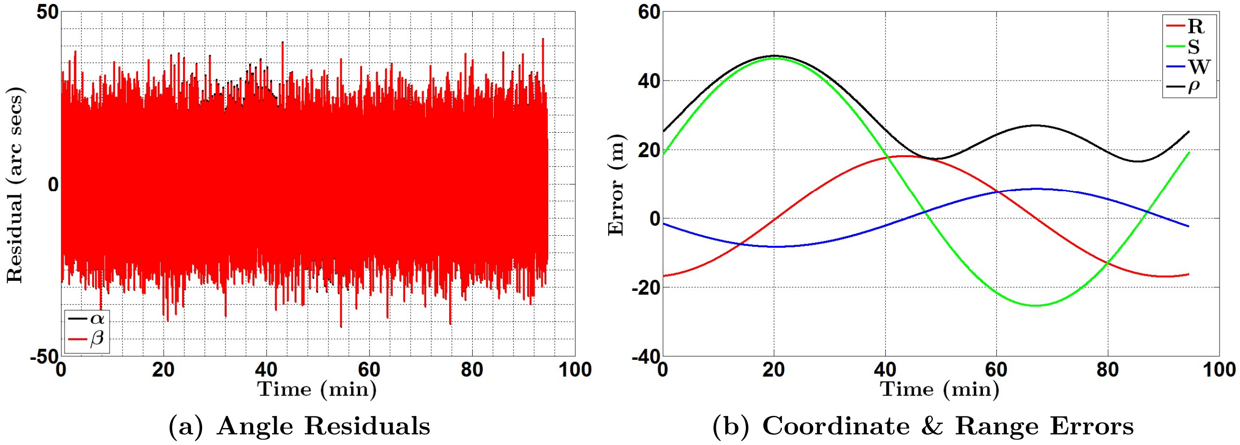
† Numerical ***A*** matrix & numerically integrated reference orbit

⋆ Two-Body ***A*** matrix & torus reference orbit

‡ Two-Body ***A*** matrix & numerically integrated reference orbit

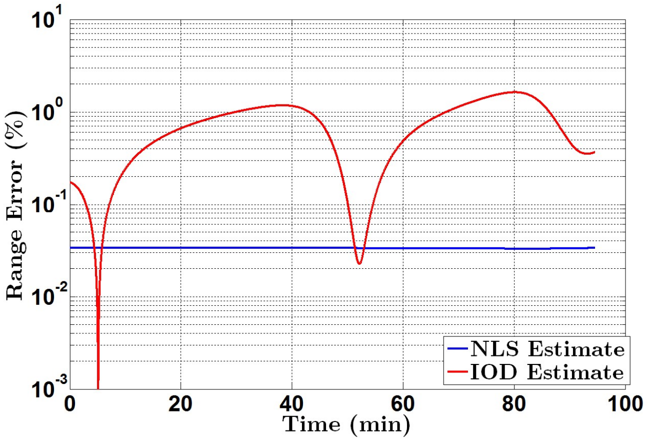
Estimation with the Numerical *A* Matrix & Integrated Reference Orbit

The NLS filter results are shown in Figure 73 using the numerical ***A*** matrix for local motion dynamics and a numerically integrated reference orbit. The RMS of the angle residuals are 10.03 and 10.03 arc sec in α and β, respectively. The accuracy of the estimate in Hill's frame is shown in Figure 73(b) where the range error peaks at 47 m. The percent range error holds relatively steady at ~0.033% of the true range, which is significantly better than the last trial with the torus.



(a) Angle Residuals

(b) Coordinate & Range Errors

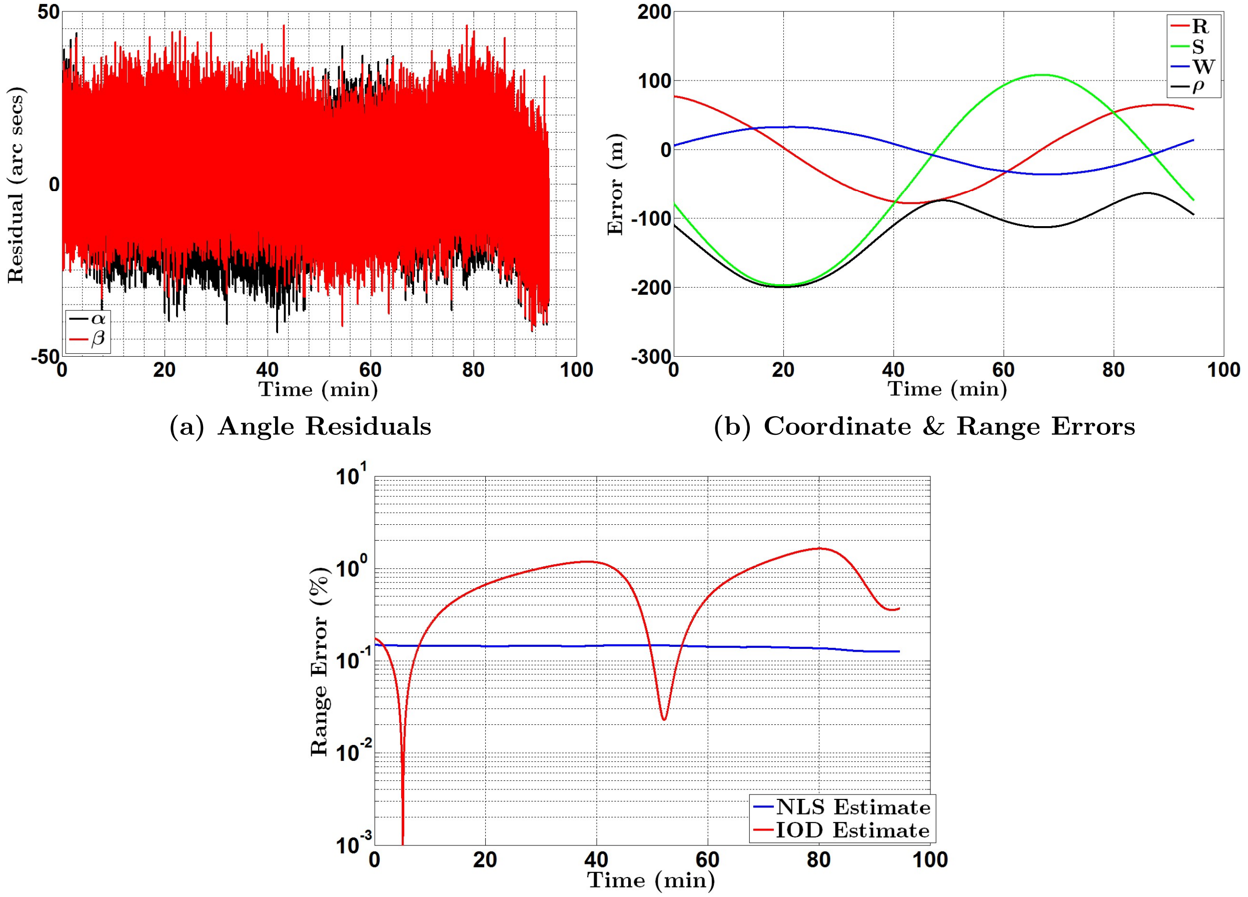


(c) Percent Range Error

Figure 73: Case 08C NLS Results (Numerical $\boldsymbol{A}$ Matrix & Integrated Ref. Orbit). The angle residuals are shown in (a) where α is indicated by the black curve (obscured by β) and β is indicated by the red curve. The relative position coordinate and range errors are shown in (b) where the radial direction is indicated in red, the along-track direction is indicated in green, the cross-track direction is indicated in blue and the range is indicated in black. Finally, the percent range error is presented in (c). The red curve designates the IOD solution and the blue curve designates the NLS solution.

Estimation with the Two-Body $\boldsymbol{A}$ Matrix & Torus Reference Orbit

Results for the analytical two-body $\boldsymbol{A}$ matrix with the IOD reference torus are shown in Figure 74 where we have left out the torus construction residual since it has not changed. The RMS of the angle residuals are 10.67 and 11.33 arc sec in α and β, respectively. The accuracy of the estimate in Hill's frame is shown in Figure 74(b) where the range error peaks at 200 m. The percent range error holds relatively steady at ~0.14% of the true range.



(a) Angle Residuals

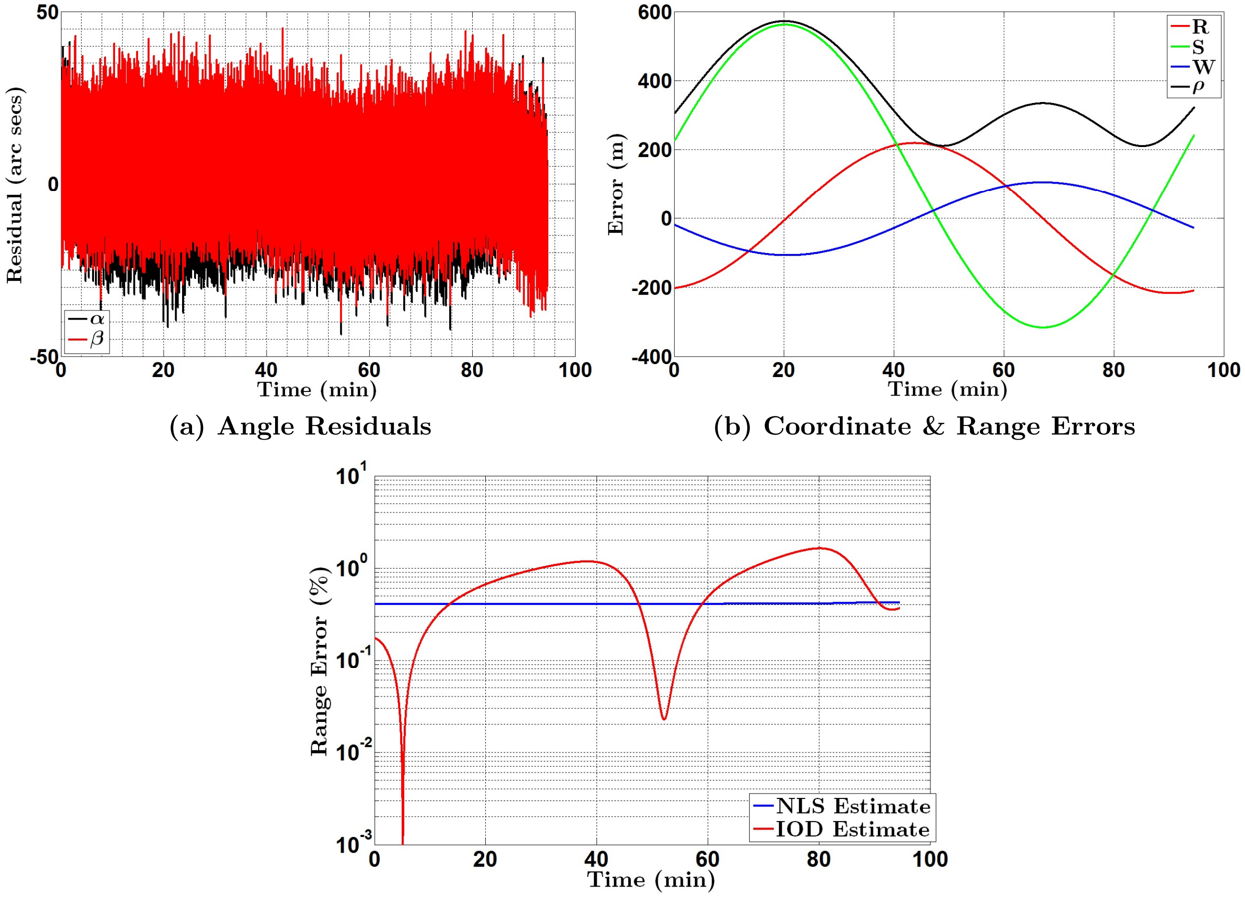
(b) Coordinate & Range Errors

(c) Percent Range Error

Figure 74: Case 08C NLS Results (Two-Body *A* Matrix & Torus Ref. Orbit). The angle residuals are shown in (a) where α is indicated by the black curve and β is indicated by the red curve. The relative position coordinate and range errors are shown in (b) where the radial direction is indicated in red, the along-track direction is indicated in green, the cross-track direction is indicated in blue and the range is indicated in black. Finally, the percent range error is presented in (c). The red curve designates the IOD solution and the blue curve designates the NLS solution.

Estimation with the Two-Body *A* Matrix & Integrated Reference Orbit

One final test is performed using the analytical two-body $\boldsymbol{A}$ matrix and a numerically integrated reference orbit. The results are shown in Figure 75. The RMS of the angle residuals are 10.31 and 11.13 arc sec in α and β, respectively. The accuracy of the estimate in Hill's frame is shown in Figure 75(b) where the range error peaks at 572 m. The percent range error holds relatively steady at ~0.41% of the true range.



(a) Angle Residuals

(b) Coordinate & Range Errors

(c) Percent Range Error

Figure 75: Case 08C NLS Results (Two-Body A Matrix & Integrated Ref. Orbit). The angle residuals are shown in (a) where α is indicated by the black curve and β is indicated by the red curve. The relative position coordinate and range errors are shown in (b) where the radial direction is indicated in red, the along-track direction is indicated in green, the cross-track direction is indicated in blue and the range is indicated in black. Finally, the percent range error is presented in (c). The red curve designates the IOD solution and the blue curve designates the NLS solution.

Chapter 7

Conclusions

This thesis has established a solution to angles-only navigation for natural motion proximity operations with another space object—a problem that has evaded theoreticians and engineers alike since the Gemini and Apollo programs. Natural motion trajectories in space have been heavily studied in the context of angles-only navigation, but the weakly observable conditions of the range regime between approximately 200 m and 100 km have hampered classical angles-only orbit determination techniques and modern estimation methods. Prior to this thesis, the solution options in this regime required either Surveyor actuation (thrusting or spinning) or some form of prior knowledge of the Target spacecraft (i.e., kinematic state or physical characteristics). This thesis has expanded the solution frontier to include natural motion circumnavigation without actuation and without prior knowledge of the Target—the first of its kind.

An analytical solution was developed that proves how range can be mapped from angle and angle-rate curves of NMC trajectories within a multiple hypothesis framework. In particular, this thesis showed that a set of admissible trajectory hypotheses (formulated with nonlinear dynamics in the full geopotential) will exhibit variations in α ($\Delta\alpha$) at common $\dot{\alpha}$ waypoints. The observed α at a particular $\dot{\alpha}$ is contrasted with the admissible hypotheses where range is known. The observability conditions are optimal when $\Delta\alpha$ is largest in the multiple-hypothesis set. The ability to decipher range from this angle-mapping technique led to a novel numerical solution for performing IOD without prior information about the Target and without the cost of thrusting. The algorithms for the numerical solution were tested with simulated data from NASA's General Mission Analysis Tool.

The test cases demonstrated station-keeping and drifting NMC trajectories at LEO and GEO with compelling results. If the argument of perigee is approximated within a fraction of one degree, our novel IOD solution tends to be within a few percent of the truth and is more than sufficient to seed a nonlinear least squares estimator. Test case 07C showed the importance of using fine intervals in our heuristic search for argument of perigee and iterations in our IOD process. Coarse approximations of argument of perigee were shown to introduce unacceptable scaling errors during IOD. A tainted IOD solution can be partially salvaged by batch nonlinear least squares, but the accuracy of the differential correction is inextricably linked to the accuracy of the IOD solution and the fidelity of the dynamics model.

Our nonlinear least squares filter incorporated the full geopotential in the dynamics model through a novel application of KAM theorem. The literature suggested that the secular effects of J_2 dynamics improve range observability, so we went further by including the full geopotential in a reference KAM torus and the equations of variation that describe the local motion. The added fidelity features secular zonal effects, short and long period effects, and the sectoral terms that matter at GEO. Perhaps more importantly, the torus enables precise time synchronization with Target observations. Operationally, it is believed that numerical integration will not perform as well as a torus since the integration time steps will have to be perfectly synchronized with the observations. When observations are made at a high enough sample rate, our Hamiltonian-based STM can be made to closely follow the nonlinear dynamics exquisitely preserved by the torus. The results indicate that there is value in a high fidelity, numerical $\boldsymbol{A}$ matrix at LEO, but it comes at a computational cost and it does not make sense unless the KAM torus is highly accurate. Uncertainty in the torus reference orbit masks any improvements that can be gained from the numerical $\boldsymbol{A}$ matrix over the analytical two-body $\boldsymbol{A}$ matrix. This was confirmed by using an exact numerically integrated reference orbit instead of the KAM torus. However, the concept of using a torus is still preferable if the torus-fitting algorithm can be improved since it synchronizes the reference dynamics with observations. At GEO, the analytical two-body $\boldsymbol{A}$ matrix is preferred with a reference orbit in the full geopotential. The reference orbit already contains the global effects of the full geopotential including J_2 and the sectoral terms that cause a longitudinal drift at GEO, so any additional benefit from the numerical $\boldsymbol{A}$ matrix at GEO was indiscernible over the course of a single orbit.

This thesis provided several other important conclusions about angles-only navigation during NMC proximity operations. First, range observability varies throughout the orbit, as measured by $\Delta\alpha$ at a common $\dot{\alpha}$ waypoint in our multiple hypothesis framework. In general, for station-keeping orbits, we have identified the best observability near the global $\dot{\alpha}_{\min}$ and the worst observability near $\dot{\alpha}_{\max}$. The observability conditions change with drifting NMC trajectories. For instance, in test case 05C at GEO, the best observability conditions were found between the upper $\dot{\alpha}_{\max}$ and the global $\dot{\alpha}_{\min}$ and not near the global or the local $\dot{\alpha}_{\min}$. Additionally, our novel range resolvability metric is an important tool for being able to identify optimal conditions for mapping range from admissible hypotheses. The denominator in the resolvability metric, $\Delta\alpha$, tells us about the observability conditions at a particular $\dot{\alpha}$ waypoint and allows us to select waypoints along large arcs of the trajectory that have good observability and range resolvability for a particular sensor.

Another important conclusion is that eclipse events are not practical for disambiguating range under real-world conditions in the cases we have studied. In theory, the admissible hypotheses exhibit distinct temporal differences between entering and exiting a conical umbra model, but the variations are on the order of seconds. In practice, the visual magnitude will change gradually through the terminator and the uncertainty of a Target's physical characteristics and orientation with respect to the Sun make it impossible to infer range over these short timescales. For now, our novel angle-to-range maps are the only viable solution for determining range without prior knowledge of the Target and without Surveyor actuation.

But what does our solution mean to the world and how can it be utilized? Recall that one of the motivations for this research is to enhance local space situational awareness for peaceful and sustainable endeavors in space. From a capabilities perspective, our solution to angles-only navigation means that an institutional or even national deputization strategy for enhanced LSSA may be possible on government and industry satellites. The extent of new information gained (what can be seen that we did not see before and, from this, what can we know that we did not know before) and the impact on space operations has yet to be explored. Strategically, this tactical navigation capability may have a deterrent effect on nefarious RPO activities that could be demonstrating, maturing or operationalizing anti-satellite technology. At a minimum, our solution gives decision makers greater knowledge about the presence, behavior and kinematic state of proximal RSOs that may be unobservable from standard SSN coverage. Even more, one can imagine how formation flying missions may be conducted with minimal fuel usage to maintain configuration using our angles-only navigation solution. In the case of autonomous or manned rendezvous missions, our technique is also a valuable tool for planning fuel-optimal maneuvers, especially if the Target is unknown and non-cooperative.

7.1 Future Work

Several opportunities to improve or expand this work have been identified. Perhaps the most significant task is a system-level study on the suitability of using our angles-only navigation solution. Operationalizing our technique requires more than just a software update on existing satellites that wish to use it for LSSA. What are the impacts to the primary mission and the CONOPs? What about system-level resource apportionment or hardware capacity? At a minimum, candidate systems must meet two distinct conditions: 1) have an electro-optical payload or star tracker and 2) have a flight computer with sufficient computational and electrical power. Each of these will spawn lower-level requirements that are easy to address during the design phase, but may be restrictive for existing satellites already on-orbit. As it relates to the first condition, the current work assumed zero-mean Gaussian noise in the angle measurements with $\sigma = 10$ arc sec, but how does our technique perform with different sensors? Must we impose some observation quality lower limit? What about the quantity of observations? We assumed ideal lighting conditions in this work, but in practice, the lighting conditions and CONOPs will restrict which existing and future satellites could practically employ our methodologies. This also links back to the camera in terms of the instantaneous field of view if the CONOPs constrain slewing. As it relates to the second condition, what are the limiting factors? Even if the computational requirements can be physically met, how does this influence the electrical power design? For instance, if using secondary batteries, how much greater is the depth of discharge and how does this affect the lifecycle? Do other subsystems need to compensate for the increased demands? How? Understanding the suitability and outcomes from our methodology across a diverse set of fielded space systems is the biggest barrier to its immediate utility.

A significantly less involved task is to consider changing $\hat{\boldsymbol{\rho}}_1$—a component of the nondimensional admissible region parameters—to point toward the approximate argument of perigee after it is found. Presently, $\hat{\boldsymbol{\rho}}_1$ is defined in the direction of the upper relative apsis. The original definition was born from case studies with small differentials in argument of perigee, but the eccentricity approximation is degraded with large differentials in argument of perigee when $\hat{\boldsymbol{\rho}}_1$ is not pointing to perigee as described in Section 5.3.1. If and once $\hat{\boldsymbol{\rho}}_1$ is changed, work should be done to study larger eccentricities. This thesis has focused solely on nearly circular orbits. It is believed that the geometric constraints impossed by the admissible region parameters are extendable to modest eccentricities, but it is not known where our IOD technique breaks down.

A third task is to study how range observability changes throughout an orbit with variations in differential elements. Our methodology only uses $\dot{\alpha}$ waypoints near the most observable $\dot{\alpha}_{\min}$ to scale the IOD solution, but this is thought to be less desirable for some drifting NMC configurations. A parametric study of the differential elements should be undertaken to identify characteristics of observability and resolvability in different orbit configurations. This study should be repeated at several altitudes to show a trend across orbital regimes.

The fourth task pertains to the KAM torus. Our results indicate that the hard-won gains in the numerical $\boldsymbol{A}$ matrix are lost if the reference orbit is not just as accurate. Wiesel has been actively working on torus orbit fitting approaches for more than a decade and has recently developed a low eccentricity theory for satellite orbits that may be able to improve the torus for the test cases we considered, particularly resonant orbits at GEO [156]. Additionally, one may consider improvements to the reference model and STM with third-body effects, and even non-conservative forces such as solar radiation and air drag.

The fifth task is to consider how the principles and general observations of this body of work can be extended to other non-NMC relative orbits. What parameters need to be used to specify the admissible region for a multiple hypothesis framework? Once a set of admissible hypotheses are formed, the angle vs. angle-rate curves can be studied to look for opportunities to disambiguate range with a similar range mapping technique.

Finally, a little more time (no pun intended) should be given to the study of eclipse-to-range maps. Although there is no evidence yet that this technique is operationally viable, there has been no effort to generate orbit configurations that maximize the differential time between eclipse events in a multiple hypothesis framework. There may be greater utility from this method with non-NMC trajectories such as the daunting v-bar station-keeping configuration or a co-elliptical trajectory. Additionally, an optimal configuration between the relative trajectory orientation and the sun vector may produce larger photometry differences than have currently been observed. If this were the case, then it would be helpful to quantify the temporal resolution of range under optimal conditions.

List of Symbols

0	Subscript Indicates Epoch
$\mathbf{0}_k$	$k \times k$ Matrix of Zeros
$\odot$	Subscript Indicates Sun
$\oplus$	Subscript Indicates Earth
a	Semi-Major Axis
aux	Subscript Indicates an Auxiliary Orbit
b	Sub/Superscript Indicates the Body Frame
d	Skewness Factor
e	Eccentricity, Superscript Indicates Earth-Fixed Frame
f	Focal Length, Numerically Integrated Coordinate Data
$\boldsymbol{f}$	Perturbation Vector
$\boldsymbol{g}$	Gradient Vector
h	Sub/Superscript Indicates Hill's Frame
h/i	Subscript Indicates Hill's Frame with Respect to Inertial Frame
i	Inclination, Superscript Indicates the Inertial Frame, Subscript Indicates Iteration
$\hat{\boldsymbol{\imath}}$	Basis Vector Along x-axis
$\hat{\boldsymbol{\jmath}}$	Basis Vector Along y-axis
$\hat{\boldsymbol{k}}$	Basis Vector Along z-axis
m	Range Resolution Metric
m	Subscript Indicates Order of Geopotential Expansion in Spherical Harmonics
n	Mean Motion, Subscript Indicates Degree of Geopotential Expansion in Spherical Harmonics
$\hat{\boldsymbol{n}}$	Line of Nodes
o	Superscript Indicates the Camera Frame
obs	Subscript Indicates an Observation
$\boldsymbol{p}$	Generalized Momenta Vector
p_v	Penumbra Vertical Distance in Eclipse Model
pen	Subscript Indicates Penumbra
$\boldsymbol{q}$	Generalized Coordinates Vector
r	Schneider's Shape Parameter,
$\boldsymbol{r}$	Residual Vector
$\mathbf{r}$	Position Vector
$\hat{\mathbf{r}}$	Unit Position Vector, Radial Basis Vector in Hill's Frame
ref	Subscript Indicates a Reference
rs	Refers to the Surveyor's Orbital Plane Along the $\hat{\mathbf{r}}$- and $\hat{\boldsymbol{s}}$-axis
s	Subscript Indicates Surveyor
$\hat{\boldsymbol{s}}$	Along-Track Basis Vector in Hill's Frame
s_h	Horizontal Distance Along the Earth-Sun Line in Eclipse Model
s_v	Vertical Distance Perpendicular to the Earth-Sun Line in Eclipse Model

sw	Refers to the Tangential Cross-Track Plane Along the $\hat{\mathbf{s}}$- and $\widehat{\boldsymbol{w}}$-axis
t	Time, Subscript Indicates Target
t/h	Subscript Indicates Target with Respect to Hill's Frame
t/s	Subscript Indicates Target with Respect to Surveyor
u	Argument of Latitude
u_v	Umbra Vertical Distance in Eclipse Model
umb	Subscript Indicates Umbra
$\mathbf{v}$	Velocity Vector
vs	Superscript Indicates Virtual Surveyor
$\widehat{\boldsymbol{w}}$	Cross-Track Basis Vector in Hill's Frame
wp	Subscript Indicates Waypoint
x	Cartesian Coordinate, Earth Primary/Secondary Point in Eclipse Model
y	Cartesian Coordinate, Earth-Vertex Distance in Eclipse Model
z	Cartesian Coordinate
$\dot{x}$	Cartesian Velocity Component
$\dot{y}$	Cartesian Velocity Component
$\dot{z}$	Cartesian Velocity Component
$\boldsymbol{A}$	Real-Valued $n \times n$ Coefficient Matrix
$\boldsymbol{C}_k$	Fourier Coefficients Vector
$\bar{C}_{n,m}$	Full Normalized Dimensionless Gravity Model Coefficients
$\boldsymbol{D}_k$	Complex Series Coefficients
E	Eccentric Anomaly, Expectation
$\boldsymbol{F}$	Constant Matrix of Eigenvectors
$\mathscr{F}$	Fourier Transform
$\boldsymbol{G}$	Hessian Matrix
G_R	Receiver Gain
G_T	Transmitter Gain
$\boldsymbol{H}$	Orbit Angular Momentum, Measurement Sensitivity Matrix
$\mathscr{H}$	Hamiltonian
I	Symplectic Action Variable
$\boldsymbol{I}_k$	$k \times k$ Identity Matrix
J_2	Dominant Zonal Term in Earth's Gravitational Field
$\boldsymbol{\mathscr{J}}$	Diagonal Matrix of Eigenvalues
$\mathscr{K}$	Hamiltonian
L	System Losses
$\mathscr{L}$	Lagrangian
M	Mean Anomaly
$\boldsymbol{M}$	Measurement Vector
P	Spectral Power
$\boldsymbol{P}$	Variance-Covariance Matrix, Torus Momenta Vector
$\bar{P}_{n,m}$	Fully Normalized Legendre Polynomials
P_R	Received Power
P_T	Transmitted Power
$\boldsymbol{Q}$	Observation Variance-Covariance Matrix, Torus Angle Variables Vector
$R_\oplus$	Earth Radius
$\boldsymbol{R}^{ab}$	3×3 Rotation Matrix that Transforms Coordinates from Frame b to Frame a
R_R	Distance from Receiver to Target

R_T	Distance from Transmitter to Target
$\boldsymbol{S}_k$	Fourier Coefficients Vector
$\bar{S}_{n,m}$	Full Normalized Dimensionless Gravity Model Coefficients
T	Kozai Orbital Period, Specific Kinetic Energy, Time Interval
$\boldsymbol{T}$	Observation/Observability Matrix
V	Potential Energy
$\boldsymbol{X}$	Cartesian State Vector
$\widetilde{\boldsymbol{X}}$	Estimated Cartesian State Vector
χ_p	Hanning Window Function of Order p
$\boldsymbol{Z}$	Symplectic Matrix
α	IP Angle Observation (Hill's Frame)
α_0	IP Angle at Epoch
α_{pen}	Penumbra Angle in Eclipse Model
α_{umb}	Umbra Angle in Eclipse Model
β	OOP Angle Observation (Hill's Frame)
δ	Schneider's IP Angle
δa	Differential Semi-Major Axis
δe	Differential Eccentricity
δi	Differential Inclination
$\delta\boldsymbol{r}$	Relative Position Difference
δs	Measurement Model Sensitivity Metric
δu	Differential Argument of Latitude
$\delta\widetilde{\boldsymbol{X}}(t_0)$	Estimated Correction to the Reference State Vector at Epoch
$\delta\theta$	System-Level Angle Resolution
$\delta\lambda$	ROE Relative Mean Longitude
$\delta\nu$	Differential True Anomaly
$\delta\rho$	Woffinden's Detectability Range Error Metric, Small Variation in Range, Range Resolution
$\delta\omega$	Differential Argument of Perigee
$\delta\Omega$	Differential RAAN
ε	Small Real-Valued Perturbing Parameter
θ	ROE Phase Angle, Symplectic Angle Variable, Angle Between ${}^{i}\mathbf{r}$ and ${}^{i}\mathbf{v}$, Greenwich Mean Sideral Time, Colatitude
λ	Wavelength, Longitude
μ	Gravitational Parameter
ν	True Anomaly, Cycle Frequency
ρ	Range or Relative Distance
$\hat{\boldsymbol{\rho}}$	LOS Vector in Hill's Frame
σ	RCS, Schneider's OOP Angle, Standard Deviation
ς	Angle Between Negative Sun Vector and Satellite's Position Vector
$\bar{\tau}$	Dimensionless Time Interval,
φ	ROE phase angle
ψ	Geocentric Latitude
ω	Argument of Perigee, Angular Frequency
ω_1	Anomalistic Frequency
ω_2	Nodal Frequency
ω_3	Apsidal Frequency
$\omega_\oplus$	Earth Rotation Rate
$\boldsymbol{\omega}_{a/b}$	Angular Velocity of Frame a with Respect to Frame b

$\Delta\alpha_{\mathrm{T}}$	Angle Disparity in Kozai Period
$\Delta t\|_{\mathrm{eclipse}}$	Time Differential of Eclipse Entry or Exit
$\Delta\mathrm{v}$	Delta-Velocity
$\boldsymbol{\Phi}(t, t_0)$	State Transition Matrix
Ω	RAAN

List of Acronyms

2B	Two-Body
3D	Three-Dimensional
A2/AD	Anti-Access/Area-Denial
AFRL	Air Force Research Laboratory
AFRL/RV	Air Force Research Laboratory's Space Vehicles Directorate
AGI	Analytical Graphics, Inc.
ANGELS	Automated Navigation and Guidance Experiment for Local Space
AON	Angles-Only Navigation
ARGON	Advanced Rendezvous Demonstration Using GPS and Optical Navigation
ASAT	Anti-Satellite
AT&L	Acquisition, Technology and Logistics
AutoNav	Autonomous Navigation
AUV	Autonomous Underwater Vehicles
AVANTI	Autonomous Vision Navigation and Target Identification
BOT	Bearing-Only Tracking
BS	Bachelor of Science
C2ISR	Command and Control, Intelligence, Surveillance and Reconnaissance
C/A	Coarse Acquisition
CCL	Commerce Control List
COE	Classical Orbital Element
CONOPS	Concept of Operations
COTS	Commercial-Off-The-Shelf
CSM	Command and Service Module
CW	Clohessy-Wiltshire
DARPA	Defense Advanced Research Projects Agency
DLR	Deutsches Zentrum für Luft- und Raumfahrt e.V. (German Aerospace Center)
DoD	Department of Defense
DOF	Degrees of Freedom
DRO	Distant Retrograde Orbit
ECEF	Earth-Centered Earth-Fixed
ECI	Earth-Centered Inertial
EGM	Earth Gravity Model
EKF	Extended Kalman Filter
EM	Electromagnetic
EM-1	Exploration Mission 1
EO	Electro-Optical
ESM	Electronic Support Measures
FOV	Field of View
GEO	Geosynchronous Equatorial Orbit
GMAT	General Mission Analysis Tool
GNC	Guidance, Navigation and Control
GPS	Global Positioning System
GPS-R	GPS Reflectometry
GSSAP	Geosynchronous Space Situational Awareness Program
HAX	Haystack Auxiliary Radar

HEO	Highly Elliptical Orbit
HMM	Hidden Markov Model
ICBM	Intercontinental Ballistic Missile
IL	Instrumentation Laboratory
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IOD	Initial Orbit Determination
IP	In-Plane
IR	Infrared
ISS	International Space Station
JPL	Jet Propulsion Laboratory
JSpOC	Joint Space Operations Center
KAM	Kolmogorov-Arnold-Moser
L-Maps	Landmark Maps
LEO	Low Earth Orbit
LiDAR	Portmanteau of Light and Radar
LINCOV	Linear Covariance
LM	Lunar Module
LOR	Lunar Orbit Rendezvous
LOS	Line of Sight
LSD-SLAM	Large-Scale Direct Monocular Simultaneous Localization and Mapping
LSSA	Local Space Situational Awareness
LTV	Linear Time-Varying
LVLH	Local Vertical Local Horizontal
MDP	Markov Decision Process
MIT/LL	Massachusetts Institute of Technology Lincoln Laboratory
MOL	Manned Orbital Laboratory
MP	Modified Polar
MPC	Minor Planet Center
MS	Master of Science
MSFC	Marshall Space Flight Center
NAFF	Numerical Algorithm of the Fundamental Frequency
NASA	National Aeronautics and Space Administration
NLS	Nonlinear Least Squares
NMC	Natural Motion Circumnavigation
NR	Newton-Raphson
NSSS	National Security Space Strategy
OD	Orbit Determination
ONS	One Night Stand
OOP	Out-of-Plane
PhD	Doctor of Philosophy
PLA	People's Liberation Army
PNT	Position, Navigation and Timing
POMDP	Partially Observable Markov Decision Process
PRISMA	Prototype Research Instruments and Space Mission Technology Advancement
PTAM	Parallel Tracking and Mapping
RAAN	Right Ascension of the Ascending Node
RCS	Radar Cross Section
RF	Radio Frequency
RMS	Root Mean Square
ROE	Relative Orbital Elements
RPO	Rendezvous and Proximity Operations
RSO	Resident Space Object

RTAB-Map	Real-Time Appearance-Based Mapping
SBR	Spaceborne Radar
SBSS	Space-Based Space Surveillance
ScD	Doctor of Science
SEL	Single Event Latch-up
SFM	Structure from Motion
SGPS	Spaceborne Global Positioning System
SLAM	Simultaneous Localization and Mapping
SNR	Signal-to-Noise Ratio
SPHERES	Synchronized Position Hold Engage Re-orient Experiment Satellites
SSA	Space Situational Awareness
SSCO	Satellite Servicing Capabilities Office
SSL	Space System's Laboratory
SSN	Space Surveillance Network
STK	System Tool Kit
STM	State Transition Matrix
STS	Shuttle Transportation System
SWaP	Size, Weight and Power
TCFT	Truncated, Continuous Fourier Transform
TEES	Texas A&M Engineering Experiment Station
TID	Total Ionization Dose
TLE	Two-Line Elements
TMA	Target Motion Analysis
UAV	Uninhabited Aerial Vehicle
UD	Unit Upper Triangular Matrix
USAF	United States Air Force
USML	United States Munitions List
UTC	Universal Trajectory Charts
VSA	Very Short Arc

References

[1] Weeden, B., "Dancing in the Dark Redux: Recent Russian Rendezvous and Proximity Operations in Space," *The Space Review*, 5 October 2015.

[2] McCoy, T., and Abbakumova, N., "A Mysterious Russian Space Object Could Be the Return of the 'Satellite Killer,'" *The Washington Post*, 18 November 2014.

[3] Jones, S., "Object 2014-28E: Space Junk or Russian Satellite Killer?" *The Financial Times*, 17 November 2014.

[4] Zak, A., "Kosmos-2499: Is It a Spy or an Assassin... or Both?" *RussianSpaceWeb.com*, 16 June 2016, Web, 15 August 2016 <http://russianspaceweb.com/Cosmos-2499.htm>

[5] Krepinevich, A. F., "Why AirSea Battle," *Center for Strategic and Budgetary Assessments (CSBA)*, Washington, D.C., February 2010.

[6] Van Tol, J., Gunzinger, M., Krepinevich, A., and Thomas, J., "AirSea Battle: A Point-of-Departure Operational Concept," *Center for Strategic and Budgetary Assessments (CSBA)*, Washington, D.C., May 2010.

[7] Van Tol, J., Gunzinger, M., Krepinevich, A., and Thomas, J., "AirSea Battle: A Point-of-Departure Operational Concept" (presentation slides), *Center for Strategic and Budgetary Assessments (CSBA)*, Washington, D.C., May 2010, Web, July 2013, <http://www.csbaonline.org/4Publications/PubLibrary/R.20100518.Slides_AirSea_Batt/R.20100518.Slides_AirSea_Batt.pdf>

[8] National Security Space Strategy Unclassified Summary, Department of Defense and Office of the Director of National Intelligence, January 2011, Web, 24 July 2016, <http://archive.defense.gov/home/features/2011/0111_nsss/docs/NationalSecuritySpaceStrategyUnclassifiedSummary_Jan2011.pdf>

[9] LeMay Center for Doctrine, "Space Operations," Joint Publication 3-14, 29 May 2013.

[10] Shelton, W. L., "Space and Cyberspace - Foundational Capabilities for the Joint Warfighter and the Nation," Air Force Association Air Warfare Symposium, Orlando, FL, 21 February 2014, Speech.

[11] Stansbery, E. G., Matney, M. J., Krisko, P. H., Anz-Meador, P. D., Horstman, M. F., Vavrin, A. B., Jarkey, D. R., Xu, Y., "NASA Orbital Debris

Engineering Model ORDEM 3.0 - Verification and Validation," NASA TP-2015-218592, 1 October 2015.

[12] Gruss, M., "U.S. Space Assets Face Growing Threat from Adversaries, STRATCOM Chief Warns," *SpaceNews.com,* 28 February 2014, Web, 24 July 2016.

[13] Gruss, M., "Space Surveillance Sats Pressed into Early Service," *SpaceNews.com,* 18 September 2015, Web, 24 July 2016.

[14] Swarts, P., "Space Wars: The Air Force Awakens," *Air Force Times* N.p., 15 February 2016, Web, 24 July 2016.

[15] Lohmeyer, W., Cahoy, K., and Baker D. N., "Correlation of GEO Communication Satellite Anomalies and Space Weather Phenomena: Improved Satellite Performance and Risk Mitigation," *30th AIAA International Communications Satellite Systems Conference (ICSSC),* Ottawa, Canada, 2012.

[16] Tyson, P., "Secrets of Ancient Navigators," *NOVA*, 6 October 1998.

[17] Fehse, W., *Automated Rendezvous and Docking of Spacecraft*, Cambridge Aerospace Series 16, Cambridge University Press, New York, 2003.

[18] Uhlig, T., Sellmaier, F., and Schmidhuber, M., eds., *Spacecraft Operations*, Springer, Wien Heidelberg, Germany, 2015.

[19] Woffinden, D. C., and Geller, D. K., "Navigating the Road to Autonomous Orbital Rendezvous," *Journal of Spacecraft and Rockets*, Vol. 44, No. 4, 2007, pp. 898-909.

[20] Kroes, R., and Montenbruck, O., "Spacecraft Formation Flying: Relative Positioning Using Dual-Frequency Carrier Phase," *GPS World*, Vol. 15, No. 7, 2004, pp. 37-42.

[21] Montenbruck, O., and D'Amico, S., "GPS Based Relative Navigation," In *Distributed Space Missions for Earth System Monitoring*, Springer New York, 2013, pp. 185-223.

[22] Kroes, R., Montenbruck, O., Bertiger, W., and Visser, P., "Precise GRACE Baseline Determination Using GPS," GPS Solutions, Vol. 9, No. 1, 2005, pp. 21-31.

[23] TanDEM-X: A New High Resolution Interferometric SAR Mission, DLR Microwaves and Radar Institute, n.d., Web, 26 April 2016 <http://www.dlr.de/hr/en/desktopdefault.aspx/tabid-2317>

[24] Persson, S., D'Amico, S., and Harr, J., "Flight Results from PRISMA Formation Flying and Rendezvous Demonstration," Proceedings of the 61st International Astronautical Congress, Prague, CZ, September 2010.

[25] Bauer, F. H., Moreau, M. C., Dahle-Melsaether, M. E., Petrofski, W. P., Stanton, B. J., Thomason, S., Harris, G. A., Sena, R. P., and Temple III, L. P., "The GPS Space Service Volume," *Proceedings of the 19th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2006)*, Fort Worth, TX, September 2006, pp. 2503-2514.

[26] Ruiz, J. L., and Frey, C. H., "Geosynchronous Satellite Use of GPS," *Proceedings of the 18th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2005)*, Long Beach, CA, September 2005, pp. 1227-1232.

[27] Divis, D. A., "U.S. Eases Export Regulations for GPS Receivers," Inside GNSS, 31 May 2014, Web, 13 May 2017.

[28] Automated Navigation and Guidance Experiment for Local Space (ANGELS) Fact Sheet, *United States Air Force Research Laboratory*, July 2014, Web, 4 August 2016 <http://www.kirtland.af.mil/Portals/52/documents/AFD-131204-039.pdf>

[29] Masters, D. S., *Surface Remote Sensing Applications of GNSS Bistatic Radar: Soil Moisture and Aircraft Altimetry*, Ph.D. thesis, University of Colorado, Boulder, CO, 2004.

[30] Gleason, S., *Remote Sensing of Ocean, Ice and Land Surfaces Using Bistatically Scattered GNSS Signals from Low Earth Orbit*, Ph.D. thesis, University of Surrey, Guildford, Surrey, United Kingdom, 2006.

[31] Axelrad, P., "Application of GNSS to Environmental Studies," In *Global Navigation Satellite Systems: Report of a Joint Workshop of the National Academy of Engineering and the Chinese Academy of Engineering*, National Academies Press, April 2012, pp. 179.

[32] Clarizia, M. P., Braca, P., Ruf, C. S., and Willett, P., "Target Detection Using GPS Signals of Opportunity," *Proceedings of the 18th International Conference on Information Fusion (Fusion),* IEEE, July 2015, pp. 1429-1436.

[33] Lindgren, T., Vinande, E., Akos, D., Masters, D., and Axelrad, P., "Measurement of Backscattered GPS Signals," *Proceedings of the 2006 IEEE/ION Position, Location, And Navigation Symposium (PLANS)*, April 2006, pp. 664-669.

[34] Hsu, Y. S., and Lorti, D. C., "Spaceborne Bistatic Radar: An Overview," Communications, Radar and Signal Processing, IEE Proceedings F, Vol. 133, No. 7, 1986, pp. 642-648.

[35] Gaylor, D. E., and Lightsey, E. G., "Effects of Multipath and Signal Blockage on GPS Navigation in the Vicinity of the International Space Station (ISS)," *Navigation*, Vol. 52, No. 2, 2005, pp. 61-70.

[36] Gaylor, D. E., and Lightsey, E. G., "GPS/INS Kalman Filter Design for Spacecraft Operating in the Proximity of the International Space Station," *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, TX, 11-14 August 2003.

[37] Shah, R., Walker, M. A., Voo, J., Garrison, J. L., Stout, P., Pekkarinen, K., and LeJeune, D., "Study of GPS Reflections and Multipath During Hubble Servicing Mission 4 (STS-125)," *Proceedings of the 23rd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2010)*, September 2010, pp. 2743-2749.

[38] Cohen, I., *Relative Navigation for the Hubble Servicing Mission using Reflected GPS Signals*, Master's thesis, University of Maryland, College Park, MD, 2007.

[39] Pogemiller, J. A. *Remote Sensing Using GPS Bistatic Radar with Applications to Low Earth Orbit Nanosatellites,* Master's thesis, University of Minnesota, 2009.

[40] Pogemiller, J. A., Chu, C. C., and Gebre-Egziabher, D., "A GPS Bistatic Radar for Small Satellite Applications," *Proceedings of the 23rd Annual AIAA/USU Conference on Small Satellites,* 2009.

[41] "GPS Receiver Testing," National Instruments, 11 May 2016, Web, 25 September 2016 <http://www.ni.com/white-paper/7189/en/>

[42] "High Sensitivity GPS," Furuno, n.d., Web, 25 September 2016 <http://www.furuno.com/en/gnss/technical/tec_high>

[43] Owen, W. M., "Methods of Optical Navigation," *Proceedings of the AIAA/AAS Spaceflight Mechanics Conference*, Vol. AAS 11-215, New Orleans, LA, February 2011.

[44] Woffinden, D. C., *Angles-Only Navigation for Autonomous Orbital Rendezvous*, Ph.D. thesis, Utah State University, 2008.

[45] Mammarella, M., Rodrigálvarez, M. A., Pizzichini, A., and Montero, A. M. S., "Advanced Optical Terrain Absolute Navigation for Pinpoint Lunar Landing," *Advances in Aerospace Guidance, Navigation and Control*, Springer Berlin Heidelberg, 2011, pp. 419-430.

[46] Van Pham, B., Lacroix, S., and Devy, M., "Vision-based Absolute Navigation for Descent and Landing," *Journal of Field Robotics*, Vol. 29, No. 4, 2012, pp. 627-647.

[47] Lu, T., Hu, W., Liu, C., and Yang, D., "Relative Pose Estimation of a Lander Using Crater Detection and Matching," *Optical Engineering*, Vol. 55, No. 2, February 2016, pp. 023102.

[48] Zanetti, R., and D'Souza, C., "Navigation and Dispersion Analysis of the First Orion Exploration Mission," *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference*, Vail, Colorado, 2015.

[49] Calhoun, P. C., and Dabney, R., "Solution to the Problem of Determining the Relative 6 DOF State for Spacecraft Automated Rendezvous and Docking," *Proceedings of the SPIE's 1995 Symposium on OE/Aerospace Sensing and Dual Use Photonics*, International Society for Optics and Photonics, Vol. 2466, 1995, pp. 175-184.

[50] Bhaskaran, S., "Autonomous Navigation for Deep Space Missions," *Proceedings of the 12th International Conference on Space Operations (SpaceOps 2012)*, Stockholm, Sweden, 2012.

[51] Gaskell, R. W., Barnouin-Jha, O. S., Scheeres, D. J., Konopliv, A. S., Mukai, T., Abe, S., ... and Kawaguchi, J., "Characterizing and Navigating Small Bodies with Imaging Data," *Meteoritics and Planetary Science*, Vol. 43, No. 6, 2008, pp. 1049-1061.

[52] Rayman, M. D., Varghese, P., Lehman, D. H., Livesay, L. L., "Results from the Deep Space 1 Technology Validation Mission," *Acta Astronautica*, Vol. 47, July-November 2000, pp. 475–487.

[53] MacKenzie, D. A., *Inventing Accuracy: A Historical Sociology of Nuclear Missile Guidance*, MIT Press, 1993.

[54] Mindell, D. A., *Digital Apollo: Human and Machine in Spaceflight*. MIT Press, 2011.

[55] Battin, R. H. *Astronautical Guidance*, McGraw-Hill, New York, 1964.

[56] Horn, B. K. P., *Robot Vision*, MIT Press, 1986.

[57] Fitzgibbon, A. W., and Zisserman, A., "Automatic Camera Recovery for Closed or Open Image Sequences," *Proceedings of the European Conference on Computer Vision*, Springer Berlin Heidelberg, June 1998, pp. 311-326.

[58] Pollefeys, M., Koch, R., and Van Gool, L., "Self-calibration and Metric Reconstruction Inspite of Varying and Unknown Intrinsic Camera Parameters," *International Journal of Computer Vision*, Vol. 32, No. 1, August 1999, pp. 7-25.

[59] Horn, B. K. P., and Schunck, B. G., "Determining Optical Flow," *Artificial Intelligence*, Vol. 17, No. 1-3, 1981, pp 185–203.

[60] Lucas, B. D., and Kanade, T., “An Iterative Image Registration Technique with an Application to Stereo Vision,” *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI ‘81)*, Vol. 81, No. 1, April 1981, pp. 674-679.

[61] Menze, M., and Geiger, A., “Object Scene Flow for Autonomous Vehicles,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3061-3070.

[62] Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M. J., and Szeliski, R., “A Database and Evaluation Methodology for Optical Flow,” *International Journal of Computer Vision*, Vol. 92, No. 1, 2011, pp. 1-31.

[63] Geiger, A., Lenz, P., Stiller, C., and Urtasun, R., “The Kitti Vision Benchmark Suite,” 2015, Web, 11 August 2016 <http://www.cvlibs.net/datasets/kitti/index.php>

[64] Smith, R. C., and Cheeseman, P., “On the Representation and Estimation of Spatial Uncertainty,” *The International Journal of Robotics Research*, Vol. 5, No. 4, 1986, pp. 56-68.

[65] Davison, A. J., Reid, I. D., Molton, N. D., and Stasse, O., “MonoSLAM: Real-time Single Camera SLAM,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 6, 2007, pp. 1052-1067.

[66] Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B., “FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem,” *Proceedings of the AAAI National Conference on Artificial Intelligence*, July 2002, pp. 593-598.

[67] Klein, G., and Murray, D. “Parallel Tracking and Mapping for Small AR Workspaces,” *Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, November 2007, pp. 225-234.

[68] Labbé, M., and Michaud, F., “Online Global Loop Closure Detection for Large-scale Multi-session Graph-based SLAM,” *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2014, pp. 2661-2666.

[69] Engel, J., Schöps, T., and Cremers, D., “LSD-SLAM: Large-scale Direct Monocular SLAM,” *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2014, pp. 834-849.

[70] Leonard, J. J., and Durrant-Whyte, H. F., “Simultaneous Map Building and Localization for an Autonomous Mobile Robot,” *Proceedings of the 1991 IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS)*, November 1991, pp. 1442-1447.

[71] Wu, A. D., Johnson, E. N., Kaess, M., Dellaert, F., Chowdhary, G., "Autonomous Flight in GPS-denied Environments Using Monocular Vision and Inertial Sensors," *Journal of Aerospace Information Systems*, Vol. 10, No. 4, 2013, pp. 172-186.

[72] Kendoul, F., "Survey of Advances in Guidance, Navigation, and Control of Unmanned Rotorcraft Systems," *Journal of Field Robotics*, Vol. 29, No. 2, 2012, pp. 315-378.

[73] Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., ... and Lau, K., "Stanley: The Robot that Won the DARPA Grand Challenge," *Journal of Field Robotics*, Vol. 23, No. 9, 2006, pp. 661-692.

[74] Eustice, R., Singh, H., Leonard, J. J., Walter, M. R., and Ballard, R., "Visually Navigating the RMS Titanic with SLAM Information Filters," *Proceedings of the 2005 Robotics: Science and Systems*, June 2005, pp. 57-64.

[75] Engel, J., Stückler, J., and Cremers, D., Large-scale Direct SLAM with Stereo Cameras," *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2015, pp. 1935-1942.

[76] Kim, W. S., Ansar, A. I., Steele, R. D., and Steinke, R. C., "Performance Analysis and Validation of a Stereo Vision System," *Proceedings of the 2005 IEEE International Conference on Systems, Man and Cybernetics*, Vol. 2, October 2005, pp. 1409-1416.

[77] Tweddle, B. E., *Computer Vision Based Navigation for Spacecraft Proximity Operations*, Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 2010.

[78] Tweddle, B. E., *Computer Vision-based Localization and Mapping of an Unknown, Uncooperative and Spinning Target for Spacecraft Proximity Operations*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2013.

[79] Kolb, R. C., and Hollister, F. H., "Bearing-only Target Estimation," *Proceedings of the 1st Asilomar Conference on Circuits and Systems*, 1967, pp. 935-946.

[80] Murphy, D. J., *Noisy Bearings-only Target Motion Analysis*, Ph.D. thesis, Northeastern University, Boston, MA, 1969.

[81] Lindgren, A. G., and Gong, K. F. "Position and Velocity Estimation Via Bearing Observations," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-14, July 1978, pp. 564-577.

[82] van Huyssteen, D., and Farooq, M., "Performance Analysis of Bearings-only Tracking Algorithm," *Proceedings of Acquisition, Tracking, and Pointing XII,*

International Society for Optics and Photonics (SPIE), Vol. 3365, July 1998, pp. 139-149.

[83] Nardone, S. C., and Aidala, V. J. “Observability Criteria for Bearings-only Target Motion Analysis,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 2, 1981, pp. 162-166.

[84] Fogel, E., and Gavish, M., “Nth-order Dynamics Target Observability from Angle Measurements,” *IEEE Transactions on Aerospace Electronic Systems*, Vol. 24, No. 3, May 1988, pp. 305-308.

[85] Jauffret, C., and Pillon, D., “Observability in Passive Target Motion Analysis,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 32, No. 4, 1996, pp. 1290-1300.

[86] Hammel, S. E., and Aidala, V. J., “Observability Requirements for Three-dimensional Tracking Via Angle Measurements,” IEEE Transactions on Aerospace and Electronic Systems, Vol. 2, March 1985, pp. 200-207.

[87] Monahan, R. D., *The Optimal Maneuver for Bearings Only Target Tracking*, Master’s thesis, Royal Military College of Canada, April 1994.

[88] Passerieux, J. M., and Van Cappel, D., “Optimal Observer Maneuver for Bearings-only Tracking,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 34, No. 3, 1998, pp. 777-788.

[89] Oshman, Y., and Davidson, P., “Optimization of Observer Trajectories for Bearings-only Target Localization,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 35, No. 3, 1999, pp. 892-902.

[90] Le Cadre, J. P., and Laurent-Michel, S., “Optimizing the Receiver Maneuvers for Bearings-only Tracking,” *Automatica*, Vol. 35, No. 4, 1999, pp. 591-606.

[91] Sabet, M. T., Fathi, A. R., and Daniali, H. M., “Optimal Design of the Own Ship Maneuver in the Bearing-only Target Motion Analysis Problem Using a Heuristically Supervised Extended Kalman Filter,” *Ocean Engineering*, Vol. 123, 2016, pp. 146-153.

[92] Aidala, V. J., “Kalman Filter Behavior in Bearings-only Tracking Applications,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-15, No. 1, January 1979, pp. 29–39.

[93] Aidala, V. J., and Hammel, S., “Utilization of Modified Polar Coordinates for Bearings-only Tracking,” *IEEE Transactions on Automatic Control*, Vol. 28, No. 3, 1983, pp. 283-294.

[94] Montiel, J. M. M., Civera, J., and Davison, A. J., “Unified Inverse Depth Parametrization for Monocular SLAM,” *Proceedings of the 2006 Robotics: Science and Systems Conference*, Philadelphia, PA, August 2006.

[95] Civera, J., Davison, A. J., and Montiel, J. M., "Inverse Depth Parametrization for Monocular SLAM," *IEEE Transactions on Robotics*, Vol. 24, No. 5, October 2008, pp. 932-945.

[96] Kirubarajan, T., Bar-Shalom, Y., and Lerro, D., "Bearings-only Tracking of Maneuvering Targets Using a Batch-recursive Estimator," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 37, No. 3, 2001, pp. 770-780.

[97] Le Cadre, J. P., and Trémois, O., "Bearings-only Tracking for Maneuvering Sources," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 34, No. 1, 1998, pp. 179-193.

[98] Blackman, S., and Popoli, R., *Design and Analysis of Modern Tracking Systems*, Artech House, Norwood, MA, 1999.

[99] Hassab, J. C., *Underwater Signal and Data Processing*, CRC Press, Boca Raton, FL, 1989.

[100] Kumar, M. N. V. S. S., Modalavalasa, N., Ganesh, L., Prasad, K. S., and Rao, G. S., "A New Approach for Tracking Moving Objects in Underwater Environment," *Current Science*, Vol. 110, No. 7, April 2016, pp. 1315-1323.

[101] Amelin, K. S., and Miller, A. B., "An Algorithm for Refinement of the Position of a Light UAV on the Basis of Kalman Filtering of Bearing Measurements," *Journal of Communications Technology and Electronics*, Vol. 59, No. 6, 2014, pp. 622-631.

[102] Miller, A., and Miller, B., "Tracking of the UAV Trajectory on the Basis of Bearing-only Observations," *Proceedings of the 53rd IEEE Conference on Decision and Control*, December 2014, pp. 4178-4184.

[103] Johnson, E., Calise, A., Watanabe, Y., Ha, J., and Neidhoefer, J., "Real-time Vision-based Relative Navigation," AIAA Guidance, Navigation, and Control Conference, Keystone, CO, Vol. AIAA 2006-6608, 2006.

[104] Zhang, S., and Bar-Shalom, Y., "Efficient Data Association for 3D Passive Sensors: If I have Hundreds of Targets and Ten Sensors (or More)," *Proceedings of the 14th International Conference on Information Fusion (FUSION)*, July 2011, pp. 1-7.

[105] Ezal, K. O., and Agate, C., "Tracking and Interception of Ground-based RF Sources Using Autonomous Guided Munitions with Passive Bearings-only Sensors and Tracking Algorithms," *Proceedings of Acquisition, Tracking, and Pointing XVIII, International Society for Optics and Photonics (SPIE)*, Vol. 5430, July 2004, pp. 23-34.

[106] Tuckness, D. G. and Young, S.-Y., "Autonomous Navigation for Lunar Transfer," *Journal of Spacecraft and Rockets*, Vol. 32, No. 2, March-April 1995, pp. 279-285.

[107] Gaias, G., D'Amico, S., and Ardaens, J. S., "Angles-only Navigation to a Noncooperative Satellite Using Relative Orbital Elements," Journal of Guidance, Control, and Dynamics, Vol. 37, No. 2, March-April 2014, pp. 439-451.

[108] Lovell, T. A., and Lee, T., "Nonlinear Observability for Relative Satellite Orbits with Angles-only Measurements," *24th International Symposium on Space Flight Dynamics,* 2014.

[109] Christensen, R., and Geller D. K., "Spin-Assisted Angles-only Navigation and Control for SmallSats." *Proceedings of the 37th Annual AAS Guidance and Control Conference, Breckenridge, CO*, 2014.

[110] Laplace, P. S., Mémoires de l'Academie Royale des Sciences de Paris, *Collected Works*, Vol. 10, 1780, pp. 93-146.

[111] Gauss, K. F., *Theoria Motus: Theory of the Motion of the Heavenly Bodies Moving About the Sun in Conic Sections*, Little Brown, Boston, 1857, C. H. Davis Translation.

[112] Escobal, P. R., *Methods of Orbit Determination.* Wiley, New York, 1965.

[113] Gooding, R. H., *A New Procedure for Orbit Determination Based on Three Lines of Sight (Angles Only)*, No. DRA-TR-93004, Defence Research Agency, Farnborough, United Kingdom, 1993.

[114] Sabol, C. and Vallado, D., "A Fresh Look at Angles-only Orbit Determination," *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference*, Vol. AAS 99-363, Girdwood, Alaska, 16-19 August 1999.

[115] Bingham, B. E., and Geller, D. K., "Preliminary Orbit Determination for Orbital Rendezvous using Gauss' Method," *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference and Exhibit,* Vol. AAS 07-399, Mackinac Island, Michigan, 19-23 August 2007.

[116] Bingham, B. E. Preliminary Orbit Determination for Orbital Rendezvous Using Gauss' Method, Master's report, Utah State University, Logan, Utah, 2007.

[117] Schmidt, J. K., Geller, D. K., and Chavez, F., "Improving Angles-only Navigation Performance by Selecting Sufficiently Accurate Accelerometers," *Proceedings of the 23rd Annual AIAA/USU Conference on Small Satellites,* 2009.

[118] Vallado, D. A., *Fundamentals of Astrodynamics and Applications,* 4th ed., Microcosm Press, Hawthorne, CA, 2013.

[119] Spasojevic, Z., Personal Interview, 1 July 2016.

[120] Horneman, K. R., Sizemore, A. E., Morton, B. W., Lovell, T. A., Newman, B. A., and Sinclair, A. J., "Angles-Only Initial Orbit Determination: Comparison of Relative Dynamics and Inertial Dynamics Approaches," *Proceedings of the 9th International Workshop on Satellite Constellations and Formation Flying (IWSCFF)*, Boulder, Colorado, 2017.

[121] Lovell, T. A., Personal Interview, 26 August 2016.

[122] NASA, "Apollo 9 Mission Report," MSC-PA-R-69-2, Manned Spacecraft Center, Houston, TX, May 1969.

[123] Pennington, J. E., "Simulators for Space Rendezvous Maneuver and Docking," NASA TM-X-51714, April 1964.

[124] Levin, E., and Ward, J., "Manned Control of Orbital Rendezvous," No. P-1834, *The RAND Corporation*, 20 October 1959.

[125] Brissenden, R. F., Burton, B. B., Foudriat, E. C., and Whitten, J. B., Analog Simulation of a Pilot-controlled Rendezvous," NASA TN D-747, 1961.

[126] Aldrin, Buzz, *Line-of-sight Guidance Techniques for Manned Orbital Rendezvous*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 1963.

[127] Wolowicz, C. H., Drake, H. M., and Videan, E. N., "Simulator Investigation of Controls and Display Required for Terminal Phase of Coplanar Orbital Rendezvous," NASA TN D-511, 1960.

[128] Lineberry, E. C., Jr., Brissenden, R. F., and Kurbjun, M. C., "Analytical and Preliminary Simulation Study of a Pilot's Ability to Control the Terminal Phase of a Rendezvous with Simple Optical Devices and a Timer," NASA TN D-965, 1961.

[129] McDonald, S., "NASA Langley Researchers Recall Work on First Rendezvous in Space," 15 December 2015, Web, 22 August 2016 <http://www.nasa.gov/feature/langley/nasa-langley-researchers-recall-work-on-first-rendezvous-in-space>

[130] NASA, "Gemini Program Mission Report: Gemini XII," MSC-G-R-67-1, Manned Spacecraft Center, Houston, TX, January 1967.

[131] U.S. Department of Defense, Air Force to Develop Manned Orbiting Laboratory [Press release], 10 December 1963, Retrieved from <http://www.nro.gov/foia/declass/mol/6.pdf>

[132] Schneider, A. M., Prussing, J. E., and Timin, M. E., "Manual Techniques for Space Rendezvous and Reentry," Vols. I and II, AFAL-TR-68-300, Air Force Avionics Laboratory, Wright-Patterson Air Force Base, Ohio, November 1968.

[133] Schneider, A. M., Prussing, J. E., and Timin, M. E., "A Manual Method for Space Rendezvous Navigation and Guidance," *Proceedings of the AIAA Guidance, Control, and Flight Dynamics Conference*, Vol. AIAA 68-859, Pasadena, CA, August 1968.

[134] Schneider, A. M., Prussing, J. E., and Timin, M. E., "A Manual Method for Space Rendezvous Navigation and Guidance," Journal of Spacecraft and Rockets, Vol. 6, No. 9, 1969, pp. 998-1006.

[135] Schneider, A. M., "A Linear Algorithm for Determining Relative Orbital State Using Angle Data," Journal of Spacecraft and Rockets, Vol. 7, No. 11, November 1970, pp. 1358-1360.

[136] Schneider, A. K., "Extension of Angle-Only State-Determination to Curvilinear Coordinates," Journal of Spacecraft and Rockets, Vol. 8, No. 12, 1971, pp. 1238-1238.

[137] Statement by Defense Advanced Research Projects Agency (DARPA) Acting Director: Hearings before the Subcommittee on Emerging Threats and Capabilities, of the Committee on Armed Services, U.S. Senate, 106th Cong. (2001) (testimony of Dr. Jane A. Alexander).

[138] Chari, R. J. V., *Autonomous Orbital Rendezvous Using Angles-only Navigation*, Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, May 2001.

[139] Geller, D., "Analysis of the Relative Attitude Estimation and Control Problem for Satellite Inspection and Orbital Rendezvous," *The Journal of the Astronautical Sciences*, Vol. 55, No. 2, 2007, pp. 195-214.

[140] Woffinden, D. C., and Geller, D. K., "Relative Angles-only Navigation and Pose Estimation for Autonomous Orbital Rendezvous," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1455-1469.

[141] Schmidt, J. K., *Analysis of Square-Root Kalman Filters for Angles-only Orbital Navigation and the Effects of Sensor Accuracy on State Observability,* Master's thesis, Utah State University, Logan, UT, 2010.

[142] Wiesel, W. E., *Modern Orbit Determination*, 2nd ed., Aphelion Press, Beavercreek, OH, 2010.

[143] Woffinden, D. C., and Geller, D. K., "Optimal Orbital Rendezvous Maneuvering for Angles-only Navigation," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 4, 2009, pp. 1382-1387.

[144] Woffinden, D. C., and Geller D. K., "Observability Criteria for Angles-only Navigation," *Aerospace and Electronic Systems, IEEE Transactions on*, Vol. 45, No. 3, 2009, pp. 1194-1208.

[145] Gaias, G., D'Amico, S., and Ardaens, J. S., "Angles-only Navigation to a Noncooperative Satellite Using Relative Orbital Elements," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 2, 2014, pp. 439-451.

[146] D'Amico, S., *Autonomous Formation Flying in Low Earth Orbit*, Ph.D. thesis, Delft University, Delft, Netherlands, 2010.

[147] D'Amico, S., Ardaens, J. S., Gaias, G., Schlepp, B., Benninghoff, H., Tzschichholz, T., Karlsson, T., and Jørgensen, J. L., "Flight Demonstration of Non-cooperative Rendezvous Using Optical Navigation," *Proceedings of the 23rd International Symposium on Space Flight Dynamics*, Pasadena, CA, October 2012, pp. 1-15.

[148] Gaias, G., Ardaens, J. S., and Montenbruck, O., "Model of J2 Perturbed Satellite Relative Motion with Time-varying Differential Drag," *Celestial Mechanics and Dynamical Astronomy*, Vol. 123, No. 4, 2015, pp. 411-433.

[149] Ardaens, J. S., and Gaias, G., "Flight Demonstration of Spaceborne Real-Time Angles-Only Navigation to a Noncooperative Target in Low Earth Orbit," *Proceedings of the 9th International Workshop on Satellite Constellations and Formation Flying (IWSCFF)*, Boulder, Colorado, 2017.

[150] Yim, J. R., Crassidis, J. L., and Junkins, J. L., "Autonomous Orbit Navigation of Two Spacecraft System Using Relative Line of Sight Vector Measurements," *Proceedings of the AAS/AIAA Spaceflight Mechanics Conference*, 2004.

[151] Lovell, T. A., and Lee, T., "Nonlinear Observability for Relative Satellite Orbits with Angles-only Measurements," *Proceedings of the 24th International Symposium on Space Flight Dynamics,* 2014.

[152] Sullivan, J., Koenig, A., and D'Amico, S., "Improved Maneuver-free Approach to Angles-only Navigation for Space Rendezvous," *Proceedings of the 26th AAS/AIAA Spaceflight Mechanics Conference*, Napa, CA, February 2016.

[153] Wiesel, W. E., "Earth Satellite Orbits as KAM Tori," *The Journal of the Astronautical Sciences*, Vol. 56, No. 2, 2008, pp. 151-162.

[154] Wiesel, W. E., "KAM Tori Construction Algorithms," *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference*, Wailea, Maui, Hawaii, 17-19 September 2008.

[155] Wiesel, W. E., "Earth Satellite Perturbation Theories as Approximate KAM Tori," *The Journal of the Astronautical Sciences*, Vol. 58, No. 2, 2011, pp. 153-165.

[156] Wiesel, W. E., “Low Eccentricity Earth Satellite KAM Tori,” *The Journal of the Astronautical Sciences*, Vol. 62, No. 3, 2015, pp. 197-211.

[157] Kolmogorov, A. N. “On Conservation of Conditionally Periodic Motions for a Small Change in Hamilton’s Function,” *Dokl. Akad. Nauk SSSR*, Vol. 98, 1954, pp. 527-530.

[158] Arnol’d, V. I. “Proof of a Theorem of A. N. Kolmogorov on the Invariance of Quasi-Periodic Motions Under Small Perturbations of the Hamiltonian,” *Russian Mathematical Surveys*, Vol. 18, No. 5, 1963, pp. 9-36.

[159] Moser, J. “On Invariant Curves of Area-Preserving Mappings of an Annulus,” *Nachr. Akad. Wiss. Göttingen Math.-Phys. Kl.*, Vol. II, 1962, pp. 1-20.

[160] Celletti, A. and Chierchia L., “KAM Tori for N-body Problems: A Brief History,” *Celestial Mechanics and Dynamical Astronomy*, Vol. 95, No. 1, 2006, pp. 117-139.

[161] Celletti, A., Froeschlé, C. and Lega, E., “From Regular to Chaotic Motions through the Work of Kolmogorov,” in *The Kolmogorov Legacy in Physics*, Ed. Livi, R. and Vulpiani, A., Springer-Verlag Berlin Heidelberg, New York, 2003, pp. 33-59.

[162] Poincaré, H., “New Methods of Celestial Mechanics,” Ed. Goroff, D. L., *American Institute of Physics*, Woodbury, NY, 1993.

[163] Wiesel, W. E., *Modern Astrodynamics*, Aphelion Press, Beavercreek, OH, 2003.

[164] Dahlquist, G. and Björck, A., *Numerical Methods*, Dover, 2003.

[165] Clohessy, W. H. and Wiltshire, R. S., “Terminal Guidance for Satellite Rendezvous,” *J. Aerospace Sciences*, Vol. 27, 1960, pp. 653-678.

[166] Abramowitz, M., and Stegun, I. A. (Eds.), *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*, Vol. 55. Courier Corporation, 1964.

[167] Kunz, D. L., “Intermediate Dynamics Course Notes,” MECH 521. Air Force Institute of Technology, Dayton, OH, Fall 2009.

[168] Alfriend, K., Vadali, S. R., Gurfil, P., How, J., and Breger, L., *Spacecraft Formation Flying: Dynamics, Control and Navigation*, Vol. 2, Butterworth-Heinemann, 2009.

[169] Meeus, J., *Astronomical Algorithms*, 2nd ed., Willmann-Bell, Richmond, VA, 1991.

[170] Lemoine, F. G., Kenyon, S. C., Factor, J. K., Trimmer, R. G., Pavlis, N. K., Chinn, D. S., Cox, C. M., Klosko, S. M., Luthcke, S. B., Torrence, M. H., Wang, Y. M., Williamson, R. G., Pavlis, E. C., Rapp R. H., and Olson, T. R.,

"The Development of the Joint NASA GSFC and NIMA Geopotential Model EGM96," NASA/TP-1998-206861, NASA Goddard Space Flight Center, Greenbelt, MD, 1998.

[171] Milani, A., Gronchi, G. F., Vitturi, M. D. M., and Knežević, Z., "Orbit Determination with Very Short Arcs: I. Admissible Regions," *Celestial Mechanics and Dynamical Astronomy*, Vol. 90, No.1-2, September 2004, pp. 57-85.

[172] Hussein, I., Roscoe, C., Wilkins, M., and Schumacher, P., "Probabilistic Admissible Region for Short-arc Angles-only Observations," *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies (AMOS) Conference*, Maui, HI, 9-12 September 2014, Vol. 1, pp. 76.

[173] Linares, R., Jah, M. K., Crassidis, J. L., and Nebelecky, C. K., "Space Object Shape Characterization and Tracking Using Light Curve and Angles Data," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 1, 2013, pp. 13-25.

[174] Kuga, H. K., and Carrara, V., "Fortran-and C-codes for Higher Order and Degree Geopotential and Derivatives Computation," *Proceedings of XVI SBSR-Brazilian Symposium on Remote Sensing*, Iguassu Falls, Brazil, April 2013, pp. 13-18.

[175] Bucha, B., Janák, J., "A MATLAB-Based Graphical User Interface Program for Computing Functionals of the Geopotential Up to Ultra-High Degrees and Orders," *Computers & Geosciences*, Vol. 56, 2013, pp. 186–196.

[176] Fantino, E., and Casotto, S., "Methods of Harmonic Synthesis for Global Geopotential Models and their First-, Second-and Third-Order Gradients," *Journal of Geodesy*, Vol. 83, No. 7, 2009, pp. 595-619.

[177] Tabor, M., *Chaos and Integrability in Nonlinear Dynamics: An Introduction*, Wiley, New York, 1989

[178] Merrit, D., "Non-integrable Galactic Dynamics," *Proceedings of the 54th Scottish Universities Summer School in Physics*, Blair Atholl, 23 July - 5 August 2000, pp. 145-165.

[179] Binney, J., and Spergel, D., "Spectral Stellar Dynamics," *The Astrophysical Journal*, Vol. 252, 1982, pp. 308-321.

[180] Binney, J., and Spergel, D., "Spectral Stellar Dynamics–II. The Action Integrals," *Monthly Notices of the Royal Astronomical Society*, Vol. 206, No. 1, 1984, pp. 159-177.

[181] Laskar, J., "Introduction to Frequency Map Analysis," *in Hamiltonian Systems with Three or More Degrees of Freedom*, edited by Simo, C., Kluwer Academic Publishers, 1999, pp. 134-150.

[182] Laskar, J., “Frequency Map Analysis and Quasiperiodic Decompositions,” *in Hamiltonian Systems and Fourier Analysis: New Prospects for Gravitational Dynamics*, edited by Benest, D., Froeschlé, C., and Lega, E., Advances in Astronomy and Astrophysics, Cambridge Scientific Publishers, Cambridge, UK, 2005.

Appendix A

Nonlinear Measurement Sensitivity Derivation

A.1 Sensitivity in the X Direction

$$x_o = \rho \sin(\alpha) \cos(\beta) \tag{A.1}$$

$$\delta x \approx [\delta\alpha \quad \delta\beta]\boldsymbol{g}_x + \frac{1}{2}[\delta\alpha \quad \delta\beta]\boldsymbol{G}_x \begin{bmatrix}\delta\alpha\\ \delta\beta\end{bmatrix} \tag{A.2}$$

$$\boldsymbol{g}_x = \left(\frac{\partial x}{\partial \boldsymbol{M}}\right)^T = \begin{bmatrix}\partial x/\partial\alpha\\ \partial x/\partial\beta\end{bmatrix} = \begin{bmatrix}\rho\cos(\alpha)\cos(\beta)\\ -\rho\sin(\alpha)\sin(\beta)\end{bmatrix} \tag{A.3}$$

$$\boldsymbol{G}_x = \begin{bmatrix}\partial^2 x/\partial\alpha^2 & \partial^2 x/\partial\alpha\partial\beta\\ \partial^2 x/\partial\beta\partial\alpha & \partial^2 x/\partial\beta^2\end{bmatrix} = \begin{bmatrix}-\rho\sin(\alpha)\cos(\beta) & -\rho\cos(\alpha)\sin(\beta)\\ -\rho\cos(\alpha)\sin(\beta) & -\rho\sin(\alpha)\cos(\beta)\end{bmatrix} \tag{A.4}$$

$$\begin{aligned}\delta x \approx{}& \rho\cos(\alpha)\cos(\beta)\,\delta\alpha - \rho\sin(\alpha)\sin(\beta)\,\delta\beta\\ &+\frac{1}{2}[-\rho\sin(\alpha)\cos(\beta)\,\delta\alpha^2 - 2\rho\cos(\alpha)\sin(\beta)\,\delta\alpha\delta\beta\\ &-\rho\sin(\alpha)\cos(\beta)\,\delta\beta^2]\end{aligned} \tag{A.5}$$

Assuming $\delta\alpha \approx \delta\beta = \mathcal{O}(\varepsilon)$ and $\varepsilon \ll 1$, substitute and simplify δx for three different forms:

$$\mathcal{O}(\delta x) \approx z\varepsilon - y\sin(\alpha)\,\varepsilon - x\varepsilon^2 - y\cos(\alpha)\,\varepsilon^2 \tag{A.6}$$

$$\mathcal{O}(\delta x) \approx \rho[\cos(\alpha+\beta)\,\varepsilon - \sin(\alpha+\beta)\,\varepsilon^2] \tag{A.7}$$

$$\mathcal{O}(\delta x) \approx z\varepsilon - \frac{xy}{\sqrt{x^2+z^2}}\varepsilon - x\varepsilon^2 - \frac{yz}{\sqrt{x^2+z^2}}\varepsilon^2 \tag{A.8}$$

We make a very small error by ignoring the second order terms when $\varepsilon \ll 1$ so that $\mathcal{O}(\delta x)$ may be approximated as

$$\mathcal{O}(\delta x) \approx \rho\cos(\alpha+\beta)\,\varepsilon \approx z\varepsilon - \frac{xy}{\sqrt{x^2+z^2}}\varepsilon \tag{A.9}$$

A.2 Sensitivity in the Y Direction

$$y_o = \rho\sin(\beta) \tag{A.10}$$

$$\delta y \approx [\delta\alpha \quad \delta\beta]\boldsymbol{g}_y + \frac{1}{2}[\delta\alpha \quad \delta\beta]\boldsymbol{G}_y \begin{bmatrix}\delta\alpha\\ \delta\beta\end{bmatrix} \tag{A.11}$$

$$\boldsymbol{g}_y = \left(\frac{\partial y}{\partial \boldsymbol{M}}\right)^T = \begin{bmatrix}\partial y/\partial\alpha\\ \partial y/\partial\beta\end{bmatrix} = \begin{bmatrix}0\\ \rho\cos(\beta)\end{bmatrix} \tag{A.12}$$

$$\boldsymbol{G}_y = \begin{bmatrix}\partial^2 y/\partial\alpha^2 & \partial^2 y/\partial\alpha\partial\beta\\ \partial^2 y/\partial\beta\partial\alpha & \partial^2 y/\partial\beta^2\end{bmatrix} = \begin{bmatrix}0 & 0\\ 0 & -\rho\sin(\beta)\end{bmatrix} \tag{A.13}$$

$$\delta y \approx \rho\cos(\beta)\,\delta\beta - \frac{1}{2}\rho\sin(\beta)\,\delta\beta^2 \tag{A.14}$$

Assuming $\delta\alpha \approx \delta\beta = \mathcal{O}(\varepsilon)$ and $\varepsilon \ll 1$, substitute and simplify δy for two different forms:

$$\mathcal{O}(\delta y) \approx \frac{x}{\sin(\alpha)}\varepsilon - \frac{1}{2}y\varepsilon^2 \tag{A.15}$$

$$\mathcal{O}(\delta y) \approx \sqrt{x^2+z^2}\varepsilon - \frac{y}{2}\varepsilon^2 \tag{A.16}$$

We make a very small error by ignoring the second order terms when $\varepsilon \ll 1$ so that $\mathcal{O}(\delta y)$ may be approximated as

$$\mathcal{O}(\delta y) \approx \frac{x}{\sin(\alpha)}\varepsilon \approx \sqrt{x^2+z^2}\varepsilon \tag{A.17}$$

A.3 Sensitivity in the Z Direction

$$z_o = \rho\cos(\alpha)\cos(\beta) \tag{A.18}$$

$$\delta z \approx [\delta\alpha \quad \delta\beta]\boldsymbol{g}_z + \frac{1}{2}[\delta\alpha \quad \delta\beta]\boldsymbol{G}_z \begin{bmatrix}\delta\alpha\\ \delta\beta\end{bmatrix} \tag{A.19}$$

$$\boldsymbol{g}_z = \left(\frac{\partial z}{\partial \boldsymbol{M}}\right)^T = \begin{bmatrix}\partial z/\partial\alpha\\ \partial z/\partial\beta\end{bmatrix} = \begin{bmatrix}-\rho\sin(\alpha)\cos(\beta)\\ -\rho\cos(\alpha)\sin(\beta)\end{bmatrix} \tag{A.20}$$

$$\boldsymbol{G}_z = \begin{bmatrix} \partial^2 z/\partial\alpha^2 & \partial^2 z/\partial\alpha\partial\beta \\ \partial^2 z/\partial\beta\partial\alpha & \partial^2 z/\partial\beta^2 \end{bmatrix} = \begin{bmatrix} -\rho\cos(\alpha)\cos(\beta) & \rho\sin(\alpha)\sin(\beta) \\ \rho\sin(\alpha)\sin(\beta) & -\rho\cos(\alpha)\cos(\beta) \end{bmatrix} \tag{A.21}$$

$$\begin{aligned} \delta z \approx & -\rho\sin(\alpha)\cos(\beta)\,\delta\alpha - \rho\cos(\alpha)\sin(\beta)\,\delta\beta \\ & + \frac{1}{2}[-\rho\cos(\alpha)\cos(\beta)\,\delta\alpha^2 + 2\rho\sin(\alpha)\sin(\beta)\,\delta\alpha\delta\beta \\ & - \rho\cos(\alpha)\cos(\beta)\,\delta\beta^2] \end{aligned} \tag{A.22}$$

Assuming $\delta\alpha \approx \delta\beta = \mathcal{O}(\varepsilon)$ and $\varepsilon \ll 1$, substitute and simplify δz for three different forms:

$$\mathcal{O}(\delta z) \approx -x\varepsilon - y\cos(\alpha)\,\varepsilon - z\varepsilon^2 + y\sin(\alpha)\,\varepsilon^2 \tag{A.23}$$

$$\mathcal{O}(\delta z) \approx -\rho[\sin(\alpha+\beta)\,\varepsilon + \cos(\alpha+\beta)\,\delta\varepsilon^2] \tag{A.24}$$

$$\mathcal{O}(\delta z) \approx -x\varepsilon - \frac{yz}{\sqrt{x^2+z^2}}\varepsilon - z\varepsilon^2 + \frac{xy}{\sqrt{x^2+z^2}}\varepsilon^2 \tag{A.25}$$

We make a very small error by ignoring the second order terms when $\varepsilon \ll 1$ so that $\mathcal{O}(\delta z)$ may be approximated as

$$\mathcal{O}(\delta z) \approx -\rho\sin(\alpha+\beta)\,\varepsilon \approx -x\varepsilon - \frac{yz}{\sqrt{x^2+z^2}}\varepsilon \tag{A.26}$$

Appendix B

Supplementary Camera Topics

B.1 Angular Resolution

The smallest resolvable distance in an image spans a characteristic 2D angle known as the angular resolution. The 3D world projection onto a 2D image has a spatial resolution in the horizontal and vertical directions (δx and δy) which corresponds to a horizontal and vertical angular resolution—$\delta\alpha = \varepsilon_\alpha$ and $\delta\beta = \varepsilon_\beta$, respectively.

A finite number of pixels in the image sensor is the limiting factor for angular resolution. Consider a camera with a pixel resolution of $n_x \times n_y$ where n_x is the number of pixels in the horizontal direction, and n_y is the number of pixels the vertical direction. The field of view (FOV) of the camera in the horizontal and vertical directions is given by

$$FOV_x = 2\tan^{-1}\left(\frac{pn_x}{2f}\right) \tag{B.1}$$

$$FOV_y = 2\tan^{-1}\left(\frac{pn_y}{2f}\right) \tag{B.2}$$

where p is pixel size or pixel pitch of the image sensor, n is the number of pixels, i.e. for a horizontal resolution of 1024 pixels, $n_x = 1024$ and f is the focal length. The 2-

dimensional geometry showing the relationship between the FOV, range and focal length are shown in Figure 76.

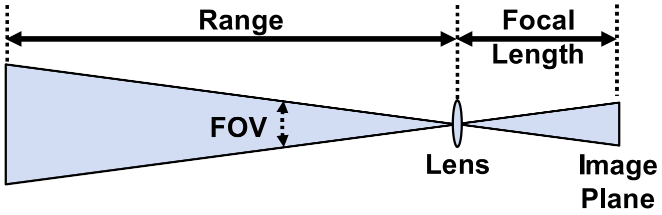


Figure 76: Camera Focal Length & FOV. The extent of the observable space spans the field of view (FOV). The view is projected onto the image plane by a lens that bends incident light rays proportional to their arrival angle.

Light is collected on the sensor's image plane array by a discreet number of solar cells or pixels that convert photons into electrons as crudely depicted in Figure 77.[40] Each one of the pixels represents a fraction of the FOV, which is the angular resolution. If the pixels are square, the spatial and angular resolution is the same in both directions.

$$\varepsilon_\alpha = \frac{FOV_x}{n_x} \tag{B.3}$$

$$\varepsilon_\beta = \frac{FOV_y}{n_y} \tag{B.4}$$

[40] Two common image sensors are the charge coupled device (CCD) and complementary metal oxide semiconductor (CMOS).

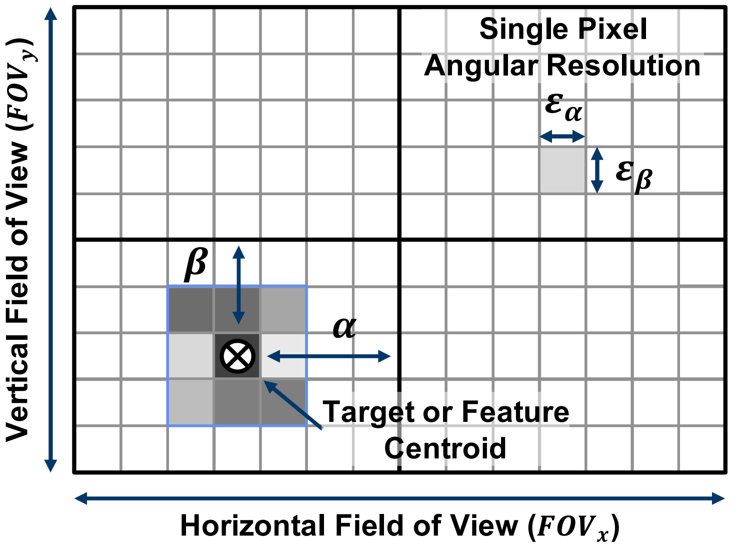


Figure 77: Image Plane Array. The angular resolution of a camera is a function of the FOV and the finite number of pixels on the sensor.

The angular resolution is physically restricted by the lens aperture size or collection area. Larger optics collect more photons and can observe fainter objects faster, but even with a large aperture and infinitely small pixels, angular resolution of the sensor is diffraction limited. In other words, there is a theoretical minimum resolvable distance (the distance between distinguishable objects in the image). The diffraction-limited angular resolution is

$$\varepsilon \cong 1.220\frac{\lambda}{D} \tag{B.5}$$

where λ is the wavelength of light and D is the diameter of the lens aperture. The angular resolution of visible light at the diffraction limit is depicted as a function of aperture diameter in Figure 78. These curves represent the lowest possible angular resolution for a given aperture diameter.

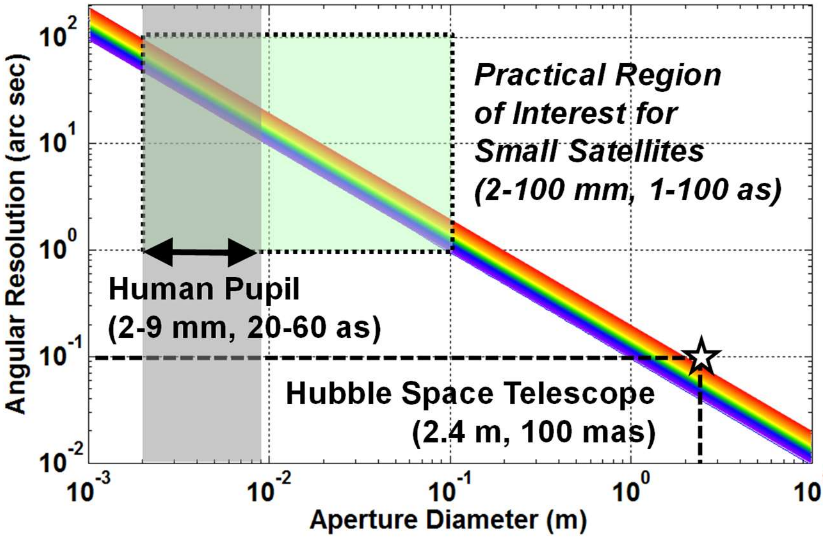


Figure 78: Diffraction Limited Angular Resolution. The angular resolution of visible light as a function of aperture diameter.

Appendix C

Analysis of Cylindrical Model Assumption for Earth Eclipse

C.1 Low Earth Orbit

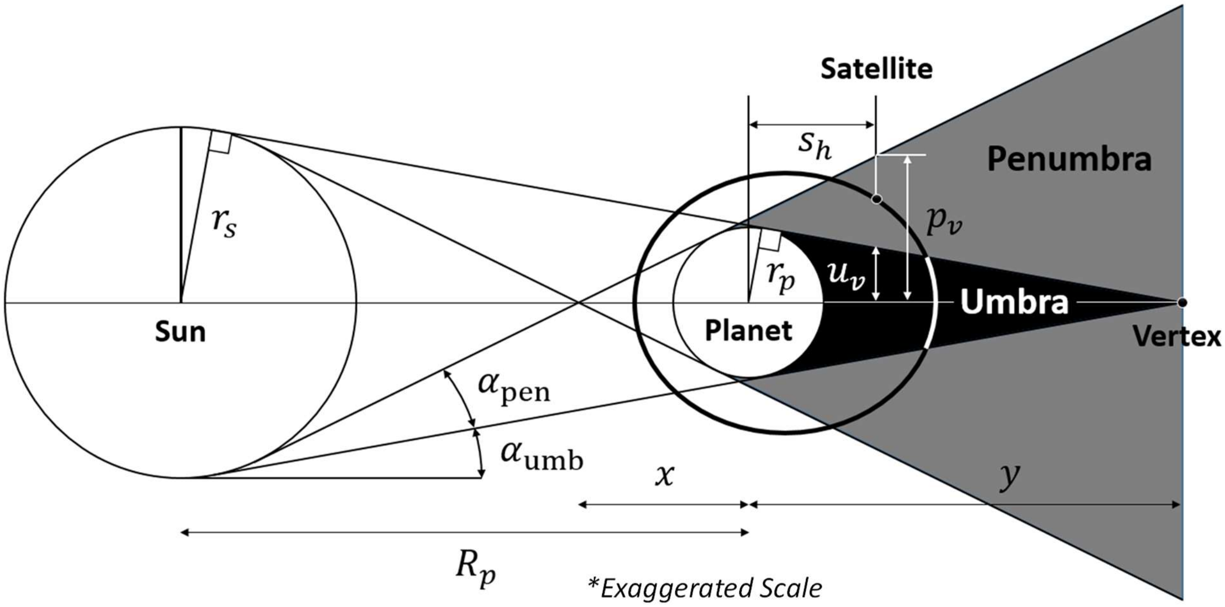


Figure 79: Conical Model for Planetary Eclipse.

To make a cylindrical assumption for the umbra, the distance R_p must be much greater than the radius of the Sun and the Earth such that the geometric angles α_{umb} and α_{pen}

approach zero. We can calculate the angle values as Vallado does ([118] pp. 299) for the Sun-Earth system as follows

$$\sin(\alpha_{\text{umb}}) = \frac{r_s - r_p}{R_p} \tag{C.1}$$

$$\sin(\alpha_{\text{umb}}) = \frac{696000 - 6378}{149599870}$$

$$\alpha_{\text{umb}} = 0.264121687°$$

$$\sin(\alpha_{\text{pen}}) = \frac{r_s + r_p}{R_p} \tag{C.2}$$

$$\sin(\alpha_{\text{pen}}) = \frac{696000 + 6378}{149599870}$$

$$\alpha_{\text{pen}} = 0.269007205°$$

While this bodes well for the cylindrical model assumption, the angles alone are not enough to make the assumption. The other essential piece of information is to check that the delta between p_v and u_v is small at the satellite's farthest eclipsed point from the sun so that we can ignore the relatively small amount of time in the penumbra. Assuming that the satellite is in a circular orbit with an altitude of 500 km, $s_h = r_p +$ 500 km the values of p_v and u_v are

$$p_v = \tan(\alpha_{\text{pen}})\,(x + s_h) \tag{C.3}$$

$$x = \frac{r_p}{\sin(\alpha_{\text{pen}})} \tag{C.4}$$

$$x = \frac{6378}{\sin(0.269007205°)} = 1358453.668 \text{ km}$$

$$p_v = \tan(0.269007205°)\,(1358453.668 + 6878) = 6410.363 \text{ km}$$

$$u_v = \tan(\alpha_{\text{umb}})\,(y - s_h) \tag{C.5}$$

$$y = \frac{r_p}{\sin(\alpha_{\text{umb}})} \tag{C.6}$$

$$y = \frac{6378}{\sin(0.264121687°)} = 1383581.107 \text{ km}$$

$$u_v = \tan(0.264121687°)\,(1383581.107 - 6878) = 6346.361 \text{ km}$$

The difference between p_v and u_v is

$$p_v - u_v = 6410.363 - 6346.361 = 64 \text{ km}$$

Compared to the Earth radius, the quantity $(p_v - u_v)$ is very small—64 km vs 6378 km—so the amount of time spent traversing the penumbra is on the order of seconds. Only now is it safe to make the assumption of a cylindrical umbra for a ballpark answer.

C.2 Geosynchronous Equatorial Orbit

It should be obvious that the cylindrical Earth shadow model breaks down as orbital altitude increases since the penumbral region radius grows and the umbral region radius shrinks toward the vertex. The extent to which this occurs at GEO is explored here. To evaluate the validity of a cylindrical umbra model for a geostationary satellite, we will assume the March or September equinox when the sun is aligned with the equatorial plane.

Assuming the arc of the orbit through the penumbra and umbra is short in duration compared to the 24-hr sidereal day, the path lengths through the penumbra and umbra for geostationary satellites may be reasonably estimated as a straight line. We can use twice the quantity of p_v and u_v as defined previously.

The satellite's furthest point from the sun in a geostationary orbit is by definition $s_h = a = 42164$ km. We can then use this information to calculate p_v and u_v:

$$p_v = \tan(\alpha_{\text{pen}})\,(x + s_h)$$

$$p_v = \tan(0.269007205°)\,(1358453.668 + 42164) = 6576 \text{ km}$$

$$u_v = \tan(\alpha_{\text{umb}})\,(y - s_h)$$

$$u_v = \tan(0.264121687°)\,(1383581.107 - 42164) = 6184 \text{ km}$$

The total path length through the penumbra and umbra are estimated as $2p_v =$ 13152 km and $2u_v = 12367$ km, respectively. Vallado (pp. 299) states that the satellite traverse 13098 km in the penumbral region and 12412 km in the umbral region, so the straight-line approximation is only off by about 0.4%.

Since the satellite travels at a constant orbital velocity in a geostationary orbit, we can just as easily translate these path lengths into times. The period for a geostationary satellite is 24 sidereal hours (a little less than 24 solar hours):

$$P = \frac{86400 \text{ solar sec}}{1 \text{ solar day}} \times \frac{1 \text{ solar day}}{1.002737909350795 \text{ sidereal days}} \tag{C.7}$$

$$P = 86164.090518 \text{ solar sec}$$

Equivalent ratios of time and arc length render the duration of the penumbra crossing, $\Delta\tau_{\text{GEO-pen}}$, and the umbra crossing, $\Delta\tau_{\text{GEO-umb}}$.

$$\frac{\Delta\tau_{\text{GEO-pen}}}{P} = \frac{d_{\text{pen}}}{2\pi a} \tag{C.8}$$

$$\frac{\Delta\tau_{\text{GEO-pen}}}{86164} = \frac{13152}{2\pi(42164)}$$

$$\Delta\tau_{\text{GEO-pen}} = 4278 \text{ sec} = 71.3 \text{ min}$$

$$\frac{\Delta\tau_{\text{GEO-umb}}}{P} = \frac{d_{\text{umb}}}{2\pi a} \tag{C.9}$$

$$\frac{\Delta\tau_{\text{GEO-umb}}}{86164} = \frac{12367}{2\pi(42164)}$$

$$\Delta\tau_{\text{GEO-umb}} = 4022 \text{ sec} = 67.0 \text{ min}$$

Vallado (pp. 299) quantifies the duration in the penumbral and umbral regions for GEO as 71 min and 67 min, respectively, so the straight-line approximation is very reasonable.

Not surprisingly, the duration spent in the penumbral region at GEO is greater than the duration spent in the umbral region at GEO. Thus, the cylindrical model breaks down somewhere before GEO where it becomes necessary to reintroduce the more detailed model of the umbra and penumbra. The precise altitude at which this occurs is entirely dependent on the application and users should beware. A good rule of thumb is to keep the modeling error below 1% and the criterion may be written as

$$\frac{(p_v - u_v)}{r_\oplus} > 0.01 \tag{C.10}$$

We can solve the criterion for s_h algebraically:

$$\begin{aligned} p_v - u_v &> 0.01 r_\oplus \\ \tan(\alpha_{\text{pen}})\,(x + s_h) - \tan(\alpha_{\text{umb}})\,(y - s_h) &> 0.01 r_\oplus \\ s_h &> \frac{0.01 r_\oplus - \tan(\alpha_{\text{pen}})\,x + \tan(\alpha_{\text{umb}})\,y}{\tan(\alpha_{\text{pen}}) + \tan(\alpha_{\text{umb}})} \end{aligned} \tag{C.11}$$

The geometric parameters for the Earth-Sun system were previously shown to be:

$$\begin{aligned} \alpha_{\text{umb}} &= 0.264121687° & x &= 1358453.668 \text{ km} \\ \alpha_{\text{pen}} &= 0.269007205° & y &= 1383581.107 \text{ km} \end{aligned}$$

Thus,

$$s_h > \frac{0.01(6378.137) - \tan(0.269007205°)\,1358453.668 + \tan(0.264121687°)\,1383581.107}{\tan(0.269007205°) + \tan(0.264121687°)}$$

$$s_h > 6854.3 \text{ km}$$

$$h = a - r_{\oplus} \tag{C.12}$$

$$h = 6854.3 - 6378.137$$

$$h = 476 \text{ km}$$

If the altitude of the satellite is greater than 476 km, then the radial distance that the penumbra extends beyond the umbra has grown to 1% of the Earth's radius and a cylindrical Earth shadow model may not be advisable.

Appendix D

Gravitational Second-Order Partial Derivatives in Cartesian Coordinates

D.1 Diagonal Elements

$$\frac{\partial^2 V}{\partial q_x^2} = \frac{\partial}{\partial x}\left[\frac{\partial V}{\partial r}\cdot\frac{\partial r}{\partial x}\right] + \frac{\partial}{\partial x}\left[\frac{\partial V}{\partial \lambda}\cdot\frac{\partial \lambda}{\partial x}\right] + \frac{\partial}{\partial x}\left[\frac{\partial V}{\partial \psi}\cdot\frac{\partial \psi}{\partial x}\right] \tag{D.1}$$

$$\begin{split}\frac{\partial^2 V}{\partial q_x^2} = &\left(\frac{\partial}{\partial x}\left[\frac{\partial V}{\partial r}\right]\cdot\frac{\partial r}{\partial x} + \frac{\partial V}{\partial r}\cdot\frac{\partial}{\partial x}\left[\frac{\partial r}{\partial x}\right]\right) + \left(\frac{\partial}{\partial x}\left[\frac{\partial V}{\partial \lambda}\right]\cdot\frac{\partial \lambda}{\partial x} + \frac{\partial V}{\partial \lambda}\cdot\frac{\partial}{\partial x}\left[\frac{\partial \lambda}{\partial x}\right]\right)\\ &+ \left(\frac{\partial}{\partial x}\left[\frac{\partial V}{\partial \psi}\right]\cdot\frac{\partial \psi}{\partial x} + \frac{\partial V}{\partial \psi}\cdot\frac{\partial}{\partial x}\left[\frac{\partial \psi}{\partial x}\right]\right)\end{split} \tag{D.2}$$

$$\begin{split}\frac{\partial^2 V}{\partial q_x^2} = &\left(\frac{\partial}{\partial x}\left[\frac{\partial V}{\partial r}\right]\cdot\frac{\partial r}{\partial x} + \frac{\partial V}{\partial r}\frac{\partial^2 r}{\partial x^2}\right) + \left(\frac{\partial}{\partial x}\left[\frac{\partial V}{\partial \lambda}\right]\cdot\frac{\partial \lambda}{\partial x} + \frac{\partial V}{\partial \lambda}\frac{\partial^2 \lambda}{\partial x^2}\right)\\ &+ \left(\frac{\partial}{\partial x}\left[\frac{\partial V}{\partial \psi}\right]\cdot\frac{\partial \psi}{\partial x} + \frac{\partial V}{\partial \psi}\frac{\partial^2 \psi}{\partial x^2}\right)\end{split} \tag{D.3}$$

$$\begin{aligned}\frac{\partial^2 V}{\partial q_x^2} = {} & \left(\frac{\partial}{\partial r}\left[\frac{\partial V}{\partial r}\right]\frac{\partial r}{\partial x} + \frac{\partial}{\partial \lambda}\left[\frac{\partial V}{\partial r}\right]\frac{\partial \lambda}{\partial x} + \frac{\partial}{\partial \psi}\left[\frac{\partial V}{\partial r}\right]\frac{\partial \psi}{\partial x}\right)\frac{\partial r}{\partial x} \\ & + \left(\frac{\partial}{\partial r}\left[\frac{\partial V}{\partial \lambda}\right]\frac{\partial r}{\partial x} + \frac{\partial}{\partial \lambda}\left[\frac{\partial V}{\partial \lambda}\right]\frac{\partial \lambda}{\partial x} + \frac{\partial}{\partial \psi}\left[\frac{\partial V}{\partial \lambda}\right]\frac{\partial \psi}{\partial x}\right)\frac{\partial \lambda}{\partial x} \\ & + \left(\frac{\partial}{\partial r}\left[\frac{\partial V}{\partial \psi}\right]\frac{\partial r}{\partial x} + \frac{\partial}{\partial \lambda}\left[\frac{\partial V}{\partial \psi}\right]\frac{\partial \lambda}{\partial x} + \frac{\partial}{\partial \psi}\left[\frac{\partial V}{\partial \psi}\right]\frac{\partial \psi}{\partial x}\right)\frac{\partial \psi}{\partial x} + \frac{\partial V}{\partial r}\frac{\partial^2 r}{\partial x^2} + \frac{\partial V}{\partial \lambda}\frac{\partial^2 \lambda}{\partial x^2} \\ & + \frac{\partial V}{\partial \psi}\frac{\partial^2 \psi}{\partial x^2}\end{aligned} \tag{D.4}$$

$$\begin{aligned}\frac{\partial^2 V}{\partial q_x^2} = {} & \left(\frac{\partial^2 V}{\partial r^2}\frac{\partial r}{\partial x} + \frac{\partial^2 V}{\partial \lambda \partial r}\frac{\partial \lambda}{\partial x} + \frac{\partial^2 V}{\partial \psi \partial r}\frac{\partial \psi}{\partial x}\right)\frac{\partial r}{\partial x} + \left(\frac{\partial^2 V}{\partial r \partial \lambda}\frac{\partial r}{\partial x} + \frac{\partial^2 V}{\partial \lambda^2}\frac{\partial \lambda}{\partial x} + \frac{\partial^2 V}{\partial \psi \partial \lambda}\frac{\partial \psi}{\partial x}\right)\frac{\partial \lambda}{\partial x} \\ & + \left(\frac{\partial^2 V}{\partial r \partial \psi}\frac{\partial r}{\partial x} + \frac{\partial^2 V}{\partial \lambda \partial \psi}\frac{\partial \lambda}{\partial x} + \frac{\partial^2 V}{\partial \psi^2}\frac{\partial \psi}{\partial x}\right)\frac{\partial \psi}{\partial x} + \frac{\partial V}{\partial r}\frac{\partial^2 r}{\partial x^2} + \frac{\partial V}{\partial \lambda}\frac{\partial^2 \lambda}{\partial x^2} + \frac{\partial V}{\partial \psi}\frac{\partial^2 \psi}{\partial x^2}\end{aligned} \tag{D.5}$$

$$\begin{aligned}\frac{\partial^2 V}{\partial q_x^2} = {} & \frac{\partial^2 V}{\partial r^2}\left(\frac{\partial r}{\partial x}\right)^2 + \frac{\partial^2 V}{\partial \lambda \partial r}\frac{\partial \lambda}{\partial x}\frac{\partial r}{\partial x} + \frac{\partial^2 V}{\partial \psi \partial r}\frac{\partial \psi}{\partial x}\frac{\partial r}{\partial x} + \frac{\partial^2 V}{\partial r \partial \lambda}\frac{\partial r}{\partial x}\frac{\partial \lambda}{\partial x} + \frac{\partial^2 V}{\partial \lambda^2}\left(\frac{\partial \lambda}{\partial x}\right)^2 \\ & + \frac{\partial^2 V}{\partial \psi \partial \lambda}\frac{\partial \psi}{\partial x}\frac{\partial \lambda}{\partial x} + \frac{\partial^2 V}{\partial r \partial \psi}\frac{\partial r}{\partial x}\frac{\partial \psi}{\partial x} + \frac{\partial^2 V}{\partial \lambda \partial \psi}\frac{\partial \lambda}{\partial x}\frac{\partial \psi}{\partial x} + \frac{\partial^2 V}{\partial \psi^2}\left(\frac{\partial \psi}{\partial x}\right)^2 + \frac{\partial V}{\partial r}\frac{\partial^2 r}{\partial x^2} \\ & + \frac{\partial V}{\partial \lambda}\frac{\partial^2 \lambda}{\partial x^2} + \frac{\partial V}{\partial \psi}\frac{\partial^2 \psi}{\partial x^2}\end{aligned} \tag{D.6}$$

$$\begin{aligned}\frac{\partial^2 V}{\partial q_x^2} = {} & \frac{\partial^2 V}{\partial r^2}\left(\frac{\partial r}{\partial x}\right)^2 + \frac{\partial^2 V}{\partial \lambda^2}\left(\frac{\partial \lambda}{\partial x}\right)^2 + \frac{\partial^2 V}{\partial \psi^2}\left(\frac{\partial \psi}{\partial x}\right)^2 + 2\,\frac{\partial^2 V}{\partial \lambda \partial r}\frac{\partial \lambda}{\partial x}\frac{\partial r}{\partial x} + 2\,\frac{\partial^2 V}{\partial \psi \partial r}\frac{\partial \psi}{\partial x}\frac{\partial r}{\partial x} \\ & + 2\,\frac{\partial^2 V}{\partial \psi \partial \lambda}\frac{\partial \psi}{\partial x}\frac{\partial \lambda}{\partial x} + \frac{\partial V}{\partial r}\frac{\partial^2 r}{\partial x^2} + \frac{\partial V}{\partial \lambda}\frac{\partial^2 \lambda}{\partial x^2} + \frac{\partial V}{\partial \psi}\frac{\partial^2 \psi}{\partial x^2}\end{aligned} \tag{D.7}$$

$\partial^2 V/\partial q_y^2$ and $\partial^2 V/\partial q_z^2$ have the same form. Just replace x with y and z, respectively.

D.2 Off-diagonal Elements

$$\frac{\partial^2 V}{\partial q_x \partial q_y} = \frac{\partial}{\partial x}\left[\frac{\partial V}{\partial r}\cdot\frac{\partial r}{\partial y}\right] + \frac{\partial}{\partial x}\left[\frac{\partial V}{\partial \lambda}\cdot\frac{\partial \lambda}{\partial y}\right] + \frac{\partial}{\partial x}\left[\frac{\partial V}{\partial \psi}\cdot\frac{\partial \psi}{\partial y}\right] \tag{D.8}$$

$$\begin{aligned}\frac{\partial^2 V}{\partial q_x \partial q_y} = {} & \left(\frac{\partial}{\partial x}\left[\frac{\partial V}{\partial r}\right] \cdot \frac{\partial r}{\partial y} + \frac{\partial V}{\partial r} \cdot \frac{\partial}{\partial x}\left[\frac{\partial r}{\partial y}\right]\right) + \left(\frac{\partial}{\partial x}\left[\frac{\partial V}{\partial \lambda}\right] \cdot \frac{\partial \lambda}{\partial y} + \frac{\partial V}{\partial \lambda} \cdot \frac{\partial}{\partial x}\left[\frac{\partial \lambda}{\partial y}\right]\right) \\ & + \left(\frac{\partial}{\partial x}\left[\frac{\partial V}{\partial \psi}\right] \cdot \frac{\partial \psi}{\partial y} + \frac{\partial V}{\partial \psi} \cdot \frac{\partial}{\partial x}\left[\frac{\partial \psi}{\partial y}\right]\right)\end{aligned} \tag{D.9}$$

$$\begin{aligned}\frac{\partial^2 V}{\partial q_x \partial q_y} = {} & \left(\frac{\partial}{\partial x}\left[\frac{\partial V}{\partial r}\right] \cdot \frac{\partial r}{\partial y} + \frac{\partial V}{\partial r}\frac{\partial^2 r}{\partial x \partial y}\right) + \left(\frac{\partial}{\partial x}\left[\frac{\partial V}{\partial \lambda}\right] \cdot \frac{\partial \lambda}{\partial y} + \frac{\partial V}{\partial \lambda}\frac{\partial^2 \lambda}{\partial x \partial y}\right) \\ & + \left(\frac{\partial}{\partial x}\left[\frac{\partial V}{\partial \psi}\right] \cdot \frac{\partial \psi}{\partial y} + \frac{\partial V}{\partial \psi}\frac{\partial^2 \psi}{\partial x \partial y}\right)\end{aligned} \tag{D.10}$$

$$\begin{aligned}\frac{\partial^2 V}{\partial q_x \partial q_y} = {} & \left(\frac{\partial}{\partial r}\left[\frac{\partial V}{\partial r}\right]\frac{\partial r}{\partial x} + \frac{\partial}{\partial \lambda}\left[\frac{\partial V}{\partial r}\right]\frac{\partial \lambda}{\partial x} + \frac{\partial}{\partial \psi}\left[\frac{\partial V}{\partial r}\right]\frac{\partial \psi}{\partial x}\right)\frac{\partial r}{\partial y} \\ & + \left(\frac{\partial}{\partial r}\left[\frac{\partial V}{\partial \lambda}\right]\frac{\partial r}{\partial x} + \frac{\partial}{\partial \lambda}\left[\frac{\partial V}{\partial \lambda}\right]\frac{\partial \lambda}{\partial x} + \frac{\partial}{\partial \psi}\left[\frac{\partial V}{\partial \lambda}\right]\frac{\partial \psi}{\partial x}\right)\frac{\partial \lambda}{\partial y} \\ & + \left(\frac{\partial}{\partial r}\left[\frac{\partial V}{\partial \psi}\right]\frac{\partial r}{\partial x} + \frac{\partial}{\partial \lambda}\left[\frac{\partial V}{\partial \psi}\right]\frac{\partial \lambda}{\partial x} + \frac{\partial}{\partial \psi}\left[\frac{\partial V}{\partial \psi}\right]\frac{\partial \psi}{\partial x}\right)\frac{\partial \psi}{\partial y} + \frac{\partial V}{\partial r}\frac{\partial^2 r}{\partial x \partial y} \\ & + \frac{\partial V}{\partial \lambda}\frac{\partial^2 \lambda}{\partial x \partial y} + \frac{\partial V}{\partial \psi}\frac{\partial^2 \psi}{\partial x \partial y}\end{aligned} \tag{D.11}$$

$$\begin{aligned}\frac{\partial^2 V}{\partial q_x^2} = {} & \left(\frac{\partial^2 V}{\partial r^2}\frac{\partial r}{\partial x} + \frac{\partial^2 V}{\partial \lambda \partial r}\frac{\partial \lambda}{\partial x} + \frac{\partial^2 V}{\partial \psi \partial r}\frac{\partial \psi}{\partial x}\right)\frac{\partial r}{\partial y} + \left(\frac{\partial^2 V}{\partial r \partial \lambda}\frac{\partial r}{\partial x} + \frac{\partial^2 V}{\partial \lambda^2}\frac{\partial \lambda}{\partial x} + \frac{\partial^2 V}{\partial \psi \partial \lambda}\frac{\partial \psi}{\partial x}\right)\frac{\partial \lambda}{\partial y} \\ & + \left(\frac{\partial^2 V}{\partial r \partial \psi}\frac{\partial r}{\partial x} + \frac{\partial^2 V}{\partial \lambda \partial \psi}\frac{\partial \lambda}{\partial x} + \frac{\partial^2 V}{\partial \psi^2}\frac{\partial \psi}{\partial x}\right)\frac{\partial \psi}{\partial y} + \frac{\partial V}{\partial r}\frac{\partial^2 r}{\partial x \partial y} + \frac{\partial V}{\partial \lambda}\frac{\partial^2 \lambda}{\partial x \partial y} \\ & + \frac{\partial V}{\partial \psi}\frac{\partial^2 \psi}{\partial x \partial y}\end{aligned} \tag{D.12}$$

$$\begin{aligned}\frac{\partial^2 V}{\partial q_x \partial q_y} = {} & \frac{\partial^2 V}{\partial r^2}\frac{\partial r}{\partial x}\frac{\partial r}{\partial y} + \frac{\partial^2 V}{\partial \lambda \partial r}\frac{\partial \lambda}{\partial x}\frac{\partial r}{\partial y} + \frac{\partial^2 V}{\partial \psi \partial r}\frac{\partial \psi}{\partial x}\frac{\partial r}{\partial y} + \frac{\partial^2 V}{\partial r \partial \lambda}\frac{\partial r}{\partial x}\frac{\partial \lambda}{\partial y} + \frac{\partial^2 V}{\partial \lambda^2}\frac{\partial \lambda}{\partial x}\frac{\partial \lambda}{\partial y} \\ & + \frac{\partial^2 V}{\partial \psi \partial \lambda}\frac{\partial \psi}{\partial x}\frac{\partial \lambda}{\partial y} + \frac{\partial^2 V}{\partial r \partial \psi}\frac{\partial r}{\partial x}\frac{\partial \psi}{\partial y} + \frac{\partial^2 V}{\partial \lambda \partial \psi}\frac{\partial \lambda}{\partial x}\frac{\partial \psi}{\partial y} + \frac{\partial^2 V}{\partial \psi^2}\frac{\partial \psi}{\partial x}\frac{\partial \psi}{\partial y} + \frac{\partial V}{\partial r}\frac{\partial^2 r}{\partial x \partial y} \\ & + \frac{\partial V}{\partial \lambda}\frac{\partial^2 \lambda}{\partial x \partial y} + \frac{\partial V}{\partial \psi}\frac{\partial^2 \psi}{\partial x \partial y}\end{aligned} \tag{D.13}$$

By similarity

$$\frac{\partial^2 V}{\partial q_y \partial q_x} = \frac{\partial^2 V}{\partial q_x \partial q_y} \tag{D.14}$$

$\partial^2 V/\partial q_x \partial q_z$ has the same form. Just replace y with z in Equation (D.13). The same is true for $\partial^2 V/\partial q_y \partial q_z$ by replacing y with z and x with y.

Appendix E

Source Code

E.1 Proteus Front End

```
%% CONSTANTS
global MU mu DU du TU J2 omegae Omegae angumb angpen geo_degree geo_order ...
    deltaJD dtheta use_skew2nu search_argp
MU = 398600.4415; % [km^3/sec^2]
DU = 6378.1363; % Earth radius [km]
TU = 13.44686457; % [min]
J2 = 0.0010826269;
mu = 1.0;
du = 1.0;
Omegae = 7.292115e-5; % Earth angular velocity [rad/sec] f/ WGS-84
omegae = Omegae*60*TU; % Earth angular velocity [rad/TU]
rs = 696000.0; % Sun radius
AU = 149597870.0;
angumb = atan((rs-DU)/AU);
angpen = atan((rs+DU)/AU);
% Julian Date correction (difference between ECEF2ECI and GMAT ECI values)
deltaJD = - 0.0000025444;
dtheta = -0.00000004005; % [deg]

%% INPUTS
% Specify geopotential fidelity
geo_degree = 20; geo_order = 20;
% Argument of perigee search override
search_argp = 1; % 1 = Search, 0 = Use Surveyor
% Specify if using torus
use_torus = 0;
% Specify case file for Surveyor and Target orbits
input.runCase = 'case001';

%% LOAD CASE FILE
codeDir = pwd;
mkdir('Cases',sprintf('%s',input.runCase))
caseFile = strcat(codeDir,'\Cases\',sprintf('%s.m',input.runCase));
run(caseFile)

% If truth is from a TLE, use Vallado's code to convert to COEs
```

```
if strcmp(InputForm,'tle')==1
    ateme = [0.001;0.002;0.003];
    typerun = 'v'; % Verification mode, see twoline2rv
    typeinput = 'e'; % Epoch, see twoline2rv
    whichconst = 84; % WGS-84
    [tumin,mu,radiusearthkm,xke,j2,j3,j4,j3oj2] = getgravc(whichconst);

    longstr1 = TLE1;
    % Last three are startmfe, stopmfe, deltamin:
    longstr2 = [TLE2,'     0.00          0.0           0.00  '];

    % Convert the char string to sgp4 elements
    [satrec,startmfe,stopmfe,deltamin] = twoline2rv(whichconst,longstr1, ...
        longstr2, typerun, typeinput);
    % Call the propagator to get the initial state vector value
    tsince = stopmfe;
    [satrec,rteme,vteme] = sgp4(satrec,tsince);

    % Transform teme to ecef vectors
    jdutc = satrec.jdsatepoch + tsince/1440.0;
    [year,mon,day,hr,minute,sec] = invjday(jdutc);

    % Assume tle epoch is utc and dut1, xp, etc are zero
    dut1 = 0.0;
    dat  = 32;
    xp   =  0.0;  % "
    yp   =  0.0;
    lod  =  0.0;
    ddpsi = 0.0;  % "
    ddeps = 0.0;

    % If have values for the date in question
    % dut1 = 0.2048315;
    % dat  = 32;
    % xp   =  0.109600;  % "
    % yp   =  0.284144;
    % lod  =  0.0004116;
    % ddpsi = -0.054055;  % "
    % ddeps = -0.006183;

    timezone = 0;
    order = 106; % Number of terms for nutation, see truemean
    eqeterms = 2; % Use the extra eqeq terms in j2000, see teme2eci
    opt = 'a'; % Specify the iau00 approach, see teme2eci

    [ut1,tut1,jdut1,utc,tai,tt,ttt,jdtt,tdb,ttdb,jdtdb] = convtime(year, ...
        mon,day,hr,minute,sec,timezone,dut1,dat);

    [recef,vecef,aecef] = teme2ecef(rteme',vteme',ateme,ttt,jdut1,lod,xp,yp);

    [reci,veci,aeci] = teme2eci  (rteme',vteme',ateme,ttt,order,eqeterms,opt);

    [p,a,ecc,incl,omega,argp,nu,m,arglat,truelon,lonper] = rv2coe(reci,veci);

    % Surveyor Orbital Elements
    input.coes(1,1) = a; % semi-major axis [km]
    input.coes(1,2) = ecc; % eccentricity
    input.coes(1,3) = incl*180/pi; % inclination [deg]
    input.coes(1,4) = omega*180/pi; % RAAN [deg]
    input.coes(1,5) = argp*180/pi; % argument of perigee [deg]
    input.coes(1,6) = 0; %nu*180/pi; % true anomaly [deg]

    % Target Orbital Elements
    input.coes(2,1) = input.coes(1,1) + dsma_target; % semi-major axis [km]
    input.coes(2,2) = input.coes(1,2) + decc_target; % eccentricity
    input.coes(2,3) = input.coes(1,3) + dinc_target; % inclination [deg]
    input.coes(2,4) = input.coes(1,4) + dRAAN_target; % RAAN [deg]
    input.coes(2,5) = input.coes(1,5) + dAOP; % argument of perigee [deg]
    input.coes(2,6) = input.coes(1,6) + dnu; % true anomaly [deg]

    input.epoch = datestr(datenum([year,mon,day,hr,minute,sec]), ...
        'dd mmm yyyy HH:MM:SS.FFF'); % UTC

    input.state_type = 'Keplerian';
    input.coord_sys = 'EarthMJ2000Eq';
    input.elements = input.coes;
end

%% Propagate satellites for one orbital period
options = [0,dt,0,1,0]; % [prop_time,dt,torus,output,orbitview]
SatDat = gmatOrbitProp(input.state_type,input.coord_sys,input.coes,...
    input.epoch,'UTCGregorian',options);

%% Generate LOS observations from truth orbits
[Surveyor,Target] = losProc(SatDat,input,[bias,stdev,1]);

%% Yates' IOD technique
[hyp,mapping,eclipse,d,d_truth,delta_d,alpha_disp] = losIOD(input,SatDat,...
    Surveyor,Target,dt,bias,stdev);
iod.elements = mapping.coes;
iod.state_type = 'Keplerian';
iod.coord_sys = input.coord_sys;
iod.epoch = input.epoch;

%% Propagate IOD solution
options = [0,dt,0,1,0]; % [prop_time,dt,torus,output,orbitview]
SatDat_iod = gmatOrbitProp('Keplerian',iod.coord_sys,[input.coes(1,:);...
    iod.elements],iod.epoch,'UTCGregorian',options);

%% Generate LOS observations from IOD solution
[Surveyor,Target_iod] = losProc(SatDat_iod,input,[bias,stdev,1]);

%% Assess range error from IOD
range_error_iod = (Target_iod.rho_true(:,1)-Target.rho_true(:,1));
range_error_percent_iod = (Target_iod.rho_true(:,1)-
Target.rho_true(:,1))./Target.rho_true(:,1)*100;
figure(22)
hold on
plot(SatDat_iod.time(:,1)/60,range_error_iod,'-r','LineWidth',3)
plot(SatDat_iod.time(:,1)/60,range_error_percent_iod,'-b','LineWidth',3)
grid on; grid minor;
box on
```

```
xlabel('\textbf{Time (min)}','Interpreter','latex','FontSize',33)
ylabel('\textbf{Range Error}','Interpreter','latex','FontSize',33)
h = legend('\textbf{Absolute Range Error (km)}', ...
    '\textbf{Percent Range Error (\%)}','location','NorthEast');
set(h,'Interpreter','latex','FontSize',32);
set(gca,'FontSize',32,'FontWeight','bold')
axis normal;
saveas(gcf,strcat(codeDir,'\Cases\',sprintf('%s',input.runCase),'\Fig22.fig'))

%% Improve upon the IOD solution with NLS
if geo_degree > 0 && use_torus == 1
    % Propagate satellite for one year (6 mos forward & 6 mos backward)
    options = [16070400,60,1,0,0]; % [prop_time,dt,torus,output,orbitview]
    SatDat_torus = gmatOrbitProp('Keplerian',iod.coord_sys,...
        iod.elements,iod.epoch,'UTCGregorian',options);
    % Spectral decomposition of IOD orbit
    spec_decomp; % Copyright Max Yates
end
% Nonlinear Least Squares
method = 'direct'; % '2body', 'numerical', or 'direct'
NLSQ;
```

E.2 Example Case File Forms (COE version & TLE version)

```
%% case001.m
InputForm = 'coes';

% Surveyor Orbital Elements
input.coes(1,1) = 6777.329; % semi-major axis [km]
input.coes(1,2) = 0.0012606; % eccentricity
input.coes(1,3) = 51.98716; % inclination [deg]
input.coes(1,4) = 75.49392; % RAAN [deg]
input.coes(1,5) = 102.64716; % argument of perigee [deg]
input.coes(1,6) = 0; % true anomaly [deg]

% Target Orbital Elements
input.coes(2,1) = 6777.329; % semi-major axis [km]
input.coes(2,2) = 0.005; % eccentricity
input.coes(2,3) = 51.98716; % inclination [deg]
input.coes(2,4) = 75.49392; % RAAN [deg]
input.coes(2,5) = 102.64716; % argument of perigee [deg]
input.coes(2,6) = 0; % true anomaly [deg]

input.epoch = '14 Jun 2016 12:00:00.000';

input.state_type = 'Keplerian';
input.coord_sys = 'EarthMJ2000Eq';
input.elements = input.coes;

% Sensor parameters
bias = 0; % [arc sec]
stdev = 0; % [arc sec]
dt = 0.1; % sample rate [sec]


%% case003.m
InputForm = 'tle';

% Surveyor TLE - COSMOS 2513
% https://www.celestrak.com/NORAD/elements/geo.txt [20170119]
TLE1='1 41121U 15075A   17017.57363789 -.00000148  00000-0  00000+0 0  9994';
TLE2='2 41121   0.0814  90.4206 0003078 212.8794 100.5875  1.00270486  4138';

% Target Orbital Element Offsets
dsma_target = 0; % semi-major axis [km]
decc_target = 0.00075; % eccentricity
dinc_target = 0.1; % inclination [deg]
dRAAN_target = -0.03; % RAAN [deg]
dAOP = 0; % argument of perigee [deg]
dnu = 0.06; % true anomaly [deg]

% sensor parameters
bias = 0; % [arc sec]
stdev = 10; % [arc sec]
dt = 1; % sample rate [sec]
```

E.3 GMAT Interface for Orbit Propagation

```
%  [function]   gmatOrbitProp
%
%  [purpose]:   Function calls the GMAT server to propoagate orbits.
%               Requires GMAT 2011A with MATLAB server capability.
%
%  [useage]:    [SatDat] = gmatOrbitProp(state_type,coord_sys,elements,...
%                             epoch,epoch_format,OPTIONS)
%
%  [inputs]:    state_type = use GMAT state string: Keplerian or Cartesian
%
%               coord_sys = Coordinate system; use same string as GMAT:
%                           EarthMJ2000Eq, EarthFixed, etc.
%
%               elements  = Matrix (nx6) containing initial conditions
%                           for the ith of n satellites being propagated.
%                           State type dictates format. If Cartesian, use
%                           position and velocity vector. If Keplerian, use
%                           COEs, i.e. elements(i,1:6)
%                           where
%                              elements(i,1) = semi-major axis [km]
%                              elements(i,2) = eccentricity
%                              elements(i,3) = inclination [deg]
%                              elements(i,4) = RAAN [deg]
%                              elements(i,5) = argument of perigee [deg]
%                              elements(i,6) = true anomaly [deg]
%
%               epoch     = Time associated with initial conditions
%
%            epoch_format = Date format from GMAT such as 'UTCGregorian'
%
%               OPTIONS   = [prop_time,dt,torus,output,orbitview] where:
%
%               prop_time = Duration of propagation [sec]. Use 0 if the
%                           ith satellite's Keplerian period is desired.
%                           Use 16070400 for a 1-year torus (6-mos forward
%                           and backward).
%
%               dt        = Sampling frequency for returned data [sec]. Use
%                           0 for data increments based on a period ratio
%                           of floor(period/5400) where 5400 sec amounts to
%                           ~1 sec interval at LEO.
%
%               torus     = If set to 1, perform forward & backward
%                           propagation and construct Fourier series torus
%                           files, else 0 enables normal forward
%                           propagation
%
%               output    = If set to 0, limited outputs for hypotheses.
%
%               orbitview = If set to 1, GMAT will display my custom
%                           six-view window to inspect the relative orbit.
%
%  [outputs]:   SatDat    = Structure containing the GMAT extracted data.
%                            The exact content is reconfigurable in the
%                            code, but uses the form SatDat.time(:,i)
%
%  [refs]:
%
%  [history]:   9 Dec 2016 - Max Yates

function [SatDat] = gmatOrbitProp(state_type,coord_sys,elements,epoch,...
    epoch_format,OPTIONS)

global J2 MU DU angumb angpen geo_order geo_degree

prop_time = OPTIONS(1);
torus = OPTIONS(3);
output = OPTIONS(4);
orbitview = OPTIONS(5);

if geo_degree == 0
    % calculate Keplerian period of Sat1
    if strcmp(state_type,'Keplerian') == 1
        period = 2*pi*sqrt(elements(1,1)^3/MU); % [sec]
    elseif strcmp(state_type,'Cartesian') == 1
        [p,a,ecc,incl,omega,argp,nu,m,arglat,truelon,lonper ] = rv2coe...
            (elements(1,1:3),elements(1,4:6));
        period = 2*pi*sqrt(a^3/MU); % [sec]
    end
elseif geo_degree > 0
    % calculate Kozai period of Sat1
    if strcmp(state_type,'Keplerian') == 1
        Mdot0=sqrt(MU/(elements(1,1)^3))*(-((3*J2*(DU^2)/(2*...
            (elements(1,1)^2)*(1-elements(1,2)^2)^1.5))*(3/2*...
            (sin(elements(1,3)*pi/180))^2-1)));
        n = sqrt(MU/elements(1,1)^3); % [rad/sec] mean motion
        period = 2*pi/(n+Mdot0); % [sec]
    elseif strcmp(state_type,'Cartesian') == 1
        [p,a,ecc,incl,omega,argp,nu,m,arglat,truelon,lonper ] = rv2coe...
            (elements(1,1:3),elements(1,4:6));
        Mdot0=sqrt(MU/(a^3))*(-((3*J2*(DU^2)/(2*(a^2)*(1-ecc^2)^1.5))*...
            (3/2*(sin(incl))^2-1)));
        n = sqrt(MU/a^3); % [rad/sec] mean motion
        period = 2*pi/(n+Mdot0); % [sec]
    end
end

% sampling frequency for returned data [sec]
if torus==0
    if OPTIONS(2)==0
        % 5400 selected for ~1 sec interval on 90 min period
        dt = floor(period/5400);
    else
        dt = OPTIONS(2);
    end
elseif torus==1
    dt = 60;
end
```

```
n = size(elements,1); % number of satellites

% NOTE: Must manually start MATLAB server from within GMAT before executing
OpenGMAT % Open connection to GMAT for the sending of commands

for j = 1:(torus+1) % torus loop counts the forward & backward propagation

    ClearGMAT % Clear all objects currently set in GMAT

    %% User-Modified Default Celestial Bodies
    eval('GMAT Earth.NutationUpdateInterval = 26070400;')

    %% Spacecraft
    for i = 1:n % satellite loop
        Sat = ['Sat',num2str(i)];
        eval(strcat('Create Spacecraft',sprintf(' %s;',Sat)));
        if strcmp(epoch_format,'UTCGregorian') == 1
            % UTCG Format: '15 Dec 2017 14:55:35.017'
            eval(strcat('GMAT',sprintf(' %s',Sat),...
                '.DateFormat = UTCGregorian;'));
            eval(strcat('GMAT',sprintf(' %s',Sat),'.Epoch =',...
                sprintf(' ''%s'';',epoch)));
        elseif strcmp(epoch_format,'UTCModJulian') == 1
            eval(strcat('GMAT',sprintf(' %s',Sat),...
                '.DateFormat = UTCModJulian;'));
            eval(strcat('GMAT',sprintf(' %s',Sat),'.Epoch =',...
                sprintf(' %.16f;',epoch)));
        end
        eval(strcat('GMAT',sprintf(' %s',Sat),'.CoordinateSystem =',...
            sprintf(' %s;',coord_sys)));
        eval(strcat('GMAT',sprintf(' %s',Sat),'.DisplayStateType =',...
            sprintf(' %s;',state_type)));
        if strcmp(state_type,'Keplerian') == 1
            eval(strcat('GMAT',sprintf(' %s',Sat),'.SMA =',...
                sprintf(' %.16f;',elements(i,1)))); % [km]
            eval(strcat('GMAT',sprintf(' %s',Sat),'.ECC =',...
                sprintf(' %.16f;',elements(i,2)))); % [nd]
            eval(strcat('GMAT',sprintf(' %s',Sat),'.INC =',...
                sprintf(' %.16f;',elements(i,3)))); % [deg]
            eval(strcat('GMAT',sprintf(' %s',Sat),'.RAAN =',...
                sprintf(' %.16f;',elements(i,4)))); % [deg]
            eval(strcat('GMAT',sprintf(' %s',Sat),'.AOP =',...
                sprintf(' %.16f;',elements(i,5)))); % [deg]
            eval(strcat('GMAT',sprintf(' %s',Sat),'.TA =',...
                sprintf(' %.16f;',elements(i,6)))); % [deg]
        elseif strcmp(state_type,'Cartesian') == 1
            eval(strcat('GMAT',sprintf(' %s',Sat),'.X =',...
                sprintf(' %.16f;',elements(i,1)))); % [km]
            eval(strcat('GMAT',sprintf(' %s',Sat),'.Y =',...
                sprintf(' %.16f;',elements(i,2)))); % [nd]
            eval(strcat('GMAT',sprintf(' %s',Sat),'.Z =',...
                sprintf(' %.16f;',elements(i,3)))); % [deg]
            eval(strcat('GMAT',sprintf(' %s',Sat),'.VX =',...
                sprintf(' %.16f;',elements(i,4)))); % [deg]
            eval(strcat('GMAT',sprintf(' %s',Sat),'.VY =',...
                sprintf(' %.16f;',elements(i,5)))); % [deg]
            eval(strcat('GMAT',sprintf(' %s',Sat),'.VZ =',...
                sprintf(' %.16f;',elements(i,6)))); % [deg]
        end
        eval(strcat('GMAT',sprintf(' %s',Sat),'.DryMass = 14;')); % [kg]
        eval(strcat('GMAT',sprintf(' %s',Sat),'.Cd = 0;')); % [nd]
        eval(strcat('GMAT',sprintf(' %s',Sat),'.Cr = 0;')); % [nd]
        eval(strcat('GMAT',sprintf(' %s',Sat),'.DragArea = 0;')); % [m^2]
        eval(strcat('GMAT',sprintf(' %s',Sat),'.SRPArea = 0;')); % [m^2]
        % SPICE
        eval(strcat('GMAT',sprintf(' %s',Sat),'.NAIFId = -123456789;'));
        eval(strcat('GMAT',sprintf(' %s',Sat),...
            '.NAIFIdReferenceFrame = -123456789;'));
        eval(strcat('GMAT',sprintf(' %s',Sat),'.Id = ''SatId'';'));
        eval(strcat('GMAT',sprintf(' %s',Sat),...
            '.Attitude = CoordinateSystemFixed;'));
        eval(strcat('GMAT',sprintf(' %s',Sat),...
            '.ModelFile = ''../data/vehicle/models/aura.3ds'';'));
        eval(strcat('GMAT',sprintf(' %s',Sat),'.ModelOffsetX = 0;'));
        eval(strcat('GMAT',sprintf(' %s',Sat),'.ModelOffsetY = 0;'));
        eval(strcat('GMAT',sprintf(' %s',Sat),'.ModelOffsetZ = 0;'));
        eval(strcat('GMAT',sprintf(' %s',Sat),'.ModelRotationX = 0;'));
        eval(strcat('GMAT',sprintf(' %s',Sat),'.ModelRotationY = 0;'));
        eval(strcat('GMAT',sprintf(' %s',Sat),'.ModelRotationZ = 0;'));
        eval(strcat('GMAT',sprintf(' %s',Sat),'.ModelScale = 0.001;'));
        eval(strcat('GMAT',sprintf(' %s',Sat),...
            '.AttitudeDisplayStateType = ''Quaternion'';'));
        eval(strcat('GMAT',sprintf(' %s',Sat),...
            '.AttitudeRateDisplayStateType = ''AngularVelocity'';'));
        eval(strcat('GMAT',sprintf(' %s',Sat),...
            '.AttitudeCoordinateSystem = EarthMJ2000Eq;'));
        eval(strcat('GMAT',sprintf(' %s',Sat),...
            '.EulerAngleSequence = ''321'';'));
    end

    %% ForceModels
    eval('Create ForceModel LowEarthProp_ForceModel;');
    eval('GMAT LowEarthProp_ForceModel.CentralBody = Earth;');
    eval('GMAT LowEarthProp_ForceModel.PrimaryBodies = {Earth};');
    eval('GMAT LowEarthProp_ForceModel.Drag = None;');
    eval('GMAT LowEarthProp_ForceModel.SRP = Off;');
    eval('GMAT LowEarthProp_ForceModel.ErrorControl = RSSStep;');
    eval(strcat('GMAT LowEarthProp_ForceModel.GravityField.Earth.Degree =',...
        sprintf(' %i;',geo_degree)));
    eval(strcat('GMAT LowEarthProp_ForceModel.GravityField.Earth.Order =',...
        sprintf(' %i;',geo_order)));
    % EGM96.cof or JGM3.cof
    eval('GMAT LowEarthProp_ForceModel.GravityField.Earth.PotentialFile =
''EGM96.cof'';');

    %% Propagators
    eval('Create Propagator LowEarthProp;');
    eval('GMAT LowEarthProp.FM = LowEarthProp_ForceModel;');
    % PrinceDormand78 or RungeKutta89
    eval('GMAT LowEarthProp.Type = PrinceDormand78;');
    eval(strcat('GMAT LowEarthProp.InitialStepSize =',sprintf(' %i;',dt)));
    eval('GMAT LowEarthProp.Accuracy = 9.999999999999999e-012;');
    eval(strcat('GMAT LowEarthProp.MinStep =',sprintf(' %i;',dt)));
    %eval('GMAT LowEarthProp.MinStep = 0.001;');
    eval(strcat('GMAT LowEarthProp.MaxStep =',sprintf(' %i;',dt)));
    eval('GMAT LowEarthProp.MaxStepAttempts = 50;');
    eval('GMAT LowEarthProp.StopIfAccuracyIsViolated = true;');

    %% Coordinate Systems (non-default ones)
    eval('Create CoordinateSystem SunMJ2000Eq;');
    eval('GMAT SunMJ2000Eq.Origin = Sun;');
    eval('GMAT SunMJ2000Eq.Axes = MJ2000Eq;');
    if orbitview==1
        eval('Create CoordinateSystem LVLH;');
        eval('GMAT LVLH.Origin = Sat1;');
        eval('GMAT LVLH.Axes = ObjectReferenced;');
        eval('GMAT LVLH.YAxis = N;');
        %eval('GMAT LVLH.XAxis = V;');
        eval('GMAT LVLH.ZAxis = R;');
        eval('GMAT LVLH.Primary = Earth;');
        eval('GMAT LVLH.Secondary = Sat1;');
    end

    %% Subscribers
    for i = 1:n % satellite loop
        Sat = ['Sat',num2str(i)];
        Report = ['ReportFile',num2str(i)];
        eval(strcat('Create ReportFile',sprintf(' %s;',Report)));
        eval(strcat('GMAT',sprintf(' %s.',Report),...
            'SolverIterations = Current;'));
        if torus==0
            ReportFileLoc = strcat(...
                'C:\GmatVS11a\GmatDevelopment\application\output\',...
                sprintf('%s',Report),'.txt');
            eval(strcat('GMAT',sprintf(' %s.',Report),'Filename =',...
                sprintf(' ''%s'';',ReportFileLoc)));
        elseif torus==1
            if j==1
                eval('GMAT ReportFile1.Filename =
''C:\GmatVS11a\GmatDevelopment\application\output\forward.txt'';');
            elseif j==2
                eval('GMAT ReportFile1.Filename =
''C:\GmatVS11a\GmatDevelopment\application\output\backward.txt'';');
            end
        end

        eval(strcat('GMAT',sprintf(' %s.',Report),'Precision = 16;'));
        EarthFixed = strcat(sprintf('%s.',Sat),'EarthFixed.');
        EarthMJ2000Eq = strcat(sprintf('%s.',Sat),'EarthMJ2000Eq.');
        SunMJ2000Eq = strcat(sprintf('%s.',Sat),'SunMJ2000Eq.');
        Earth = strcat(sprintf('%s.',Sat),'Earth.');
        if torus==1
            eval(strcat('GMAT',sprintf(' %s.',Report),'Add = {',...
                sprintf('%s.',Sat),'ElapsedSecs,',EarthFixed,'X,',...
                EarthFixed,'Y,',EarthFixed,'Z,',EarthFixed,'VX,',...
                EarthFixed,'VY,',EarthFixed,'VZ};'));
        elseif torus==0
            if output==0
                eval(strcat('GMAT',sprintf(' %s.',Report),'Add = {',...
                    sprintf('%s.',Sat),'ElapsedSecs,',EarthMJ2000Eq,...
                    'X,',EarthMJ2000Eq,'Y,',EarthMJ2000Eq,'Z,',...
                    EarthMJ2000Eq,'VX,',EarthMJ2000Eq,'VY,',EarthMJ2000Eq,...
                    'VZ,',Earth,'SMA,',Earth,'TA,',Earth,'RMAG,',...
                    SunMJ2000Eq,'X,',SunMJ2000Eq,'Y,',SunMJ2000Eq,'Z};'));
            else
                eval(strcat('GMAT',sprintf(' %s.',Report),'Add = {',...
                    sprintf('%s.',Sat),'ElapsedSecs,',EarthFixed,'X,',...
                    EarthFixed,'Y,',EarthFixed,'Z,',EarthFixed,'VX,',...
                    EarthFixed,'VY,',EarthFixed,'VZ,',sprintf('%s.',Sat),...
                    'UTCModJulian,',EarthMJ2000Eq,'X,',EarthMJ2000Eq,...
                    'Y,',EarthMJ2000Eq,'Z,',EarthMJ2000Eq,'VX,',...
                    EarthMJ2000Eq,'VY,',EarthMJ2000Eq,'VZ,',Earth,'SMA,',...
                    Earth,'ECC,',EarthMJ2000Eq,'INC,',EarthMJ2000Eq,...
                    'RAAN,',EarthMJ2000Eq,'AOP,',Earth,'TA,',Earth,'MA,',...
                    Earth,'RMAG,',SunMJ2000Eq,'X,',SunMJ2000Eq,'Y,',...
                    SunMJ2000Eq,'Z};'));
            end
        end
        eval(strcat('GMAT',sprintf(' %s.',Report),'WriteHeaders = Off;'));
        eval(strcat('GMAT',sprintf(' %s.',Report),'LeftJustify = On;'));
        eval(strcat('GMAT',sprintf(' %s.',Report),'ZeroFill = Off;'));
        eval(strcat('GMAT',sprintf(' %s.',Report),'ColumnWidth = 24;'));
        eval(strcat('GMAT',sprintf(' %s.',Report),'WriteReport = true;'));
    end

    if orbitview==1
        Sats2Show = 'Sat1';
        True = 'true';
        if n>1
            for i = 2:n % satellite loop
                Sat = ['Sat',num2str(i)];
                Sats2Show = strcat(Sats2Show,sprintf(', %s',Sat));
                True = strcat(True,sprintf(' %s','true'));
            end
        end
        eval('Create OrbitView OrbitView1;');
        eval('GMAT OrbitView1.SolverIterations = Current;');
        eval(strcat('GMAT OrbitView1.Add = {',sprintf('%s, Earth};',...
            Sats2Show)));
        eval('GMAT OrbitView1.CoordinateSystem = LVLH;');
        eval(strcat('GMAT OrbitView1.DrawObject = [',sprintf(...
            ' %s true ];',True)));
        eval('GMAT OrbitView1.DataCollectFrequency = 1;');
        eval('GMAT OrbitView1.UpdatePlotFrequency = 50;');
        eval('GMAT OrbitView1.NumPointsToRedraw = 0;');
        eval('GMAT OrbitView1.ShowPlot = true;');
        eval('GMAT OrbitView1.ViewPointReference = Sat1;');
        eval('GMAT OrbitView1.ViewPointVector = [ 0 0 1 ];');
        eval('GMAT OrbitView1.ViewDirection = Sat1;');
        % change this to zoom in or out
        eval('GMAT OrbitView1.ViewScaleFactor = 200;');
        eval('GMAT OrbitView1.ViewUpCoordinateSystem = LVLH;');
        eval('GMAT OrbitView1.ViewUpAxis = Y;');
        eval('GMAT OrbitView1.XYPlane = Off;');
    eval('GMAT OrbitView1.WireFrame = Off;');
    eval('GMAT OrbitView1.Axes = On;');
    eval('GMAT OrbitView1.Grid = Off;');
    eval('GMAT OrbitView1.SunLine = Off;');
    eval('GMAT OrbitView1.UseInitialView = On;');
    eval('GMAT OrbitView1.StarCount = 7000;');
    eval('GMAT OrbitView1.EnableStars = Off;');
    eval('GMAT OrbitView1.EnableConstellations = Off;');

    eval('Create OrbitView OrbitView2;');
    eval('GMAT OrbitView2.SolverIterations = Current;');
    eval(strcat('GMAT OrbitView2.Add = {',sprintf('%s, Earth};',...
        Sats2Show)));
    eval('GMAT OrbitView2.CoordinateSystem = LVLH;');
    eval(strcat('GMAT OrbitView2.DrawObject = [',sprintf(...
        ' %s true ];',True)));
    eval('GMAT OrbitView2.DataCollectFrequency = 1;');
    eval('GMAT OrbitView2.UpdatePlotFrequency = 50;');
    eval('GMAT OrbitView2.NumPointsToRedraw = 0;');
    eval('GMAT OrbitView2.ShowPlot = true;');
    eval('GMAT OrbitView2.ViewPointReference = Sat1;');
    eval('GMAT OrbitView2.ViewPointVector = [ 0 -1 0 ];');
    eval('GMAT OrbitView2.ViewDirection = Sat1;');
    % change this to zoom in or out
    eval('GMAT OrbitView2.ViewScaleFactor = 200;');
    eval('GMAT OrbitView2.ViewUpCoordinateSystem = LVLH;');
    eval('GMAT OrbitView2.ViewUpAxis = Z;');
    eval('GMAT OrbitView2.XYPlane = Off;');
    eval('GMAT OrbitView2.WireFrame = Off;');
    eval('GMAT OrbitView2.Axes = On;');
    eval('GMAT OrbitView2.Grid = Off;');
    eval('GMAT OrbitView2.SunLine = Off;');
    eval('GMAT OrbitView2.UseInitialView = On;');
    eval('GMAT OrbitView2.StarCount = 7000;');
    eval('GMAT OrbitView2.EnableStars = Off;');
    eval('GMAT OrbitView2.EnableConstellations = Off;');

    eval('Create OrbitView OrbitView3;');
    eval('GMAT OrbitView3.SolverIterations = Current;');
    eval(strcat('GMAT OrbitView3.Add = {',sprintf('%s, Earth};',...
        Sats2Show)));
    eval('GMAT OrbitView3.CoordinateSystem = LVLH;');
    eval(strcat('GMAT OrbitView3.DrawObject = [',sprintf(...
        ' %s true ];',True)));
    eval('GMAT OrbitView3.DataCollectFrequency = 1;');
    eval('GMAT OrbitView3.UpdatePlotFrequency = 50;');
    eval('GMAT OrbitView3.NumPointsToRedraw = 0;');
    eval('GMAT OrbitView3.ShowPlot = true;');
    eval('GMAT OrbitView3.ViewPointReference = Sat1;');
    eval('GMAT OrbitView3.ViewPointVector = [ 0.5 -0.5 0.5 ];');
    eval('GMAT OrbitView3.ViewDirection = Sat1;');
    % change this to zoom in or out
    eval('GMAT OrbitView3.ViewScaleFactor = 200;');
    eval('GMAT OrbitView3.ViewUpCoordinateSystem = LVLH;');
    eval('GMAT OrbitView3.ViewUpAxis = Z;');
    eval('GMAT OrbitView3.XYPlane = Off;');
```

```
        eval('GMAT OrbitView3.WireFrame = Off;');
        eval('GMAT OrbitView3.Axes = On;');
        eval('GMAT OrbitView3.Grid = Off;');
        eval('GMAT OrbitView3.SunLine = Off;');
        eval('GMAT OrbitView3.UseInitialView = On;');
        eval('GMAT OrbitView3.StarCount = 7000;');
        eval('GMAT OrbitView3.EnableStars = Off;');
        eval('GMAT OrbitView3.EnableConstellations = Off;');

        eval('Create OrbitView OrbitView4;');
        eval('GMAT OrbitView4.SolverIterations = Current;');
        eval(strcat('GMAT OrbitView4.Add = {',sprintf('%s, Earth};',...
            Sats2Show)));
        eval('GMAT OrbitView4.CoordinateSystem = LVLH;');
        eval(strcat('GMAT OrbitView4.DrawObject = [',sprintf(...
            ' %s true ];',True)));
        eval('GMAT OrbitView4.DataCollectFrequency = 1;');
        eval('GMAT OrbitView4.UpdatePlotFrequency = 50;');
        eval('GMAT OrbitView4.NumPointsToRedraw = 0;');
        eval('GMAT OrbitView4.ShowPlot = true;');
        eval('GMAT OrbitView4.ViewPointReference = Sat1;');
        eval('GMAT OrbitView4.ViewPointVector = [ 1 0 0 ];');
        eval('GMAT OrbitView4.ViewDirection = Sat1;');
        % change this to zoom in or out
        eval('GMAT OrbitView4.ViewScaleFactor = 200;');
        eval('GMAT OrbitView4.ViewUpCoordinateSystem = LVLH;');
        eval('GMAT OrbitView4.ViewUpAxis = Z;');
        eval('GMAT OrbitView4.XYPlane = Off;');
        eval('GMAT OrbitView4.WireFrame = Off;');
        eval('GMAT OrbitView4.Axes = On;');
        eval('GMAT OrbitView4.Grid = Off;');
        eval('GMAT OrbitView4.SunLine = Off;');
        eval('GMAT OrbitView4.UseInitialView = On;');
        eval('GMAT OrbitView4.StarCount = 7000;');
        eval('GMAT OrbitView4.EnableStars = Off;');
        eval('GMAT OrbitView4.EnableConstellations = Off;');

        Create XYPlot XYPlot1;
        GMAT XYPlot1.SolverIterations = Current;
        GMAT XYPlot1.XVariable = time_hr;
        GMAT XYPlot1.YVariables = {s_rel_12, w_rel_12, r_rel_12, rmag_rel};
        GMAT XYPlot1.ShowGrid = true;
        GMAT XYPlot1.ShowPlot = true;

        %% Arrays, Variables, Strings
        eval('Create Variable i step totalDuration nSteps s_rel_12 r_rel_12
w_rel_12 rmag_rel time_hr Rmag_surveyor;');
        eval('Create Variable R1_surveyor R2_surveyor R3_surveyor
rxv1_surveyor rxv2_surveyor rxv3_surveyor rxv_mag_surveyor W1_surveyor
W2_surveyor W3_surveyor;');
        eval('Create Variable S1_surveyor S2_surveyor S3_surveyor
s_lvlh_surveyor r_lvlh_surveyor w_lvlh_surveyor s_lvlh_rso r_lvlh_rso
w_lvlh_rso;');
    end

    %% Mission Sequence
    eval('BeginMissionSequence;');
    Sats2Prop = 'Sat1';
    if n>1
        for i = 2:n % satellite loop
            Sat = ['Sat',num2str(i)];
            Sats2Prop = strcat(Sats2Prop,sprintf(', %s',Sat));
        end
    end
    % currently propagates all satellites to same satStopTime
    if prop_time == 0
        % Default: 102% of period to catch peaks at the end
        satStopTime = round(period*1.02); % [sec]
    elseif prop_time == -0.02; % No extra 2%
        satStopTime = round(period); % [sec]
    else
        satStopTime = round(prop_time); % [sec]
    end
    if orbitview==1
        eval('GMAT step = 100;');
        eval(strcat('GMAT nSteps =',sprintf(' %i ',satStopTime),'/ step;'));
        eval('For i = 1:1:nSteps;');
        eval(strcat('Propagate LowEarthProp(',sprintf('%s',Sats2Prop),...
            ') {Sat1.ElapsedSecs = step};'));
        eval('GMAT time_hr = Sat1.ElapsedSecs / 3600;');
        % Calculate transformation matrix from ECI to Surveyor's LVLH frame
        % (Vallado pg 164)
        eval('GMAT Rmag_surveyor = sqrt(Sat1.EarthMJ2000Eq.X^2 +
Sat1.EarthMJ2000Eq.Y^2 + Sat1.EarthMJ2000Eq.Z^2);');
        eval('GMAT R1_surveyor = Sat1.EarthMJ2000Eq.X/Rmag_surveyor;');
        eval('GMAT R2_surveyor = Sat1.EarthMJ2000Eq.Y/Rmag_surveyor;');
        eval('GMAT R3_surveyor = Sat1.EarthMJ2000Eq.Z/Rmag_surveyor;');
        eval('GMAT rxv1_surveyor = Sat1.EarthMJ2000Eq.Y*Sat1.EarthMJ2000Eq.VZ
- Sat1.EarthMJ2000Eq.Z*Sat1.EarthMJ2000Eq.VY;');
        eval('GMAT rxv2_surveyor = Sat1.EarthMJ2000Eq.Z*Sat1.EarthMJ2000Eq.VX
- Sat1.EarthMJ2000Eq.X*Sat1.EarthMJ2000Eq.VZ;');
        eval('GMAT rxv3_surveyor = Sat1.EarthMJ2000Eq.X*Sat1.EarthMJ2000Eq.VY
- Sat1.EarthMJ2000Eq.Y*Sat1.EarthMJ2000Eq.VX;');
        eval('GMAT rxv_mag_surveyor = sqrt(rxv1_surveyor^2 + rxv2_surveyor^2
+ rxv3_surveyor^2);');
        eval('GMAT W1_surveyor = rxv1_surveyor/rxv_mag_surveyor;');
        eval('GMAT W2_surveyor = rxv2_surveyor/rxv_mag_surveyor;');
        eval('GMAT W3_surveyor = rxv3_surveyor/rxv_mag_surveyor;');
        eval('GMAT S1_surveyor = W2_surveyor*R3_surveyor -
W3_surveyor*R2_surveyor;');
        eval('GMAT S2_surveyor = W3_surveyor*R1_surveyor -
W1_surveyor*R3_surveyor;');
        eval('GMAT S3_surveyor = W1_surveyor*R2_surveyor -
W2_surveyor*R1_surveyor;');
        eval('GMAT r_lvlh_surveyor = R1_surveyor*Sat1.EarthMJ2000Eq.X +
R2_surveyor*Sat1.EarthMJ2000Eq.Y + R3_surveyor*Sat1.EarthMJ2000Eq.Z;');
        eval('GMAT s_lvlh_surveyor = S1_surveyor*Sat1.EarthMJ2000Eq.X +
S2_surveyor*Sat1.EarthMJ2000Eq.Y + S3_surveyor*Sat1.EarthMJ2000Eq.Z;');
        eval('GMAT w_lvlh_surveyor = W1_surveyor*Sat1.EarthMJ2000Eq.X +
W2_surveyor*Sat1.EarthMJ2000Eq.Y + W3_surveyor*Sat1.EarthMJ2000Eq.Z;');
        eval('GMAT r_lvlh_rso = R1_surveyor*Sat2.EarthMJ2000Eq.X +
R2_surveyor*Sat2.EarthMJ2000Eq.Y + R3_surveyor*Sat2.EarthMJ2000Eq.Z;');
```

```
        eval('GMAT s_lvlh_rso = S1_surveyor*Sat2.EarthMJ2000Eq.X +
S2_surveyor*Sat2.EarthMJ2000Eq.Y + S3_surveyor*Sat2.EarthMJ2000Eq.Z;');
        eval('GMAT w_lvlh_rso = W1_surveyor*Sat2.EarthMJ2000Eq.X +
W2_surveyor*Sat2.EarthMJ2000Eq.Y + W3_surveyor*Sat2.EarthMJ2000Eq.Z;');
        eval('GMAT r_rel_12 = r_lvlh_rso - r_lvlh_surveyor;');
        eval('GMAT s_rel_12 = s_lvlh_rso - s_lvlh_surveyor;');
        eval('GMAT w_rel_12 = w_lvlh_rso - w_lvlh_surveyor;');
        eval('GMAT rmag_rel = sqrt(r_rel_12^2 + s_rel_12^2 + w_rel_12^2);');
        eval('EndFor;');
    else
        if j==1
            eval(strcat('Propagate LowEarthProp(',sprintf('%s) {',...
                Sats2Prop),'Sat1.ElapsedSecs =',sprintf(' %i};',...
                satStopTime)));
        elseif j==2
            eval(strcat(...
                'Propagate BackProp LowEarthProp(Sat1) {Sat1.ElapsedSecs
=',...
                sprintf(' -%i};',satStopTime)));
        end
    end

    BuildRunGMAT % Sends all the GMAT lines into GMAT to Build & Run
    WaitForGMAT % Wait for GMAT to finish running

    %% Extract Data
    % Load raw GMAT data
    if torus==0
        for i = 1:n % satellite loop
            Report = ['ReportFile',num2str(i)];
            ReportFileLoc = strcat(...
                'C:\GmatVS11a\GmatDevelopment\application\output\',...
                sprintf('%s',Report),'.txt');
            GMAT_out = load(ReportFileLoc);
            if output==0 % limited outputs
                % Preallocation
                if i==1
                    rows = size(GMAT_out,1);
                    SatDat.time = zeros(rows,n);
                    SatDat.X = zeros(rows,n); % ECI
                    SatDat.Y = zeros(rows,n); % ECI
                    SatDat.Z = zeros(rows,n); % ECI
                    SatDat.r = zeros(rows,n);
                    SatDat.VX = zeros(rows,n); % ECI
                    SatDat.VY = zeros(rows,n); % ECI
                    SatDat.VZ = zeros(rows,n); % ECI
                    SatDat.a = zeros(rows,n);
                    SatDat.true_anom = zeros(rows,n);
                    SatDat.Xsun = zeros(rows,n); % ECI
                    SatDat.Ysun = zeros(rows,n); % ECI
                    SatDat.Zsun = zeros(rows,n); % ECI
                    SatDat.shadow = -1.*ones(rows,n);
                    SatDat.eclipse_entry_pen_start = zeros(100,n);
                    SatDat.eclipse_entry_umb_start = zeros(100,n);
                    SatDat.eclipse_exit_umb_stop = zeros(100,n);
                    SatDat.eclipse_exit_pen_stop = zeros(100,n);
                end

                SatDat.time(:,i) = GMAT_out(:,1);
                SatDat.X(:,i) = GMAT_out(:,2); % ECI
                SatDat.Y(:,i) = GMAT_out(:,3); % ECI
                SatDat.Z(:,i) = GMAT_out(:,4); % ECI
                SatDat.VX(:,i) = GMAT_out(:,5); % ECI
                SatDat.VY(:,i) = GMAT_out(:,6); % ECI
                SatDat.VZ(:,i) = GMAT_out(:,7); % ECI
                SatDat.a(:,i) = GMAT_out(:,8);
                SatDat.true_anom(:,i) = GMAT_out(:,9);
                SatDat.r(:,i) = GMAT_out(:,10);
                SatDat.Xsun(:,i) = SatDat.X(:,i) - GMAT_out(:,11); % ECI
                SatDat.Ysun(:,i) = SatDat.Y(:,i) - GMAT_out(:,12); % ECI
                SatDat.Zsun(:,i) = SatDat.Z(:,i) - GMAT_out(:,13); % ECI
            else % normal outputs
                % Preallocation
                if i==1
                    rows = size(GMAT_out,1);
                    SatDat.time = zeros(rows,n);
                    SatDat.UTCMJ = zeros(rows,n);
                    SatDat.X = zeros(rows,n); % ECI
                    SatDat.Y = zeros(rows,n); % ECI
                    SatDat.Z = zeros(rows,n); % ECI
                    SatDat.Xef = zeros(rows,n); % ECEF
                    SatDat.Yef = zeros(rows,n); % ECEF
                    SatDat.Zef = zeros(rows,n); % ECEF
                    SatDat.r = zeros(rows,n);
                    SatDat.VX = zeros(rows,n); % ECI
                    SatDat.VY = zeros(rows,n); % ECI
                    SatDat.VZ = zeros(rows,n); % ECI
                    SatDat.VXef = zeros(rows,n); % ECEF
                    SatDat.VYef = zeros(rows,n); % ECEF
                    SatDat.VZef = zeros(rows,n); % ECEF
                    SatDat.a = zeros(rows,n);
                    SatDat.e = zeros(rows,n);
                    SatDat.i = zeros(rows,n); % ECI
                    SatDat.RAAN = zeros(rows,n); % ECI
                    SatDat.arg_per = zeros(rows,n); % ECI
                    SatDat.true_anom = zeros(rows,n);
                    SatDat.mean_anom = zeros(rows,n);
                    SatDat.Xsun = zeros(rows,n); % ECI
                    SatDat.Ysun = zeros(rows,n); % ECI
                    SatDat.Zsun = zeros(rows,n); % ECI
                    SatDat.shadow = -1.*ones(rows,n);
                    SatDat.eclipse_entry_pen_start = zeros(rows,n);
                    SatDat.eclipse_entry_umb_start = zeros(rows,n);
                    SatDat.eclipse_exit_umb_stop = zeros(rows,n);
                    SatDat.eclipse_exit_pen_stop = zeros(rows,n);
                end

                SatDat.time(:,i) = GMAT_out(:,1);
                SatDat.Xef(:,i) = GMAT_out(:,2); % ECEF
                SatDat.Yef(:,i) = GMAT_out(:,3); % ECEF
                SatDat.Zef(:,i) = GMAT_out(:,4); % ECEF
                SatDat.VXef(:,i) = GMAT_out(:,5); % ECEF
```

```
                SatDat.VYef(:,i) = GMAT_out(:,6); % ECEF
                SatDat.VZef(:,i) = GMAT_out(:,7); % ECEF
                SatDat.UTCMJ(:,i) = GMAT_out(:,8);
                SatDat.X(:,i) = GMAT_out(:,9); % ECI
                SatDat.Y(:,i) = GMAT_out(:,10); % ECI
                SatDat.Z(:,i) = GMAT_out(:,11); % ECI
                SatDat.VX(:,i) = GMAT_out(:,12); % ECI
                SatDat.VY(:,i) = GMAT_out(:,13); % ECI
                SatDat.VZ(:,i) = GMAT_out(:,14); % ECI
                SatDat.a(:,i) = GMAT_out(:,15);
                SatDat.e(:,i) = GMAT_out(:,16);
                SatDat.i(:,i) = GMAT_out(:,17); % ECI
                SatDat.RAAN(:,i) = GMAT_out(:,18); % ECI
                SatDat.arg_per(:,i) = GMAT_out(:,19); % ECI
                SatDat.true_anom(:,i) = GMAT_out(:,20);
                SatDat.mean_anom(:,i) = GMAT_out(:,21);
                SatDat.r(:,i) = GMAT_out(:,22);
                SatDat.Xsun(:,i) = SatDat.X(:,i) - GMAT_out(:,23); % ECI
                SatDat.Ysun(:,i) = SatDat.Y(:,i) - GMAT_out(:,24); % ECI
                SatDat.Zsun(:,i) = SatDat.Z(:,i) - GMAT_out(:,25); % ECI
            end
            % Determine eclipse state
            shadow;
        end
    end
end

% Load raw GMAT data
if torus==1
    % clear all
    forward = load(...
        'C:\GmatVS11a\GmatDevelopment\application\output\forward.txt');
    backward = load(...
        'C:\GmatVS11a\GmatDevelopment\application\output\backward.txt');
    % Combine forward- and backward-propagated data if building torus
    GMAT_out = [flipud(backward(2:end,:));forward];
    SatDat.t = GMAT_out(:,1) / 60 / 13.446852; % sec to TU
    SatDat.x = GMAT_out(:,2) / 6378.135; % km to DU
    SatDat.y = GMAT_out(:,3) / 6378.135; % km to DU
    SatDat.z = GMAT_out(:,4) / 6378.135; % km to DU
    SatDat.vx = GMAT_out(:,5) / 6378.135 * 60 * 13.446852; % km/sec to DU/TU
    SatDat.vy = GMAT_out(:,6) / 6378.135 * 60 * 13.446852; % km/sec to DU/TU
    SatDat.vz = GMAT_out(:,7) / 6378.135 * 60 * 13.446852; % km/sec to DU/TU
    %clear backward forward GMAT_out

save('C:\GmatVS11a\GmatDevelopment\application\output\propagated_orbit.mat',...
        'SatDat')
end
```

E.4 Line-of-Sight Data Processing

```
%  [function]    losProc
%
%  [purpose]:    Function processes the GMAT ECI data to form
%                line-of-sight (LOS) data products in the local-vertical
%                local-horizontal (LVLH) frame.
%
%  [useage]:     [Surveyor,Target] = losProc(SatDat,input,options)
%
%  [inputs]:     SatDat    = Structure containing the GMAT extracted data.
%                            The exact content is configured in the
%                            gmatOrbitProp code, but presently includes
%                            for every ith satellite entries of the form:
%                            SatDat.X(:,i).
%                            NOTE: The 1st satellite is the Surveyor! All
%                            others are Target satellites.
%
%                input     = Input structure for the Surveyor and Target
%                            which includes the true elements and epoch of
%                            initial conditions.
%
%                options   = [bias,stdev,rate_calcs]
%                            bias = angle measurement bias [arc sec]
%                            stdev = 1-sigma angle meas uncertainty [arc sec]
%                            rate_calcs = boolean (1 turns on, 0 turns off)
%                               for whether to calculate angle rates and
%                               find min/max angle rates.
%
%  [outputs]:    Surveyor  = Structure containing the processed data for
%                            the Surveyor.
%
%                Target    = Structure containing the processed data for
%                            the Target.
%
%  [refs]:
%
%  [history]:    25 Oct 2016 - Max Yates

function [Surveyor,Target] = losProc(SatDat,input,options)

global MU

codeDir = pwd;

bias = options(1);
stdev = options(2);
rate_calcs = logical(options(3));

rows = size(SatDat.time,1);
n = size(SatDat.time,2)-1; % number of Target satellites
% sampling frequency for returned data rounded to nearest tenth [sec]
dt = roundn(SatDat.time(2,1)-SatDat.time(1,1),-2);

%% Preallocation
```

```
Surveyor.Xrsw = zeros(rows,1);
Surveyor.Yrsw = zeros(rows,1);
Surveyor.Zrsw = zeros(rows,1);
Surveyor.rotECI2RSW = zeros(3,3,rows);
Target.Xrel = zeros(rows,n);
Target.Yrel = zeros(rows,n);
Target.Zrel = zeros(rows,n);
Target.Xrel_Vallado = zeros(rows,n);
Target.Yrel_Vallado = zeros(rows,n);
Target.Zrel_Vallado = zeros(rows,n);
Target.VXrel_num = zeros(rows,n);
Target.VYrel_num = zeros(rows,n);
Target.VZrel_num = zeros(rows,n);
Target.VXrel_Vallado = zeros(rows,n);
Target.VYrel_Vallado = zeros(rows,n);
Target.VZrel_Vallado = zeros(rows,n);
Target.rho_true = zeros(rows,n);

%% ANGLES-ONLY PROCESSING
% Calculate transformation matrix from ECI to Surveyor's RSW frame
% (Vallado pg164)
reci = [SatDat.X(:,1),SatDat.Y(:,1),SatDat.Z(:,1)];
veci = [SatDat.VX(:,1),SatDat.VY(:,1),SatDat.VZ(:,1)];
% radial component unit vectors
twoNorm = sqrt(sum(abs(reci).^2,2));
rvec = [reci(:,1)./twoNorm,reci(:,2)./twoNorm,reci(:,3)./twoNorm];
% cross-track component unit vectors
wvec = cross(reci,veci);
twoNorm = sqrt(sum(abs(wvec).^2,2));
wvec = [wvec(:,1)./twoNorm,wvec(:,2)./twoNorm,wvec(:,3)./twoNorm];
% along-track component
svec = cross(wvec,rvec);

% Assemble transformation matrix from eci to rsw frame (individual
% components arranged in row vectors)
Surveyor.rotECI2RSW(1,1,:) = rvec(:,1);
Surveyor.rotECI2RSW(1,2,:) = rvec(:,2);
Surveyor.rotECI2RSW(1,3,:) = rvec(:,3);
Surveyor.rotECI2RSW(2,1,:) = svec(:,1);
Surveyor.rotECI2RSW(2,2,:) = svec(:,2);
Surveyor.rotECI2RSW(2,3,:) = svec(:,3);
Surveyor.rotECI2RSW(3,1,:) = wvec(:,1);
Surveyor.rotECI2RSW(3,2,:) = wvec(:,2);
Surveyor.rotECI2RSW(3,3,:) = wvec(:,3);

% transformation matrix from rsw to eci
% rotRSW2ECI(:,:,j) = rotECI2RSW(:,:,j)';

clear rvec svec wvec twoNorm

for i=1:n % loop through Target satellites
    % true range from inertial coordinates
    Target.rho_true(:,i) = sqrt((SatDat.X(:,i+1)-SatDat.X(:,1)).^2+...
        (SatDat.Y(:,i+1)-SatDat.Y(:,1)).^2+(SatDat.Z(:,i+1)-
SatDat.Z(:,1)).^2);
```

```
% CALCULATE POSITION & VELOCITY in Surveyor's RSW frame
for j=1:rows
    if i==1
        surveyor_r_rsw = Surveyor.rotECI2RSW(:,:,j)*reci(j,:)';
        Surveyor.Xrsw(j,1) = surveyor_r_rsw(1,1);
        Surveyor.Yrsw(j,1) = surveyor_r_rsw(2,1);
        Surveyor.Zrsw(j,1) = surveyor_r_rsw(3,1);
    end
    target_r_rsw = Surveyor.rotECI2RSW(:,:,j)*[SatDat.X(j,i+1),...
        SatDat.Y(j,i+1),SatDat.Z(j,i+1)]';
    % The target's relative distance and velocity in the RSW frame
    % NOTE: not storing Xrsw for Targets, just calculating for each
    % sat to get Xrel
    Target.Xrel(j,i) = target_r_rsw(1,1) - Surveyor.Xrsw(j,1); % km
    Target.Yrel(j,i) = target_r_rsw(2,1) - Surveyor.Yrsw(j,1); % km
    Target.Zrel(j,i) = target_r_rsw(3,1) - Surveyor.Zrsw(j,1); % km
end
% MUST use seperate for loop because numerical calculation uses
% coordinates forward in time
for j=1:rows
    % Numerically Calculate Target's Relative Velocity - five-point
    % central difference formula
    if j>2 && j<(rows-1)
        Target.VXrel_num(j,i) = (Target.Xrel(j-2,i) - ...
            8*Target.Xrel(j-1,i) + 8*Target.Xrel(j+1,i) - ...
            Target.Xrel(j+2,i))/(12*dt);
        Target.VYrel_num(j,i) = (Target.Yrel(j-2,i) - ...
            8*Target.Yrel(j-1,i) + 8*Target.Yrel(j+1,i) - ...
            Target.Yrel(j+2,i))/(12*dt);
        Target.VZrel_num(j,i) = (Target.Zrel(j-2,i) - ...
            8*Target.Zrel(j-1,i) + 8*Target.Zrel(j+1,i) - ...
            Target.Zrel(j+2,i))/(12*dt);
    elseif j<3 % forward difference formula
        Target.VXrel_num(j,i) = (-11*Target.Xrel(j,i) + ...
            18*Target.Xrel(j+1,i) - 9*Target.Xrel(j+2,i) + ...
            2*Target.Xrel(j+3,i))/(6*dt);
        Target.VYrel_num(j,i) = (-11*Target.Yrel(j,i) + ...
            18*Target.Yrel(j+1,i) - 9*Target.Yrel(j+2,i) + ...
            2*Target.Yrel(j+3,i))/(6*dt);
        Target.VZrel_num(j,i) = (-11*Target.Zrel(j,i) + ...
            18*Target.Zrel(j+1,i) - 9*Target.Zrel(j+2,i) + ...
            2*Target.Zrel(j+3,i))/(6*dt);
    elseif j>(rows-2) % backward difference formula
        Target.VXrel_num(j,i) = (11*Target.Xrel(j,i) - ...
            18*Target.Xrel(j-1,i) + 9*Target.Xrel(j-2,i) - ...
            2*Target.Xrel(j-3,i))/(6*dt);
        Target.VYrel_num(j,i) = (11*Target.Yrel(j,i) - ...
            18*Target.Yrel(j-1,i) + 9*Target.Yrel(j-2,i) - ...
            2*Target.Yrel(j-3,i))/(6*dt);
        Target.VZrel_num(j,i) = (11*Target.Zrel(j,i) - ...
            18*Target.Zrel(j-1,i) + 9*Target.Zrel(j-2,i) - ...
            2*Target.Zrel(j-3,i))/(6*dt);
    end

    % Use Vallado to check relative coordinates
    [rhill,vhill] = eci2hill(...
            [SatDat.X(j,1),SatDat.Y(j,1),SatDat.Z(j,1)],...
            [SatDat.VX(j,1),SatDat.VY(j,1),SatDat.VZ(j,1)],...
            [SatDat.X(j,i+1),SatDat.Y(j,i+1),SatDat.Z(j,i+1)],...
            [SatDat.VX(j,i+1),SatDat.VY(j,i+1),SatDat.VZ(j,i+1)]);
        Target.Xrel_Vallado(j,i) = rhill(1,1);
        Target.Yrel_Vallado(j,i) = rhill(1,2);
        Target.Zrel_Vallado(j,i) = rhill(1,3);
        Target.VXrel_Vallado(j,i) = vhill(1,1);
        Target.VYrel_Vallado(j,i) = vhill(1,2);
        Target.VZrel_Vallado(j,i) = vhill(1,3);
    end
end
clear reci veci

%% Preallocation
Target.alpha = zeros(rows,n);
Target.beta = zeros(rows,n);
Target.alpha_true = zeros(rows,n);
Target.beta_true = zeros(rows,n);
Target.alpha_dot = zeros(rows,n);
Target.alpha_dot2 = zeros(rows,n);
Target.r_dot = zeros(rows,n);
Target.eclipse_delta_start = zeros(1,n);
Target.eclipse_range_true = zeros(1,n);
Target.eclipse_target_idx = zeros(1,n);
Target.max_alphadot = nan(2,n);
Target.max_alphadot_idx = nan(2,n);
Target.alpha_at_max_alphadot = nan(2,n);
Target.range_at_max_alphadot = nan(2,n);
Target.min_alphadot = nan(2,n);
Target.min_alphadot_idx = nan(2,n);
Target.alpha_at_min_alphadot = nan(2,n);
Target.range_at_min_alphadot = nan(2,n);

% Had to break up the satellite for loop to guard against memory issues
for i=1:n % loop through Target satellites
    %% Calculate angles
    bias = bias/3600*pi/180; % [rad]
    stdev = stdev/3600*pi/180; % [rad]
    Target.alpha_true(:,i) = atan2(Target.Yrel(:,i),Target.Xrel(:,i));
    Target.beta_true(:,i) = atan2(Target.Zrel(:,i),...
        sqrt(Target.Xrel(:,i).^2+Target.Yrel(:,i).^2));
    randn('seed',0); % seed for random number generator
    Target.alpha(:,i) = atan2(Target.Yrel(:,i),Target.Xrel(:,i)) + ...
        bias + stdev*randn(rows,1);
    randn('seed',0); % seed for random number generator
    Target.beta(:,i) = atan2(Target.Zrel(:,i),...
        sqrt(Target.Xrel(:,i).^2+Target.Yrel(:,i).^2)) + ...
        bias + stdev*randn(rows,1);
    % Unwrap angle data (no 360 deg jumps)
    tol = pi;
    % uses a jump tolerance tol instead of the default value, pi.
    Target.alpha(:,i) = unwrap(Target.alpha(:,i),tol);
    if Target.alpha(end,i) < -2*pi
        Target.alpha(:,i) = Target.alpha(:,i) + 2*pi;
    end
```

```
Target.beta(:,i) = unwrap(Target.beta(:,i),tol);
Target.alpha_meas = Target.alpha; % unwrapped original measurements
Target.beta_meas = Target.beta; % unwrapped original measurements

if stdev > 0
    %% Fit a curve through alpha and beta profiles
    % Piece together multiple polyfit segments. Choose break points
    % that are small enough to fit cubics with high confidence.
    % How many break points:
    num_breaks = floor(rows/1000)+1; % add 2 for first and last
    break_pts = zeros(1,num_breaks);
    break_pts(1,1) = 1;
    break_pts(1,end) = rows;
    ws = warning('off','all');  % Turn off warning
    for j=1:(num_breaks-1)
        if j<num_breaks-1
            break_pts(1,j+1) = j*1000;
        end
        % 3rd degree polynomial (cubic)
        p = polyfit(SatDat.time(break_pts(j):break_pts(j+1),1),...
            Target.alpha(break_pts(j):break_pts(j+1),i),3);
        Target.alpha(break_pts(j):break_pts(j+1),i) = polyval(...
            p,SatDat.time(break_pts(j):break_pts(j+1),1));
        p = polyfit(SatDat.time(break_pts(j):break_pts(j+1),1),...
            Target.beta(break_pts(j):break_pts(j+1),i),3);
        Target.beta(break_pts(j):break_pts(j+1),i) = polyval(p,...
            SatDat.time(break_pts(j):break_pts(j+1),1));
    end
    warning(ws)  % Turn it back on.

    % Check how well the polynomials fit the angle data
    A = unwrap(Target.alpha_true(:,i),tol);
    if A(end,1) < -2*pi
        A = A + 2*pi;
    end
    B = unwrap(Target.beta_true(:,i),tol);
    figure(25)
    plot(SatDat.time(:,1)/60,(Target.alpha(:,1)-A)*180/pi*3600,'-r',...
        'LineWidth',2) % [arc secs]
    hold on
    plot(SatDat.time(:,1)/60,(Target.beta(:,1)-B)*180/pi*3600,'--b',...
        'LineWidth',2) % [arc secs]
    grid on; grid minor;
    xlabel('\boldmath$Time\ (min)$','Interpreter','latex',...
        'FontSize',33)
    ylabel('\boldmath$Error\ (arc\ secs)$','Interpreter','latex',...
        'FontSize',33)
    set(gca,'FontSize',32,'FontWeight','bold')
    h = legend('\boldmath$\alpha$','\boldmath$\beta$','Location',...
        'SouthWest');
    set(h,'Interpreter','latex','FontSize',32);
    axis normal;
    saveas(gcf,strcat(codeDir,'\Cases\',sprintf('%s',input.runCase),...
        '\Fig25.fig'))
    close
end
```

```
% The sine of alpha can be searched for zero crossings. No zero
% crossings indicates the relative trajectory does not enclose the
% Surveyor.
signum = sign(sin(Target.alpha(:,i))); % get sign of data
% set sign of exact data zeros to positive
signum(sin(Target.alpha(:,i))==0) = 1;
% get zero crossings by diff ~= 0
zero_crossings = find(diff(signum)~=0);
if isempty(zero_crossings)==0 && rate_calcs % only enclosed rel orbits
    %% Calculate alpha rate (as defined, alpha is in the rs-plane)
    if stdev == 0
        for j=1:rows
            % alpha rate from five-point central difference formula
            if j>2 && j<(rows-1)
                if max(Target.alpha(j-2:j+2,i))-min(...
                        Target.alpha(j-2:j+2,i))>pi
                    alpha_wrap = check_angle_wrap(...
                        Target.alpha(j-2:j+2,i));
                else
                    alpha_wrap = Target.alpha(j-2:j+2,i);
                end
                Target.alpha_dot(j,i) = (alpha_wrap(1,1) - ...
                    8*alpha_wrap(2,1) + 8*alpha_wrap(4,1) - ...
                    alpha_wrap(5,1))/(12*dt);
            elseif j<3 % forward difference formula
                if max(Target.alpha(j:j+3,i))-min(...
                        Target.alpha(j:j+3,i))>pi
                    alpha_wrap = check_angle_wrap(...
                        Target.alpha(j:j+3,i));
                else
                    alpha_wrap = Target.alpha(j:j+3,i);
                end
                Target.alpha_dot(j,i) = (-11*alpha_wrap(1,1) + ...
                    18*alpha_wrap(2,1) - 9*alpha_wrap(3,1) + ...
                    2*alpha_wrap(4,1))/(6*dt);
            elseif j>(rows-2) % backward difference formula
                if max(Target.alpha(j-3:j,i))-min(...
                        Target.alpha(j-3:j,i))>pi
                    alpha_wrap = check_angle_wrap(...
                        Target.alpha(j-3:j,i));
                else
                    alpha_wrap = Target.alpha(j-3:j,i);
                end
                Target.alpha_dot(j,i) = (11*alpha_wrap(4,1) - ...
                    18*alpha_wrap(3,1) + 9*alpha_wrap(2,1) - ...
                    2*alpha_wrap(1,1))/(6*dt);
                if abs(Target.alpha_dot(j,i)) > ...
                        1.1*abs(Target.alpha_dot(j-1,i))
                    Target.alpha_dot(j,i) = Target.alpha_dot(j-1,i);
                end
            end
            % alpha_dot calculated from Vallado pg 258
            Target.alpha_dot2(j,i) = (Target.VXrel_num(j,i)*...
                Target.Yrel(j,i) - Target.VYrel_num(j,i)*...
                Target.Xrel(j,i))/(-Target.Yrel(j,i)^2 - ...
                Target.Xrel(j,i)^2);
            Target.r_dot(j,i) = (Target.Xrel(j,i)*...
                Target.VXrel_num(j,i) + Target.Yrel(j,i)*...
                Target.VYrel_num(j,i) + Target.Zrel(j,i)*...
                Target.VZrel_num(j,i))/Target.rho_true(j,i);
        end

    elseif stdev > 0
        for k = 1:(size(break_pts,2)-1)
            for j=break_pts(k):break_pts(k+1)
                % alpha rate from five-point central difference formula
                if j>(break_pts(k)+1) && j<(break_pts(k+1)-1)
                    Target.alpha_dot(j,i) = (Target.alpha(j-2,i) - ...
                        8*Target.alpha(j-1,i) + 8*Target.alpha(j+1,i)...
                        - Target.alpha(j+2,i))/(12*dt);
                    if abs(Target.alpha_dot(j,i)) > 1.1*abs(...
                            Target.alpha_dot(j-1,i))
                        Target.alpha_dot(j,i) = Target.alpha_dot(j-1,i);
                    end
                end
                if j<(break_pts(k)+2) % forward difference formula
                    Target.alpha_dot(j,i) = (-11*Target.alpha(j,i) +...
                        18*Target.alpha(j+1,i)-9*Target.alpha(j+2,i)...
                        + 2*Target.alpha(j+3,i))/(6*dt);
                end
                % backward difference formula
                % normally -2, but using -5 to catch some typical errors
                if j>(break_pts(k+1)-5)
                    Target.alpha_dot(j,i) = (11*Target.alpha(j,i) - ...
                        18*Target.alpha(j-1,i)+9*Target.alpha(j-2,i)...
                        - 2*Target.alpha(j-3,i))/(6*dt);
                    if abs(Target.alpha_dot(j,i)) > ...
                            1.1*abs(Target.alpha_dot(j-1,i))
                        Target.alpha_dot(j,i) = Target.alpha_dot(j-1,i);
                    end
                end
                % alpha_dot calculated from Vallado pg 258
                Target.alpha_dot2(j,i) = (Target.VXrel_num(j,i)*...
                    Target.Yrel(j,i) - Target.VYrel_num(j,i)*...
                    Target.Xrel(j,i))/(-Target.Yrel(j,i)^2 - ...
                    Target.Xrel(j,i)^2);
                Target.r_dot(j,i) = (Target.Xrel(j,i)*...
                    Target.VXrel_num(j,i) + Target.Yrel(j,i)*...
                    Target.VYrel_num(j,i) + Target.Zrel(j,i)*...
                    Target.VZrel_num(j,i))/Target.rho_true(j,i);
            end
        end
        % Compare the estimated rate data to the truth
        figure(26)
        plot(Target.alpha(:,i)*180/pi,...
            (Target.alpha_dot-Target.alpha_dot2)*180/pi*3600,...
            'LineWidth',2) % [arc secs/sec]
        grid on; grid minor;
        xlabel('\boldmath$\alpha_{true}\ (arc\ secs)$','Interpreter',...
            'latex','FontSize',33)
        ylabel('\boldmath$\dot{\alpha}\ Error\ (arc\ secs/sec)$',...
        'Interpreter','latex','FontSize',33)
    set(gca,'FontSize',32,'FontWeight','bold')
    axis normal;
    saveas(gcf,strcat(codeDir,'\Cases\',sprintf('%s',...
        input.runCase),'\Fig26.fig'))
    close
end

%% Find where alpha rate is minimum
% Find indices where alpha rate is minimum in the unsmoothed data
% (Unsmoothed still describes the sequence of curvefits, due to the
% breakpoints. A polynomial will be fit to the data in the vicinity
% of the minimum.)
% https://www.mathworks.com/matlabcentral/fileexchange/25500-pea...
% kfinder-x0--sel--thresh--extrema--includeendpoints--interpolate-
[min_alphadot_idx, min_alphadot] = peakfinder(...
    Target.alpha_dot(:,i), 1e-5, -100, 1, false, false); %1e-6
if size(min_alphadot,1)>2
    % just ignore the last one
    min_alphadot_idx = min_alphadot_idx(1:2,1);
    min_alphadot = min_alphadot(1:2,1);
end
if size(min_alphadot,1)>1 && min_alphadot_idx(2,1) > ...
        (size(Target.alpha_dot(:,i),1)-5)
    % ignore the last one (garbage)
    min_alphadot_idx = min_alphadot_idx(1,1);
    % ignore the last one (garbage)
    min_alphadot = min_alphadot(1,1);
end

% Peakfinder is unreliable sometimes... need to troubleshoot later,
% but for now, use the old method:
if size(min_alphadot)==[0,0]
    % Old Method...: FILTER alpha_dot BEFORE IT CAN BE USED -
    % temp use of alpha_dot2 since it will give the same result
    [min_alphadot,min_alphadot_idx] = findpeaks(...
        Target.alpha_dot2(:,i));
    if isempty(min_alphadot)==1
        continue;
        % If da is positive & large enough, certain relative orbits
        % don't have min angular rate peaks. V shapeed, instead.
        % This occured through the sma_approx where angular rates
        % weren't needed. May have to revisit this later.
    end
elseif (size(min_alphadot,1)==2 && isequal(min_alphadot_idx(1,1),...
        min_alphadot_idx(2,1)))
    % Old Method...: FILTER alpha_dot BEFORE IT CAN BE USED -
    % temp use of alpha_dot2 since it will give the same result
    [min_alphadot,min_alphadot_idx] = findpeaks(...
        Target.alpha_dot2(:,i));
    if isempty(min_alphadot)==1
        continue;
        % If da is positive & large enough, certain relative orbits
        % don't have min angular rate peaks. V shapeed, instead.
        % This occured through the sma_approx where angular rates
        % weren't needed. May have to revisit this later.
    end
end

% Polyfit to smooth data
T = 2*pi*sqrt(SatDat.a(1,1)^3/MU);
ref = round(0.055*T/dt);
% For highly skewed NMC, there is only one min_alphadot!
if length(min_alphadot_idx)==1 || abs(min_alphadot(1,1)) < ...
        abs(min_alphadot(2,1))
    idx11 = min_alphadot_idx(1,1) - ref;
    idx12 = min_alphadot_idx(1,1) + ref;
    poly_deg = 5;
else
    idx11 = min_alphadot_idx(1,1) - ref;
    idx12 = min_alphadot_idx(1,1) + ref;
    poly_deg = 5;
end
if idx11 < 1
    idx11 = 1;
end
if idx12 > size(SatDat.time,1)
    idx12 = size(SatDat.time,1);
end
if stdev > 0
    % Re-fit curve to alpha in this smaller segment -- need to
    % estimate alphadot better here
    ws = warning('off','all');  % Turn off warning
    p = polyfit(SatDat.time(idx11:idx12,1),...
        Target.alpha_meas(idx11:idx12,i),poly_deg);
    Target.alpha(idx11:idx12,i) = polyval(...
        p,SatDat.time(idx11:idx12,1));
    warning(ws)  % Turn it back on.
    % Recalculate alphadot in this smaller segment
    for j=idx11:idx12
        % alpha rate from five-point central difference formula
        if j>(idx11+1) && j<(idx12-1)
            Target.alpha_dot(j,i) = (Target.alpha(j-2,i) - ...
                8*Target.alpha(j-1,i) + 8*Target.alpha(j+1,i) - ...
                Target.alpha(j+2,i))/(12*dt);
            if abs(Target.alpha_dot(j,i)) > ...
                    1.1*abs(Target.alpha_dot(j-1,i))
                Target.alpha_dot(j,i) = Target.alpha_dot(j-1,i);
            end
        end
        if j<(idx11+2) % forward difference formula
            Target.alpha_dot(j,i) = (-11*Target.alpha(j,i) + ...
                18*Target.alpha(j+1,i) - 9*Target.alpha(j+2,i) +...
                2*Target.alpha(j+3,i))/(6*dt);
        end
        % backward difference formula
        % normally -2, but using -5 to catch some typical errors
        if j>(idx12-5)
            Target.alpha_dot(j,i) = (11*Target.alpha(j,i) - ...
                18*Target.alpha(j-1,i) + 9*Target.alpha(j-2,i) -...
                2*Target.alpha(j-3,i))/(6*dt);
            if abs(Target.alpha_dot(j,i)) > ...
```

```
                    1.1*abs(Target.alpha_dot(j-1,i))
                Target.alpha_dot(j,i) = Target.alpha_dot(j-1,i);
            end
        end
    end
end
ws = warning('off','all');  % Turn off warning
p1 = polyfit(SatDat.time(idx11:idx12,1),...
    Target.alpha_dot(idx11:idx12,i),4);
warning(ws)  % Turn it back on.
Target.alpha_dot(idx11:idx12,i) = polyval(...
    p1,SatDat.time(idx11:idx12,1));

% Find indices where alpha rate is minimum in the smoothed data
[min_alphadot1,min_alphadot_idx1] = findpeaks(...
    Target.alpha_dot(idx11:idx12,i));

% For highly skewed NMC, there is only one min_alphadot!
if length(min_alphadot_idx)>1
    if abs(min_alphadot(2,1)) < abs(min_alphadot(1,1))
        idx21 = min_alphadot_idx(2,1) - ref;
        idx22 = min_alphadot_idx(2,1) + ref;
        poly_deg = 5;
    else
        idx21 = min_alphadot_idx(2,1) - ref;
        idx22 = min_alphadot_idx(2,1) + ref;
        poly_deg = 5;
    end
    if idx21 < 1
        idx21 = 1;
    end
    if idx22 > size(SatDat.time,1)
        idx22 = size(SatDat.time,1);
    end
    if stdev > 0
        % Re-fit curve to alpha in this smaller segment -- need to
        % estimate alphadot better here
        ws = warning('off','all');  % Turn off warning
        p = polyfit(SatDat.time(idx21:idx22,1),...
            Target.alpha_meas(idx21:idx22,i),poly_deg);
        Target.alpha(idx21:idx22,i) = polyval(...
            p,SatDat.time(idx21:idx22,1));
        warning(ws)  % Turn it back on.
        % Recalculate alphadot in this smaller segment
        for j=idx21:idx22
            % alpha rate from five-point central difference formula
            if j>(idx21+1) && j<(idx22-1)
                Target.alpha_dot(j,i) = (Target.alpha(j-2,i) - ...
                    8*Target.alpha(j-1,i) + 8*Target.alpha(j+1,i)...
                    - Target.alpha(j+2,i))/(12*dt);
                if abs(Target.alpha_dot(j,i)) > ...
                        1.1*abs(Target.alpha_dot(j-1,i))
                    Target.alpha_dot(j,i) = Target.alpha_dot(j-1,i);
                end
            end
            if j<(idx21+2) % forward difference formula
```

```
                Target.alpha_dot(j,i) = (-11*Target.alpha(j,i) +...
                    18*Target.alpha(j+1,i) -9*Target.alpha(j+2,i)...
                    + 2*Target.alpha(j+3,i))/(6*dt);
            end
            % backward difference formula
            % normally -2, but using -5 to catch some typical errors
            if j>(idx22-5)
                Target.alpha_dot(j,i) = (11*Target.alpha(j,i) - ...
                    18*Target.alpha(j-1,i) +9*Target.alpha(j-2,i)...
                    - 2*Target.alpha(j-3,i))/(6*dt);
                if abs(Target.alpha_dot(j,i)) > ...
                        1.1*abs(Target.alpha_dot(j-1,i))
                    Target.alpha_dot(j,i) = Target.alpha_dot(j-1,i);
                end
            end
        end
    end
    ws = warning('off','all');  % Turn off warning
    p2 = polyfit(SatDat.time(idx21:idx22,1),...
        Target.alpha_dot(idx21:idx22,i),4);
    warning(ws)  % Turn it back on.
    Target.alpha_dot(idx21:idx22,i) = polyval(...
        p2,SatDat.time(idx21:idx22,1));
    [min_alphadot2,min_alphadot_idx2] = findpeaks(...
        Target.alpha_dot(idx21:idx22,i));
    Target.min_alphadot(:,i) = [min_alphadot1;min_alphadot2]*...
        180/pi*3600; % asecs/sec
    Target.min_alphadot_idx(:,i) = [idx11+min_alphadot_idx1-1;...
        idx21+min_alphadot_idx2-1];

    Target.alpha(:,i) = wrap(Target.alpha(:,i));

    if Target.alpha(Target.min_alphadot_idx(1,i),i)>0
        Target.alpha_at_min_alphadot(1,i) = (Target.alpha(...
            Target.min_alphadot_idx(1,i),i)*180/pi-90)*3600; % asecs
        Target.alpha_at_min_alphadot(2,i) = (Target.alpha(...
            Target.min_alphadot_idx(2,i),i)*180/pi+90)*3600; % asecs
    else
        Target.alpha_at_min_alphadot(1,i) = (Target.alpha(...
            Target.min_alphadot_idx(1,i),i)*180/pi+90)*3600; % asecs
        Target.alpha_at_min_alphadot(2,i) = (Target.alpha(...
            Target.min_alphadot_idx(2,i),i)*180/pi-90)*3600; % asecs
    end
    Target.range_at_min_alphadot(1,i) = ...
        Target.rho_true(Target.min_alphadot_idx(1,i),i);
    Target.range_at_min_alphadot(2,i) = ...
        Target.rho_true(Target.min_alphadot_idx(2,i),i);
else
    Target.min_alphadot(1,i) = min_alphadot1*180/pi*3600; % asecs/sec
    Target.min_alphadot_idx(1,i) = idx11+min_alphadot_idx1-1;

    Target.alpha(:,i) = wrap(Target.alpha(:,i));

    if Target.alpha(Target.min_alphadot_idx(1,i),i)>0
        Target.alpha_at_min_alphadot(1,i) = (Target.alpha(...
            Target.min_alphadot_idx(1,i),i)*180/pi-90)*3600; % asecs
    else
        Target.alpha_at_min_alphadot(1,i) = (Target.alpha(...
            Target.min_alphadot_idx(1,i),i)*180/pi+90)*3600; % asecs
    end
    Target.range_at_min_alphadot(1,i) = ...
        Target.rho_true(Target.min_alphadot_idx(1,i),i);
end

%% Find where alpha rate is maximum
% uses a jump tolerance tol instead of the default value, pi.
Target.alpha(:,i) = unwrap(Target.alpha(:,i),tol);
if Target.alpha(end,i) < -2*pi
    Target.alpha(:,i) = Target.alpha(:,i) + 2*pi;
end
[max_alphadot_idx, max_alphadot] = peakfinder(...
    Target.alpha_dot(:,i), 1e-5, 100, -1, false, false); %1e-6
if size(max_alphadot,1)>2
    % use the middle peak and ignore the two outer ones
    max_alphadot = max_alphadot(2,1);
    max_alphadot_idx = max_alphadot_idx(2,1);
end
if size(max_alphadot,1)>1 && max_alphadot_idx(2,1) > ...
        (size(Target.alpha_dot(:,i),1)-5)
    % ignore the last one (garbage)
    max_alphadot_idx = max_alphadot_idx(1,1);
    % ignore the last one (garbage)
    max_alphadot = max_alphadot(1,1);
end

% Peakfinder is unreliable sometimes... need to troubleshoot later,
% but for now, use the old method:
if size(max_alphadot)==[0,0]
    % Old Method...: FILTER alpha_dot BEFORE IT CAN BE USED -
    % temp use of alpha_dot2 since it will give the same result
    [max_alphadot,max_alphadot_idx] = findpeaks(...
        -Target.alpha_dot2(:,i));
elseif size(max_alphadot,1)==2 && isequal(max_alphadot_idx(1,1),...
        max_alphadot_idx(2,1))
    % Old Method...: FILTER alpha_dot BEFORE IT CAN BE USED -
    % temp use of alpha_dot2 since it will give the same result
    [max_alphadot,max_alphadot_idx] = findpeaks(...
        -Target.alpha_dot2(1:(end-5),i));
end

% Polyfit to smooth data
if size(max_alphadot,1)>1
    ref = round(ref/2);
else
    ref = round(ref/2);
end
idx11 = max_alphadot_idx(1,1) - ref;
if idx11 < 1
    idx11 = 1;
end
idx12 = max_alphadot_idx(1,1) + ref;
if idx12 > size(SatDat.time,1)
```

```
    idx12 = size(SatDat.time,1);
end
if stdev > 0
    poly_deg = 5;
    % Re-fit curve to alpha in this smaller segment -- need to
    % estimate alphadot better here
    ws = warning('off','all');  % Turn off warning
    p = polyfit(SatDat.time(idx11:idx12,1),...
        Target.alpha_meas(idx11:idx12,i),poly_deg);
    Target.alpha(idx11:idx12,i) = polyval(...
        p,SatDat.time(idx11:idx12,1));
    warning(ws)  % Turn it back on.
    % Recalculate alphadot in this smaller segment
    for j=idx11:idx12
        % alpha rate from five-point central difference formula
        if j>(idx11+1) && j<(idx12-1)
            Target.alpha_dot(j,i) = (Target.alpha(j-2,i) - ...
                8*Target.alpha(j-1,i) + 8*Target.alpha(j+1,i) - ...
                Target.alpha(j+2,i))/(12*dt);
            if abs(Target.alpha_dot(j,i)) > ...
                    1.1*abs(Target.alpha_dot(j-1,i))
                Target.alpha_dot(j,i) = Target.alpha_dot(j-1,i);
            end
        end
        if j<(idx11+2) % forward difference formula
            Target.alpha_dot(j,i) = (-11*Target.alpha(j,i) + ...
                18*Target.alpha(j+1,i) - 9*Target.alpha(j+2,i)...
                + 2*Target.alpha(j+3,i))/(6*dt);
        end
        % backward difference formula
        % normally -2, but using -5 to catch some typical errors
        if j>(idx12-5)
            Target.alpha_dot(j,i) = (11*Target.alpha(j,i) - ...
                18*Target.alpha(j-1,i) + 9*Target.alpha(j-2,i)...
                - 2*Target.alpha(j-3,i))/(6*dt);
            if abs(Target.alpha_dot(j,i)) > ...
                    1.1*abs(Target.alpha_dot(j-1,i))
                Target.alpha_dot(j,i) = Target.alpha_dot(j-1,i);
            end
        end
    end
end
ws = warning('off','all');  % Turn off warning
p1 = polyfit(SatDat.time(idx11:idx12,1),...
    Target.alpha_dot(idx11:idx12,i),4);
warning(ws)  % Turn it back on.
Target.alpha_dot(idx11:idx12,i) = polyval(...
    p1,SatDat.time(idx11:idx12,1));

% Find indices where alpha rate is maximum in the smoothed data
[max_alphadot1,max_alphadot_idx1] = findpeaks(...
    -Target.alpha_dot(idx11:idx12,i));

% For highly skewed NMC, there is only one max_alphadot!
if length(max_alphadot_idx)>1
    idx21 = max_alphadot_idx(2,1) - ref;
```

```
if idx21 < 1
    idx21 = 1;
end
idx22 = max_alphadot_idx(2,1) + ref;
if idx22 > size(SatDat.time,1)
    idx22 = size(SatDat.time,1);
end
if stdev > 0
    poly_deg = 5;
    % Re-fit curve to alpha in this smaller segment -- need to
    % estimate alphadot better here
    ws = warning('off','all');  % Turn off warning
    p = polyfit(SatDat.time(idx21:idx22,1),...
        Target.alpha_meas(idx21:idx22,i),poly_deg);
    Target.alpha(idx21:idx22,i) = polyval(...
        p,SatDat.time(idx21:idx22,1));
    warning(ws)  % Turn it back on.
    % Recalculate alphadot in this smaller segment
    for j=idx21:idx22
        % alpha rate from five-point central difference formula
        if j>(idx21+1) && j<(idx22-1)
            Target.alpha_dot(j,i) = (Target.alpha(j-2,i) - ...
                8*Target.alpha(j-1,i) + 8*Target.alpha(j+1,i)...
                - Target.alpha(j+2,i))/(12*dt);
            if abs(Target.alpha_dot(j,i)) > ...
                    1.1*abs(Target.alpha_dot(j-1,i))
                Target.alpha_dot(j,i) = Target.alpha_dot(j-1,i);
            end
        end
        if j<(idx21+2) % forward difference formula
            Target.alpha_dot(j,i) = (-11*Target.alpha(j,i) +...
                18*Target.alpha(j+1,i) -9*Target.alpha(j+2,i)...
                + 2*Target.alpha(j+3,i))/(6*dt);
        end
        % backward difference formula
        % normally -2, but using -5 to catch some typical errors
        if j>(idx22-5)
            Target.alpha_dot(j,i) = (11*Target.alpha(j,i) -...
                18*Target.alpha(j-1,i) +9*Target.alpha(j-2,i)...
                - 2*Target.alpha(j-3,i))/(6*dt);
            if abs(Target.alpha_dot(j,i)) > ...
                    1.1*abs(Target.alpha_dot(j-1,i))
                Target.alpha_dot(j,i) = Target.alpha_dot(j-1,i);
            end
        end
    end
end
ws = warning('off','all');  % Turn off warning
p2 = polyfit(SatDat.time(idx21:idx22,1),...
    Target.alpha_dot(idx21:idx22,i),4);
warning(ws)  % Turn it back on.
Target.alpha_dot(idx21:idx22,i) = polyval(...
    p2,SatDat.time(idx21:idx22,1));
[max_alphadot2,max_alphadot_idx2] = findpeaks(...
    -Target.alpha_dot(idx21:idx22,i));
Target.max_alphadot(:,i) = ...
```

```
            [-max_alphadot1;-max_alphadot2]*180/pi*3600; % asecs/sec
        Target.max_alphadot_idx(:,i) = ...
            [idx11+max_alphadot_idx1-1;idx21+max_alphadot_idx2-1];

        Target.alpha(:,i) = wrap(Target.alpha(:,i));

        Target.alpha_at_max_alphadot(1,i) = Target.alpha(...
            Target.max_alphadot_idx(1,i),i)*180/pi; % deg
        Target.alpha_at_max_alphadot(2,i) = Target.alpha(...
            Target.max_alphadot_idx(2,i),i)*180/pi; % deg
        Target.range_at_max_alphadot(1,i) = ...
            Target.rho_true(Target.max_alphadot_idx(1,i),i);
        Target.range_at_max_alphadot(2,i) = ...
            Target.rho_true(Target.max_alphadot_idx(2,i),i);
    else
        Target.max_alphadot(1,i) = max_alphadot1*180/pi*3600; % asecs/sec
        Target.max_alphadot_idx(1,i) = idx11+max_alphadot_idx1-1;

        Target.alpha(:,i) = wrap(Target.alpha(:,i));

        Target.alpha_at_max_alphadot(1,i) = ...
            Target.alpha(Target.max_alphadot_idx(1,i),i)*180/pi; % deg
        Target.range_at_max_alphadot(1,i) = ...
            Target.rho_true(Target.max_alphadot_idx(1,i),i);
    end

    % Is target above or below surveyor at surveyor's perigee?
    [min_true_anom,min_true_anom_idx] = min(SatDat.true_anom(:,1));
    if abs(Target.alpha(min_true_anom_idx,i))>pi/2
        Target.target_at_perigee(1,i) = -1; % -1 for below
        if Target.Xrel(min_true_anom_idx,i)>0 % catch flaws in design
            keyboard
        end
    else
        Target.target_at_perigee(1,i) = 1; % +1 for above
    end
    Target.alpha_meas = wrap(Target.alpha_meas);
    Target.beta_meas = wrap(Target.beta_meas);
end
if rate_calcs == 0
    Target.alpha(:,i) = wrap(Target.alpha(:,i));
    Target.beta(:,i) = wrap(Target.beta(:,i));
    Target.alpha_meas = wrap(Target.alpha_meas);
    Target.beta_meas = wrap(Target.beta_meas);
end
%% ECLIPSE RANGING
% only if there is an eclipse
if SatDat.eclipse_entry_umb_start(1,1) ~= 0
    Target.eclipse_delta_start(1,i) = ...
        (SatDat.eclipse_entry_umb_start(1,1) - ...
        SatDat.eclipse_entry_umb_start(1,i+1))*dt;
    Target.eclipse_target_idx(1,i) = ...
        SatDat.eclipse_entry_umb_start(1,i+1);
    Surveyor.eclipse_surveyor_idx = ...
        SatDat.eclipse_entry_umb_start(1,1);
    Target.eclipse_range_true(1,i) = ...
```

```
            Target.rho_true(Surveyor.eclipse_surveyor_idx,i);
    end
end
```

E.5 Skewness Factor Determination

```
%  [function]    skewFactor
%
%  [purpose]:    Function determines the skewness of the line-of-sight (LOS)
%                profile from the restricted problem of natural motion
%                circumnavigation (NMC).
%
%  [useage]:     [d,delta_split] = skewFactor(alpha,Yrel,nu)
%
%  [inputs]:     alpha     = Matrix (m x n) containing the angle alpha in
%                            the rs-plane for the ith of n satellites.
%                            This matrix will generally come from
%                            Target.alpha.
%
%                Yrel      = Matrix (m x n) containing the relative position
%                            along the s-axis for the ith of n satellites.
%                            This matrix will generally come from
%                            Target.Yrel.
%
%  [outputs]:    d         = Array (1 x n) containing the skewness factor
%                            for each of n satellites.
%
%  [refs]:
%
%  [history]:    27 Oct 2016 - Max Yates

function [d] = skewFactor(alpha,Yrel)

m = size(alpha,1); % observations
n = size(alpha,2); % satellites

for i=1:n
    % The sine of alpha can be searched for zero crossings. No zero crossings
    % indicates the relative trajectory does not enclose the Surveyor.
    signum = sign(sin(alpha(:,i))); % get sign of data
    signum(sin(alpha(:,i))==0) = 1; % set sign of exact data zeros to positiv
    zero_crossings = find(diff(signum)~=0); % get zero crossings by diff ~= 0
    if zero_crossings > 0 % only enclosed relative orbits
        % Find where alpha is +/- 90 deg
        % Index of closest entry to +s-axis
        [posS_value, alpha90_idx1] = min(abs(alpha(:,i)-pi/2));
        % Index of closest entry to -s-axis
        [negS_value, alpha90_idx2] = min(abs(alpha(:,i)+pi/2));
        if posS_value > 1 || negS_value > 1
            disp(['Target ',num2str(i), ...
                ' is not circumnavigating the Surveyor'])
            d(1,i) = 0;
            continue
        end

        % Calculate arc length at +/- s-axis crossing
        num_obs = 5; % how many data points to use in calculating arc length
        % Want alpha90_idx1 on -s-axis and alpha90_idx2 on +s-axis
        if alpha(alpha90_idx1,i)>0
```

```
            temp = alpha90_idx1;
            alpha90_idx1 = alpha90_idx2;
            alpha90_idx2 = temp;
        end
        % End-point protection
        if alpha90_idx1-num_obs < 0
            arc_negS = abs(alpha(1,i)-alpha(1+2*num_obs,i));
        elseif alpha90_idx1+num_obs > m
            arc_negS = abs(alpha(m-2*num_obs,i)-alpha(m,i));
        else
            arc_negS = abs(alpha(alpha90_idx1-num_obs,i) - ...
                alpha(alpha90_idx1+num_obs,i));
        end
        if alpha90_idx2-num_obs < 0
            arc_posS = abs(alpha(1,i)-alpha(1+2*num_obs,i));
        elseif alpha90_idx2+num_obs > m
            arc_posS = abs(alpha(m-2*num_obs,i)-alpha(m,i));
        else
            arc_posS = abs(alpha(alpha90_idx2-num_obs,i) - ...
                alpha(alpha90_idx2+num_obs,i));
        end

        % Calculate distance ratio d on unit 2x1 ellipse
        d(1,i) = 2*tan(arc_posS/2)/(tan(arc_negS/2)+tan(arc_posS/2));
    else
        disp(['Target ',num2str(i),' is not circumnavigating the Surveyor'])
        d(1,i) = 0;
    end
end
```

E.6 Admissible Region Hypotheses

```
%  [function]   admRegHyp
%
%  [purpose]:   Function determines admissible hypotheses for the
%               restricted problem of natural motion circumnavigation.
%
%  [useage]:    [coes,d_hyp,delta_split_hyp,rel_apsis_hyp,alpha_disp_true]
%                   = admRegHyp(SatDat,input,Surveyor,alpha,beta,d,...
%                   ranges,bias,stdev)
%
%  [inputs]:    SatDat    = Structure containing the GMAT extracted data.
%                           The exact content is configured in the
%                           gmatOrbitProp code, but presently includes
%                           for every ith satellite entries of the form:
%                           SatDat.X(:,i).
%                           NOTE: The 1st satellite is the Surveyor! All
%                           others are Target satellites.
%
%               Surveyor  = Structure containing the LOS processed data for
%                           the Surveyor.
%
%               input     = Input structure for the Surveyor and Target
%                           which includes the true elements and epoch of
%                           initial conditions.
%
%               alpha     = Matrix (m x n) containing the angle alpha in
%                           the rs-plane for the ith of n satellites.
%                           This matrix will generally come from
%                           Target.alpha.
%
%               beta      = Matrix (m x n) containing the OOP angle beta
%                           for the ith of n satellites. This matrix will
%                           generally come from Target.beta.
%
%               d         = Array (1 x n) containing the skewness factor
%                           for each of n satellites.
%
%               ranges    = Array (p x 1) containing the range hypotheses
%                           along the LOS vector.
%
%               bias      = angle measurement bias [arc sec]
%
%               stdev     = 1-sigma angle measurement uncertainty [arc sec]
%
%  [outputs]:   coes      = Matrix (np x 6) containing initial conditions
%                           for np satellite hypotheses.
%                           where
%                              coes(i,1) = semi-major axis [km]
%                              coes(i,2) = eccentricity
%                              coes(i,3) = inclination [deg]
%                              coes(i,4) = RAAN [deg]
%                              coes(i,5) = argument of perigee [deg]
%                              coes(i,6) = true anomaly [deg]
%
```

```
%  [refs]:
%
%  [history]:   27 Oct 2016 - Max Yates

function [coes,d_hyp,delta_split_hyp,rel_apsis_hyp,alpha_disp_true,u_hyp]...
    = admRegHyp(SatDat,input,Surveyor,alpha,beta,d,ranges,bias,stdev)

global J2 MU DU geo_degree use_skew2nu search_argp

codeDir = pwd;
format long g

m = size(SatDat.X,1); % Observations
n = size(SatDat.X,2)-1; % Target satellites
p = size(ranges,1); % Number of hypotheses per satellite

% PREALLOCATION
coes = zeros(n*p,6);
d_hyp = zeros(1,n*p);
delta_split_hyp = zeros(1,n*p);
rel_apsis_hyp = zeros(1,n*p);
Xrel_inertial_unit = zeros(m,n);
Yrel_inertial_unit = zeros(m,n);
Zrel_inertial_unit = zeros(m,n);

% Speed up skewTestBlock by propagating Surveyor at lower sample rate ONCE
% rather than repeatedly
options = [0,0,0,0,0]; % [prop_time,dt,torus,output,orbitview]
SatDat_s = gmatOrbitProp(input.state_type,input.coord_sys,...
    input.elements(1,:),input.epoch,'UTCGregorian',options);

index = 1; % Column index from 1:np
for i=1:n
    if d(1,i) ~= 0 % Restricted to NMC orbits
        % Form LOS vector in inertial frame
        Xrel_inertial = SatDat.X(:,i+1)-SatDat.X(:,1); % Target - Surveyor
        Yrel_inertial = SatDat.Y(:,i+1)-SatDat.Y(:,1);
        Zrel_inertial = SatDat.Z(:,i+1)-SatDat.Z(:,1);
        rho = sqrt(Xrel_inertial.^2 + Yrel_inertial.^2 + Zrel_inertial.^2);
        Xrel_inertial_unit(:,i) = Xrel_inertial./rho;
        Yrel_inertial_unit(:,i) = Yrel_inertial./rho;
        Zrel_inertial_unit(:,i) = Zrel_inertial./rho;

        error_ratio = zeros(p,1);

        %% Determine if we have a negative or positive disparity.
        % Propagate Surveyor and Target satellites for one orbital
        % period. No bonus time at the end (hence -0.02 in options).
        options = [-0.02,0,0,0,0]; % [prop_time,dt,torus,output,orbitview]
        SatDat_aux0 = gmatOrbitProp('Keplerian',input.coord_sys,...
            input.coes,input.epoch,'UTCGregorian',options);
        % generate LOS observations
        [Surveyor_aux0,Target_aux0] = losProc(SatDat_aux0,input,...
            [bias,stdev,1]);

        % The sine of alpha can be searched for zero crossings. One
        % zero crossing indicates positive disparity, whereas two zero
        % crossing indicates either positive or negative disparity. No
        % zero crossings indicates the relative trajectory does not
        % enclose the Surveyor.
        signum = sign(sin(Target_aux0.alpha(:,i))); % get sign of data
        % Set sign of exact data zeros to positive
        signum(sin(Target_aux0.alpha(:,i))==0) = 1;
        % Get zero crossings by diff ~= 0
        zero_crossings = find(diff(signum)~=0);
        % POSITIVE DISPARITY (alpha profile is >360 deg in one period)
        if length(zero_crossings) > 1 && sign(Target_aux0.alpha(1,1))==-1....
                && sign(Target_aux0.alpha(end,1))==1
            alpha_disp_true = (2*pi-abs(Target_aux0.alpha(1,1))-...
                abs(Target_aux0.alpha(end,1)))*180/pi;
        elseif length(zero_crossings) > 1 && ...
                sign(Target_aux0.alpha(1,1))==1 && ...
                sign(Target_aux0.alpha(end,1))==1 && ...
                abs(Target_aux0.alpha(1,1)) > abs(Target_aux0.alpha(end,1))
            alpha_disp_true = (Target_aux0.alpha(1,1)-...
                Target_aux0.alpha(end,1))*180/pi;
        elseif length(zero_crossings) > 1 && ...
                sign(Target_aux0.alpha(1,1))==-1 && ...
                sign(Target_aux0.alpha(end,1))==-1 && ...
                abs(Target_aux0.alpha(end,1)) > abs(Target_aux0.alpha(1,1))
            alpha_disp_true = (Target_aux0.alpha(end,1)-...
                Target_aux0.alpha(1,1))*180/pi;
        % NEGATIVE DISPARITY (alpha profile is <360 deg in one period)
        elseif length(zero_crossings) == 1 && ...
                sign(Target_aux0.alpha(1,1))==1 && ...
                sign(Target_aux0.alpha(end,1))==-1
            alpha_disp_true = -(2*pi-abs(Target_aux0.alpha(1,1))-...
                abs(Target_aux0.alpha(end,1)))*180/pi;
        elseif length(zero_crossings) > 1 && ...
                sign(Target_aux0.alpha(1,1))==1 && ...
                sign(Target_aux0.alpha(end,1))==1 && ...
                abs(Target_aux0.alpha(end,1)) > abs(Target_aux0.alpha(1,1))
            alpha_disp_true = (Target_aux0.alpha(1,1)-...
                Target_aux0.alpha(end,1))*180/pi;
        elseif length(zero_crossings) > 1 && ...
                sign(Target_aux0.alpha(1,1))==-1 && ...
                sign(Target_aux0.alpha(end,1))==-1 && ...
                abs(Target_aux0.alpha(1,1)) > abs(Target_aux0.alpha(end,1))
            alpha_disp_true = (Target_aux0.alpha(1,1)-...
                Target_aux0.alpha(end,1))*180/pi;
        end

        disparity_threshold = 2.5; % [deg]
        % Can use 2x1 geometry as approximation when below this value.
        % Explore this threshold more! Somewhat arbitrary for now.

        %% Find if Target is above or below Surveyor at initialization
        if abs(alpha(1,1)) > pi/2
            above_below = -1; % below
        else
            above_below = 1; % above
        end
```

```
%% Should rd1 point to apsis above or below Surveyor & should rd2
%% point left or right?
% If we start above, then we want the full half-period between
% s-axis crossings below the Surveyor and vice-versa.
if above_below == 1
    use_above_below = -1; % 1 for above, -1 for below
elseif above_below == -1
    use_above_below = 1; % 1 for above, -1 for below
end
% Previously, I found that the osculation corrections were better
% when rd1_index is greater than rd2_index. Writing with constraint
% that rd1_index > rd2_index (the only way I've been able to
% consistently solve the osculation problem).
if above_below == -1 && use_above_below == 1 % rd2 on +s side
    use_left_right = 1; % 1 for +s side, -1 for -s side
elseif above_below == 1 && use_above_below == -1 % rd2 on -s side
    use_left_right = -1; % 1 for +s side, -1 for -s side
end

%% COE Hypothesis Loop
for j=1:p
    % Check for past saved runs -- solves the crashing problem
    if exist(fullfile(strcat(codeDir,'\Cases\',sprintf('%s',...
            input.runCase)), sprintf('hyp%i.mat',index)), 'file')
        load(strcat(codeDir,'\Cases\',sprintf('%s',...
            input.runCase),'\',sprintf('hyp%i.mat',index)))
        continue
    end

    % Relative apses. Use midway point between +/- s-axis crossings
    % Index of closest entry to +s-axis
    [posS_value, posS_index] = min(abs(alpha(:,i)-pi/2));
    % Index of closest entry to -s-axis
    [negS_value, negS_index] = min(abs(alpha(:,i)+pi/2));
    rel_apsis_hyp(1,index) = round((posS_index + negS_index)/2);
    apsis_index = rel_apsis_hyp(1,index);

    % Form Target's inertial position hypothesis at t(apsis_index)
    X_hyp = SatDat.X(apsis_index,1) + ...
        ranges(j,1)*Xrel_inertial_unit(apsis_index,i);
    Y_hyp = SatDat.Y(apsis_index,1) + ...
        ranges(j,1)*Yrel_inertial_unit(apsis_index,i);
    Z_hyp = SatDat.Z(apsis_index,1) + ...
        ranges(j,1)*Zrel_inertial_unit(apsis_index,i);
    r_hyp = sqrt(X_hyp.^2 + Y_hyp.^2 + Z_hyp.^2);
    % Check accuracy
    r_truth = sqrt(SatDat.X(apsis_index,2)^2+...
        SatDat.Y(apsis_index,2)^2+SatDat.Z(apsis_index,2)^2);
    r_error = r_hyp - r_truth;
    range_error = ranges(j,1) - rho(apsis_index,1)

    % Flag to specify if manually iterating; manually add hyp.coes
    % from first iteration below
    iterate_run = 0;
    if iterate_run == 0
        % Initially use Surveyor's semi-major axis
        a_hyp = SatDat.a(1,1);
        i_hyp = SatDat.i(1,1);
        RAAN_hyp = SatDat.RAAN(1,1);
        % Approximate Target speed -- 2Body (non-osculating)
        v_hyp = sqrt(2*MU/r_hyp-MU/a_hyp);
        % Check accuracy
        v_hyp_truth = sqrt(SatDat.VX(apsis_index,2)^2 + ...
            SatDat.VY(apsis_index,2)^2 + SatDat.VZ(apsis_index,2)^2);
        % Approximate Target eccentricity -- 2Body (non-osculating)
        e_hyp = sqrt(1-(2*(v_hyp^2)*r_hyp)/MU+...
            ((v_hyp^4)*(r_hyp^2))/(MU^2));
        % Check accuracy
        e_hyp_truth = SatDat.e(apsis_index,2);
    else
        % manually paste hypotheses from first iteration...
        % not too much of a pain for now since it is rarely needed
        hyp.coes = [];
        a_hyp = hyp.coes(j,1);
        e_hyp = hyp.coes(j,2);
        i_hyp = hyp.coes(j,3);
        RAAN_hyp = hyp.coes(j,4);
    end

    for smai = 1:1 % time consuming -- typically just 1
        % Approximate Target eccentricity -- 2Body (non-osculating)
        v_hyp = sqrt(2*MU/r_hyp-MU/a_hyp);
        e_hyp0 = sqrt(1-(2*(v_hyp^2)*r_hyp)/MU+...
            ((v_hyp^4)*(r_hyp^2))/(MU^2));

        %% Approximate Argument of Perigee
        % The Target must have approximately the same argument of
        % latitude as the Surveyor in order to sustain close
        % proximity. Search argp until alpha0 hypothesis
        % approximately matches the relative trajectory.
        if j == 1
            disp(['Approximating argument of perigee'])
            % Approximate Target argument of latitude at apsis using
            % Surveyor's RAAN
            Node = [cosd(SatDat.RAAN(1,1));sind(SatDat.RAAN(1,1));0];
            r1 = [SatDat.X(apsis_index,1);SatDat.Y(apsis_index,1);...
                SatDat.Z(apsis_index,1)] + ranges(j,1)*...
                Xrel_inertial_unit(apsis_index,i); ...
                Yrel_inertial_unit(apsis_index,i); ...
                Zrel_inertial_unit(apsis_index,i)];
            u_hyp_apsis = acosd(dot(node,r1)/(mag(node)*mag(r1)));
            % Quadrant check for u_hyp
            if r1(3,1)<0
                u_hyp_apsis = 360 - u_hyp_apsis;
            end

            darglat_s = (SatDat.arg_per(apsis_index,1) + ...
                SatDat.true_anom(apsis_index,1)) - ...
                (SatDat.arg_per(1,1)+SatDat.true_anom(1,1));
            % Check accuracy:
            darglat_t = (SatDat.arg_per(apsis_index,2) + ...
            SatDat.true_anom(apsis_index,2)) - ...
            (SatDat.arg_per(1,2)+SatDat.true_anom(1,2));

        % Approximate Target argument of latitude at epoch
        u_hyp = u_hyp_apsis - darglat_s;
        if u_hyp < 0
            u_hyp = u_hyp + 360;
        end
        % Check accuracy:
        u_hyp_truth = SatDat.arg_per(1,2)+SatDat.true_anom(1,2);

        % Search Method - Coarse 10 deg precision on first
        % iteration since none of the other COEs have been
        % determined yet.
        if search_argp == 1
            dargp=(0:10:350)';
            argpTestBlock;

            % Propagate Surveyor with argp_hyp for osculation
            % corrections
            u_hyp_s = SatDat.arg_per(1,1)+SatDat.true_anom(1,1);
            % sampling freq. for returned data rounded to nearest
            % hundredth [sec]
            dt = roundn(SatDat.time(2,1)-SatDat.time(1,1),-2);
            % [prop_time,dt,torus,output,orbitview]
            options = [-0.02,dt,0,1,0];
            SatDat2 = gmatOrbitProp(input.state_type, ...
                input.coord_sys,[input.coes(1,1), ...
                input.coes(1,2),input.coes(1,3), ...
                input.coes(1,4),argp_hyp,u_hyp_s-argp_hyp], ...
                input.epoch,'UTCGregorian',options);
        elseif search_argp == 0
            % Use argument of perigee from Surveyor at epoch
            argp_hyp = SatDat.arg_per(1,1); % [deg]
        end

        % Check accuracy of method
        argp_hyp
        argp_error = argp_hyp - SatDat.arg_per(1,i+1)
    end

    % Correct epoch eccentricity for osculations
    if abs(cosd(argp_hyp)-cosd(input.coes(1,5)))<1e-6
        e_hyp = e_hyp0-(SatDat.e(apsis_index,1)-SatDat.e(1,1));
    else
        % Use eccentricity curve from Surveyor with argp_hyp
        e_hyp = e_hyp0-(SatDat2.e(apsis_index,1)-SatDat2.e(1,1));
    end

    %% Approximate Semi-Major Axis
    if abs(alpha_disp_true) > disparity_threshold
        disp(['Approximating semi-major axis'])
        if alpha_disp_true > 0 % positive disparity
            da=(0:-1:-20)';
            pos_neg = -1;
        else
```

```
            da=(0:1:20)';
            pos_neg = 1;
        end
        smaTestBlock;
        if pos_neg == 1
            da=(da(k-1,1):0.1:da(k,1))';
        elseif pos_neg == -1
            da=(da(k-1,1):-0.1:da(k,1))';
        end
        smaTestBlock;
        if pos_neg == 1
            da=(da(k-1,1):0.01:da(k,1))';
        elseif pos_neg == -1
            da=(da(k-1,1):-0.01:da(k,1))';
        end
        smaTestBlock;

        % split the difference for da_hyp
        coeffs = polyfit([disparities(1,k-1), ...
            disparities(1,k)],[da(k-1,1),da(k,1)],1);
        da_hyp = polyval(coeffs,alpha_disp_true)
        nu_hyp0(j,:) = nu_hyp; % save to look at during debug
        % split the difference for nu_hyp and d_hyp
        coeffs = polyfit([disparities(1,k-1), ...
            disparities(1,k)],[nu_hyp(1,k-1),nu_hyp(1,k)],1);
        nu_hyp = polyval(coeffs,alpha_disp_true);
        coeffs = polyfit([disparities(1,k-1), ...
            disparities(1,k)],[d_aux(1,k-1),d_aux(1,k)],1);
        d_hyp(1,index) = polyval(coeffs,alpha_disp_true);

        a_hyp = a_hyp + da_hyp;

        % Update Target eccentricity
        v_hyp = sqrt(2*MU/r_hyp-MU/a_hyp);
        e_hyp0 = sqrt(1-(2*(v_hyp^2)*r_hyp)/MU+...
            ((v_hyp^4)*(r_hyp^2))/(MU^2));
        % Correct epoch eccentricity for osculations
        if abs(cosd(argp_hyp)-cosd(input.coes(1,5)))<1e-6
            e_hyp = e_hyp0 - (SatDat.e(apsis_index,1)-...
                SatDat.e(1,1));
        else
            % Use eccentricity curve from Surveyor with argp_hyp
            e_hyp = e_hyp0 - (SatDat2.e(apsis_index,1)-...
                SatDat2.e(1,1));
        end
    end
end

% Check accuracy of method
e_error = e_hyp - SatDat.e(1,i+1)

%% Calculate Unit Orbit Angular Momentum
if use_left_right == 1
    ref = pi/2; % Reference angle for finding index for D
elseif use_left_right == -1
    ref = -pi/2; % Reference angle for finding index for D
```

```
end

% Can use 2x1 geometry as approximation
if abs(alpha_disp_true) < disparity_threshold % [deg]
    % Robust for all cases: d>1, d<1, above or below
    theta3 = atan(abs((d(1,i)-1))/0.5);
    scaled_range_hyp = 0.5/cos(theta3);
    % Range hypothesis projected into rs-plane
    projected_range = ranges(j,1)*cos(beta(apsis_index));
    if use_left_right == -1
        D = projected_range*(d(1,i)/scaled_range_hyp);
    elseif use_left_right == 1
        D = projected_range*((2-d(1,i))/scaled_range_hyp);
    end
    rd1_index = apsis_index;
    % Linear index of closest entry
    [value, rd2_index] = min(abs(alpha(:)-ref));
    rd1 = [SatDat.X(rd1_index,1);SatDat.Y(rd1_index,1);...
        SatDat.Z(rd1_index,1)] + ranges(j,1)*...
        [Xrel_inertial_unit(rd1_index,i);...
        Yrel_inertial_unit(rd1_index,i);...
        Zrel_inertial_unit(rd1_index,i)];
    rd2 = [SatDat.X(rd2_index,1);SatDat.Y(rd2_index,1);...
        SatDat.Z(rd2_index,1)] + (D/cos(beta(rd2_index))).*...
        [Xrel_inertial_unit(rd2_index,i);...
        Yrel_inertial_unit(rd2_index,i);...
        Zrel_inertial_unit(rd2_index,i)];
end

% Use scaled auxiliary trajectory geometry as default
if abs(alpha_disp_true) < disparity_threshold
    [nu_hyp,d_hyp,~] = skew2nu(SatDat_s,a_hyp,e_hyp,...
        input.coes(1,3),input.coes(1,4),argp_hyp,u_hyp,...
        input,d(1,i),[0,0,1]);
else
    % nu_hyp comes from sma_approx
end
% Propagate auxiliary trajectory
% Sampling freq for returned data rounded to nearest hundredth
dt = roundn(SatDat.time(2,1)-SatDat.time(1,1),-2); % [sec]
options = [0,dt,0,1,0]; % [prop_time,dt,torus,output,orbitview]
SatDat_aux = gmatOrbitProp('Keplerian',input.coord_sys,...
    [input.coes(1,:);a_hyp,e_hyp,input.coes(1,3),...
    input.coes(1,4),argp_hyp,nu_hyp],input.epoch,...
    'UTCGregorian',options);
% Generate LOS observations
[Surveyor_aux,Target_aux] = losProc(SatDat_aux,input,[0,0,1]);

rd1_index = apsis_index;
% Linear index of closest entry
[value, rd2_index] = min(abs(alpha(:)-ref));
rd1 = [SatDat.X(rd1_index,1);SatDat.Y(rd1_index,1);...
    SatDat.Z(rd1_index,1)]+Target_aux.rho_true(rd1_index,1)/...
    cos(beta(rd1_index,1))*[Xrel_inertial_unit(rd1_index,i);...
    Yrel_inertial_unit(rd1_index,i);...
    Zrel_inertial_unit(rd1_index,i)];
```

```
rd2 = [SatDat.X(rd2_index,1);SatDat.Y(rd2_index,1);...
    SatDat.Z(rd2_index,1)]+(Target_aux.rho_true(rd2_index,1)/...
    cos(beta(rd2_index))).*[Xrel_inertial_unit(rd2_index,i);...
    Yrel_inertial_unit(rd2_index,i);...
    Zrel_inertial_unit(rd2_index,i)];

if (rd2_index > rd1_index) && (abs(alpha(rd1_index))+...
        abs(alpha(rd2_index)) < pi)
    H_unit = unit(cross(rd1,rd2));
elseif (rd1_index > rd2_index) && (abs(alpha(rd1_index))+...
        abs(alpha(rd2_index)) < pi)
    H_unit = unit(cross(rd2,rd1));
elseif (rd2_index > rd1_index) && (abs(alpha(rd1_index))+...
        abs(alpha(rd2_index)) > pi)
    H_unit = unit(cross(rd2,rd1));
elseif (rd1_index > rd2_index) && (abs(alpha(rd1_index))+...
        abs(alpha(rd2_index)) > pi)
    H_unit = unit(cross(rd1,rd2));
End

%% Calculate Inclination
i_hyp = acos(H_unit(3))*180/pi;
% Correct inclination for osculation with Surveyor data
i_hyp1 = i_hyp-((SatDat2.i(rd1_index,1)-SatDat2.i(1,1))+...
     (SatDat2.i(rd2_index,1)-SatDat2.i(1,1)))/2;
i_hyp2 = i_hyp-(SatDat2.i(round((rd1_index+rd2_index)/2),1)-...
    SatDat2.i(1,1));
i_hyp = (i_hyp1 + i_hyp2)/2;
% Check accuracy of method
i_error = i_hyp - SatDat.i(1,i+1)

%% Calculate RAAN
ncap_ish = cross([0,0,1],H_unit);
ncap = unit(ncap_ish);
RAAN_hyp = acos(ncap(1))*180/pi;
% Quadrant check for RAAN_hyp
if ncap(2)<0
    RAAN_hyp = 360 - RAAN_hyp;
end
% Correct RAAN for osculation & regression with Surveyor data
RAAN_hyp1 = RAAN_hyp-((SatDat2.RAAN(rd1_index,1)-...
    SatDat2.RAAN(1,1))+(SatDat2.RAAN(rd2_index,1)-...
    SatDat2.RAAN(1,1)))/2;
RAAN_hyp2 = RAAN_hyp-(SatDat2.RAAN(round((rd1_index+...
    rd2_index)/2),1)-SatDat2.RAAN(1,1));
RAAN_hyp = (RAAN_hyp1 + RAAN_hyp2)/2;
% Check accuracy of method
RAAN_error = RAAN_hyp - SatDat.RAAN(1,i+1)

%% Approximate Target True Anomaly
[nu_hyp,d_hyp(1,index),delta_split_hyp(1,index)] = ...
    skew2nu(SatDat_s,a_hyp,e_hyp,i_hyp,RAAN_hyp,...
    argp_hyp,u_hyp,input,d(1,i),[bias,stdev,1]);

%% Hypothesis COEs
coes(index,:) = [a_hyp,e_hyp,i_hyp,RAAN_hyp,argp_hyp,nu_hyp]
```

```
            %% Sanity Check
            % Propagate Target estimate
            % Sampling freq for returned data rounded to nearest hundredth
            dt = roundn(SatDat.time(2,1)-SatDat.time(1,1),-2); % [sec]
            options = [0,dt,0,0,0]; % [prop_time,dt,torus,output,orbitview]
            SatDat_hyp = gmatOrbitProp('Keplerian',input.coord_sys,...
                [input.coes(1,:);coes(index,:)],input.epoch,...
                'UTCGregorian',options);
            if abs(alpha_disp_true) < disparity_threshold
                Xrel_inertial_check = SatDat_hyp.X(apsis_index,i+1)-...
                    SatDat_hyp.X(apsis_index,1); % Target - Surveyor
                Yrel_inertial_check = SatDat_hyp.Y(apsis_index,i+1)-...
                    SatDat_hyp.Y(apsis_index,1);
                Zrel_inertial_check = SatDat_hyp.Z(apsis_index,i+1)-...
                    SatDat_hyp.Z(apsis_index,1);
                rho_check = sqrt(Xrel_inertial_check.^2 + ...
                    Yrel_inertial_check.^2 + Zrel_inertial_check.^2);
                % Check accuracy
                error_ratio(j,1) = abs(rho_check - ranges(j,1))/ranges(j,1)
            else
                Xrel_inertial_check = SatDat_hyp.X(apsis_index,i+1)-...
                    SatDat_hyp.X(apsis_index,1); % Target - Surveyor
                Yrel_inertial_check = SatDat_hyp.Y(apsis_index,i+1)-...
                    SatDat_hyp.Y(apsis_index,1);
                Zrel_inertial_check = SatDat_hyp.Z(apsis_index,i+1)-...
                    SatDat_hyp.Z(apsis_index,1);
                rho_check = sqrt(Xrel_inertial_check.^2 + ...
                    Yrel_inertial_check.^2 + Zrel_inertial_check.^2);
                % Check accuracy
                error_ratio(j,1) = abs(rho_check - ranges(j,1))/ranges(j,1)
            end
            if error_ratio(j,1)>0.1 % liberal for now
                %coes(index,:) = [];
                %continue;
            end
            index = index+1;
            save(strcat(codeDir,'\Cases\',sprintf('%s',input.runCase),...
                '\',sprintf('hyp%i.mat',index-1)))
        end
    end
end
end

function F = getTrueAnom(nu_hyp,MU,rp,e_hyp,v_hyp)
    % Wiesel pg 70, Eqn 2.78
    % Rewrite the equation in the form F(x) = 0
    F = sqrt(MU/rp*(sind(nu_hyp)^2) + MU/rp*(e_hyp+cosd(nu_hyp))^2) - v_hyp;
end
```

E.7 True Anomaly Search with Skewness Factor Heuristic

```
%  [function]    skew2nu
%
%  [purpose]:    Function optimally searches for the true anomaly of a Target
%                hypothesis so as to produce a skewness factor that matches
%                the LOS profile skewness of the observed satellite.
%
%  [useage]:     [nu_hyp,d_hyp] = skew2nu(SatDat_s,a_hyp,e_hyp,i_hyp,...
%                    RAAN_hyp,argp_hyp,u_hyp,input,d,options)
%
%  [inputs]:     SatDat_s   = Structure containing the GMAT extracted data
%                             for the SURVEYOR.
%
%                a_hyp      = Target semi-major axis hypothesis. [km]
%
%                e_hyp      = Target eccentricity hypothesis.
%
%                i_hyp      = Target inclination hypothesis. [deg]
%
%                RAAN_hyp   = Target RAAN hypothesis. [deg]
%
%                argp_hyp   = Target argument of perigee hypothesis. [deg]
%
%                u_hyp      = Target argument of latitude hypothesis. [deg]
%
%                input      = Input structure for the Surveyor and Target
%                             which inclues the true elements and epoch of
%                             initial conditions.
%
%                d          = Skewness factor from Target observations.
%
%                options    = [bias,stdev,gran]
%                             bias = angle measurement bias [arc sec]
%                             stdev = 1-sigma angle meas. uncertainty [arc sec]
%                             gran = level of search granularity (1 = high
%                                    precision, 0 = low precision).
%
%  [outputs]:    nu_hyp     = Target true anomaly hypothesis [deg]
%
%                d_hyp      = Skewness factor for the Target hypothesis.
%
%  [refs]:
%
%  [history]:    24 Jan 2017 - Max Yates

function [nu_hyp,d_hyp] = skew2nu(SatDat_s,a_hyp,e_hyp,i_hyp,RAAN_hyp, ...
    argp_hyp,u_hyp,input,d,options)

bias = options(1);
stdev = options(2);
gran = logical(options(3));

% d>1 skews left, d<1 skews right
if d>1
```

```
    sign_nu = -1;
else
    sign_nu = 1;
end

% Even with sign_nu specified correctly, there are known cases where nu
% goes slightly negative due to the full geopotential
go_negative = 1;
found_nu = 0;

while go_negative == 1 && found_nu == 0
    dnu=sign_nu*(0.1:0.1:2)'; % if u_hyp is close enough, else widen search
    skewTestBlock
    if k==1
        dnu=sign_nu*(0:0.01:0.1)';
    else
        dnu=sign_nu*(sign_nu*dnu(k-1,1):0.01:sign_nu*dnu(k,1))';
    end
    skewTestBlock
    if go_negative == -1
        break
    end
    if k==1
        dnu=sign_nu*(0:0.001:0.01)';
    else
        dnu=sign_nu*(sign_nu*dnu(k-1,1):0.001:sign_nu*dnu(k,1))';
    end
    skewTestBlock
    if gran == 1
        if k==1
            dnu=sign_nu*(0:0.0001:0.001)';
        else
            dnu=sign_nu*(sign_nu*dnu(k-1,1):0.0001:sign_nu*dnu(k,1))';
        end
        skewTestBlock
        if k==1
            dnu=sign_nu*(0:0.00001:0.0001)';
        else
            dnu=sign_nu*(sign_nu*dnu(k-1,1):0.00001:sign_nu*dnu(k,1))';
        end
        skewTestBlock
    end
    found_nu = 1;
end

if go_negative == -1
    sign_nu = -sign_nu;

    dnu=sign_nu*(0.1:0.1:5)';
    skewTestBlock
    if k==1
        dnu=sign_nu*(0:0.01:0.1)';
    else
        dnu=sign_nu*(sign_nu*dnu(k-1,1):0.01:sign_nu*dnu(k,1))';
    end
    skewTestBlock
    if k==1
        dnu=sign_nu*(0:0.001:0.01)';
    else
        dnu=sign_nu*(sign_nu*dnu(k-1,1):0.001:sign_nu*dnu(k,1))';
    end
    skewTestBlock
    if gran == 1
        if k==1
            dnu=sign_nu*(0:0.0001:0.001)';
        else
            dnu=sign_nu*(sign_nu*dnu(k-1,1):0.0001:sign_nu*dnu(k,1))';
        end
        skewTestBlock
        if k==1
            dnu=sign_nu*(0:0.00001:0.0001)';
        else
            dnu=sign_nu*(sign_nu*dnu(k-1,1):0.00001:sign_nu*dnu(k,1))';
        end
        skewTestBlock
    end
end

if k==1
    d_hyp = d_hyp_temp(1,k);
    if abs(cosd(argp_hyp)-cosd(input.coes(1,5)))<1e-6 % same argp as Surveyor
        % Target true anomaly [deg]
        nu_hyp = input.coes(1,6) + dnu(k,1);
    else
        nu_hyp = wrap((u_hyp - argp_hyp + dnu(k,1))*pi/180)*180/pi;
    end
else % split the difference
    d_hyp = (d_hyp_temp(1,k)+d_hyp_temp(1,k-1))/2;
    if abs(cosd(argp_hyp)-cosd(input.coes(1,5)))<1e-6 % same argp as Surveyor
        % Target true anomaly [deg]
        nu_hyp = input.coes(1,6) + (dnu(k,1)+dnu(k-1,1))/2;
    else
        nu_hyp = wrap((u_hyp - argp_hyp + (dnu(k,1) + ...
            dnu(k-1,1))/2)*pi/180)*180/pi;
    end
end
```

E.8 Initial Orbit Determination

```
%  [function]    losIOD
%
%  [purpose]:    Function determines an initial orbit estimate for the
%                Target.
%
%  [useage]:     [hyp,mapping,eclipse,d,d_truth,delta_d] = losIOD(input,...
%                   SatDat,Surveyor,Target,dt,bias,stdev)

%
%  [inputs]:     input      = Structure contains truth coes for Surveyor and
%                             Target as well as the epoch and runcase string
%
%                SatDat     = Structure containing the GMAT extracted data.
%                             The exact content is configured in the
%                             gmatOrbitProp code, but presently includes
%                             for every ith satellite entries of the form:
%                             SatDat.X(:,i).
%                             NOTE: The 1st satellite is the Surveyor! All
%                             others are Target satellites.
%
%                Surveyor   = Structure containing the processed inertial
%                             data for the Surveyor.
%
%                Target     = Structure containing the processed inertial
%                             data for the Target.
%
%                dt         = Data sample rate [sec]
%
%                bias       = angle measurement bias [arc sec]
%
%                stdev      = 1-sigma angle measurement uncertainty [arc sec]
%
%  [outputs]:
%
%  [refs]:
%
%  [history]:    31 Jan 2017 - Max Yates

function [hyp,mapping,eclipse,d,d_truth,delta_d,alpha_disp] = ...
    losIOD(input,SatDat,Surveyor,Target,dt,bias,stdev)

global MU geo_degree use_skew2nu
codeDir = pwd;

% Find skewness factor
[d,delta_split] = skewFactor(Target.alpha,Target.Yrel);
d_truth = abs(min(Target.Yrel))/(abs(min(Target.Yrel)) + ...
    abs(max(Target.Yrel)))*2;

% Generate orbit hypotheses in admissible region
ranges = linspace(100,10,10)';
[hyp.coes,hyp.d,hyp.delta_split_hyp,hyp.rel_apses,alpha_disp,hyp.u_hyp] = ...
    admRegHyp(SatDat,input,Surveyor,Target.alpha,Target.beta,d,ranges,0,0);
```

```
% Propagate hypotheses for one orbital period
options = [0,dt,0,0,0]; % [prop_time,dt,torus,output,orbitview]
SatDat_hyp = gmatOrbitProp('Keplerian',input.coord_sys, ...
    [input.coes(1,:);hyp.coes],input.epoch,'UTCGregorian',options);

% Generate LOS observations from hypotheses
[Surveyor,Target_hyp] = losProc(SatDat_hyp,input,[0,0,1]);

% Which side of trajectory has the best range observability?
hyp.dalpha_span(1,1) = abs(Target_hyp.alpha_at_min_alphadot(1,end) - ...
    Target_hyp.alpha_at_min_alphadot(1,1));
hyp.dalpha_span(1,2) = abs(Target_hyp.alpha_at_min_alphadot(2,end) - ...
    Target_hyp.alpha_at_min_alphadot(2,1));
if hyp.dalpha_span(1,1) > hyp.dalpha_span(1,2)
    which_side = 1;
else
    which_side = 2;
end

% Find outlier hypotheses using Theil-Sen estimator
b = Theil_Sen_Regress(Target_hyp.alpha_at_min_alphadot(which_side,:)', ...
    Target_hyp.range_at_min_alphadot(which_side,:)'); % slope
% Polynomial coefficients: b is slope, second part is from equation of a line
% Given a point (farthest point) and a slope
p0 = [b,(-b*Target_hyp.alpha_at_min_alphadot(which_side,1) + ...
    Target_hyp.range_at_min_alphadot(which_side,1))];
Theil_Sen_ranges = polyval(p0, ...
    Target_hyp.alpha_at_min_alphadot(which_side,:)); % [km]
Theil_Sen_difference = Target_hyp.range_at_min_alphadot(which_side,:) - ...
    Theil_Sen_ranges;
meann = mean(Theil_Sen_difference); % mean
stdd = std(Theil_Sen_difference); % standard deviation
% Flag points that are 1 standard deviations away from the mean
I = bsxfun(@gt, abs(bsxfun(@minus, Theil_Sen_difference, meann)), 1*stdd);
% Find the indices of the zero elements in I; these are the "good" hypotheses
hyp.good = find(~I);

%% Initial Range Estimates from Min Alpha Dot
% Range estimate from the most observable side
p1 = polyfit(Target_hyp.alpha_at_min_alphadot(which_side,hyp.good), ...
    Target_hyp.range_at_min_alphadot(which_side,hyp.good),2);
mapping.range_true = Target.range_at_min_alphadot(which_side,1); % [km]
mapping.range_est=polyval(p1,Target.alpha_at_min_alphadot(which_side,1));%[km]
mapping.range_error = mapping.range_est - ...
    Target.range_at_min_alphadot(which_side,1); % [km]
mapping.range_percent_error = ...
    mapping.range_error/Target.range_at_min_alphadot(which_side,1)*100;

% Use range estimate from most skewed side to estimate eccentricity
p3 = polyfit(Target_hyp.range_at_min_alphadot(which_side,hyp.good), ...
    hyp.coes(hyp.good,2)',2);
mapping.ecc_from_min_alphadot = polyval(p3,mapping.range_est); % [nd]

%% Range estimate from the less observable side
if which_side==1
```

```
    which_side = 2;
elseif which_side==2
    which_side = 1;
end
p2 = polyfit(Target_hyp.alpha_at_min_alphadot(which_side,hyp.good), ...
    Target_hyp.range_at_min_alphadot(which_side,hyp.good),2);
mapping.range_true2 = Target.range_at_min_alphadot(which_side,1); % [km]
mapping.range_est2=polyval(p2,Target.alpha_at_min_alphadot(which_side,1));
mapping.range_error2 = mapping.range_est2 - ...
    Target.range_at_min_alphadot(which_side,1); % [km]
mapping.range_percent_error2 = ...
    mapping.range_error2/Target.range_at_min_alphadot(which_side,1)*100;

% Use range estimates from both sides to estimate eccentricity
if isnan(mapping.range_est2) ~= 1
    end_2_end = Target_hyp.range_at_min_alphadot(1,hyp.good) + ...
        Target_hyp.range_at_min_alphadot(2,hyp.good);
        p4 = polyfit(end_2_end,hyp.coes(hyp.good,2)',2);
        mapping.ecc_from_min_alphadot2 = ...
            polyval(p4,mapping.range_est+mapping.range_est2); % [nd]
end

%% Eclipse Analysis
eclipse_analysis
p5 = polyfit(Target_hyp.eclipse_delta_start(1,:), ...
    Target_hyp.eclipse_range_true(1,:),3);
eclipse.range_true = Target.eclipse_range_true; % [km]
eclipse.range_est = polyval(p5,Target.eclipse_delta_start(1,1)); % [km]
eclipse.range_error = eclipse.range_est - Target.eclipse_range_true; % [km]
eclipse.range_percent_error = ...
    eclipse.range_error/Target.eclipse_range_true*100 % [%]
% Can eclipse help scale eccentricity?
eclipse_support = 0;
if abs(Target_hyp.eclipse_delta_start(1,end) - ...
    Target_hyp.eclipse_delta_start(1,1)) > 10 % 10 sec threshold
    eclipse_support = 1;
    p6 = polyfit(Target_hyp.eclipse_delta_start(1,hyp.good), ...
        hyp.coes(hyp.good,2)',2);
    mapping.ecc_from_eclipse = polyval(p6,Target.eclipse_delta_start(1,1));
end

%% Estimate Target COEs
mapping.coes = zeros(1,6);
n = size(hyp.coes,1); % number of hypotheses

% RESET: Pick range estimate from the most observable side
if hyp.dalpha_span(1,1) > hyp.dalpha_span(1,2)
    which_side = 1;
else
    which_side = 2;
end

%% Eccentricity from Arcs Approaching Min Alpha Dot
% Establish upper and lower boundaries on arcs
A = [10,Target.min_alphadot_idx(1,1),1; % 10 indicates an alpha minimum
     10,Target.min_alphadot_idx(2,1),2; % 10 indicates an alpha minimum
     20,Target.max_alphadot_idx(1,1),1; % 20 indicates an alpha maximum
     20,Target.max_alphadot_idx(2,1),2];% 20 indicates an alpha maximum
B = sortrows(A,2);
if B(1,1)==10 && B(1,3)==which_side
    index1 = 1;
    index2 = B(1,2);
    index3 = B(2,2);
    peak2peak = abs(Target.alpha_dot(index3,1) - Target.alpha_dot(index2,1));
    ref1 = Target.alpha_dot(index2,1)-0.15*peak2peak;
    ref2 = Target.alpha_dot(index2,1)-0.01*peak2peak;
    ref3 = Target.alpha_dot(index3,1)+0.85*peak2peak;
elseif B(2,1)==10 && B(2,3)==which_side
    index1 = B(1,2);
    index2 = B(2,2);
    index3 = B(3,2);
    peak2peak = abs(Target.alpha_dot(index3,1) - Target.alpha_dot(index2,1));
    ref1 = Target.alpha_dot(index2,1)-0.15*peak2peak;
    ref2 = Target.alpha_dot(index2,1)-0.01*peak2peak;
    ref3 = Target.alpha_dot(index3,1)+0.85*peak2peak;
elseif B(3,1)==10 && B(3,3)==which_side
    index1 = B(2,2);
    index2 = B(3,2);
    index3 = B(4,2);
    peak2peak = abs(Target.alpha_dot(index3,1) - Target.alpha_dot(index2,1));
    ref1 = Target.alpha_dot(index2,1)-0.15*peak2peak;
    ref2 = Target.alpha_dot(index2,1)-0.01*peak2peak;
    ref3 = Target.alpha_dot(index3,1)+0.85*peak2peak;
elseif B(4,1)==10 && B(4,3)==which_side
    index1 = B(3,2);
    index2 = B(4,2);
    index3 = size(Target.alpha_dot,1);
    peak2peak = abs(Target.alpha_dot(index3,1) - Target.alpha_dot(index2,1));
    ref1 = Target.alpha_dot(index2,1)-0.15*peak2peak;
    ref2 = Target.alpha_dot(index2,1)-0.01*peak2peak;
    ref3 = Target.alpha_dot(index3,1)+0.85*peak2peak;
end

for i = 1:n+1
    if i==1
        clear p8 p9
        [value, ind] = min(abs(Target.alpha_dot(index1:index2,1)-ref1));
        index(1,i) = index1 + ind -1;
        lower_bound1 = Target.alpha_dot(index(1,i),1);
        [value, ind] = min(abs(Target.alpha_dot(index1:index2,1)-ref2));
        index(2,i) = index1 + ind -1;
        upper_bound1 = Target.alpha_dot(index(2,i),1);
        [value, ind] = min(abs(Target.alpha_dot(index2:index3,1)-ref2));
        index(3,i) = index2 + ind -1;
        lower_bound2 = Target.alpha_dot(index(3,i),1);
        [value, ind] = min(abs(Target.alpha_dot(index2:index3,1)-ref3));
        index(4,i) = index2 + ind -1;
        upper_bound2 = Target.alpha_dot(index(4,i),1);

        % Polyfit for Target arc
        poly_deg = 5; % linear = 1, quadratic = 2, cubic = 3
        if stdev > 0
        % Re-fit curve to alpha in this smaller segment -- need to
        % estimate alphadot better here
        for j=1:2
            if j==1
                idx1 = index(1,i); idx2 = index(2,i);
            elseif j==2
                idx1 = index(3,i); idx2 = index(4,i);
            end
            p7 = polyfit(SatDat.time(idx1:idx2,1), ...
                Target.alpha_meas(idx1:idx2,i),poly_deg);
            Target.alpha(idx1:idx2,i) = polyval(p7, ...
                SatDat.time(idx1:idx2,1));
            % Recalculate alphadot in this smaller segment
            for j=idx1:idx2
                % alpha rate from five-point central difference formula
                if j>(idx1+1) && j<(idx2-1)
                    Target.alpha_dot(j,i) = (Target.alpha(j-2,i) - ...
                        8*Target.alpha(j-1,i) + 8*Target.alpha(j+1,i)-...
                        Target.alpha(j+2,i))/(12*dt);
                    if abs(Target.alpha_dot(j,i)) > ...
                        1.1*abs(Target.alpha_dot(j-1,i))
                        Target.alpha_dot(j,i) = Target.alpha_dot(j-1,i);
                    end
                end
                if j<(idx1+2) % forward difference formula
                    Target.alpha_dot(j,i) = (-11*Target.alpha(j,i) + ...
                        18*Target.alpha(j+1,i)-9*Target.alpha(j+2,i)+...
                        2*Target.alpha(j+3,i))/(6*dt);
                end
                if j>(idx2-5)
                    Target.alpha_dot(j,i) = (11*Target.alpha(j,i) - ...
                        18*Target.alpha(j-1,i)+9*Target.alpha(j-2,i)-...
                        2*Target.alpha(j-3,i))/(6*dt);
                    if abs(Target.alpha_dot(j,i)) > ...
                        1.1*abs(Target.alpha_dot(j-1,i))
                        Target.alpha_dot(j,i) = Target.alpha_dot(j-1,i);
                    end
                end
            end
        end
    end
    p8(1,:) = polyfit(Target.alpha_dot(index(1,i):index(2,i),1), ...
        Target.alpha(index(1,i):index(2,i),1),poly_deg);
    p9(1,:) = polyfit(Target.alpha_dot(index(3,i):index(4,i),1), ...
        Target.alpha(index(3,i):index(4,i),1),poly_deg);
else
    [value, ind] = ...
        min(abs(Target_hyp.alpha_dot(index1:index2,i-1)-ref1));
    index(1,i) = index1 + ind -1;
    [value, ind] = ...
        min(abs(Target_hyp.alpha_dot(index1:index2,i-1)-ref2));
    index(2,i) = index1 + ind -1;
    [value, ind] = ...
        min(abs(Target_hyp.alpha_dot(index2:index3,i-1)-ref2));
    index(3,i) = index2 + ind -1;
    [value, ind] = ...
```

```
            min(abs(Target_hyp.alpha_dot(index2:index3,i-1)-ref3));
        index(4,i) = index2 + ind -1;

        % Polyfits for each of the hypothesis arcs
        p8(i,:) = polyfit(Target_hyp.alpha_dot(index(1,i):index(2,i), ...
            i-1),Target_hyp.alpha(index(1,i):index(2,i),i-1),poly_deg);
        p9(i,:) = polyfit(Target_hyp.alpha_dot(index(3,i):index(4,i), ...
            i-1),Target_hyp.alpha(index(3,i):index(4,i),i-1),poly_deg);
    end
end

% Angle rates to use for comparing the polyfits from each hypothesis
waypoints1 = linspace(lower_bound1,upper_bound1,100);

alpha_at_wp = zeros(n+1,size(waypoints1,2));
for j = 1:size(waypoints1,2)
    for i = 1:n+1
        alpha_at_wp(i,j) = polyval(p8(i,:),waypoints1(1,j));
    end
    p10(j,:) = polyfit(alpha_at_wp((hyp.good+1),j),hyp.coes(hyp.good,2),2);
    mapping.ecc_arc(j,1) = polyval(p10(j,:),alpha_at_wp(1,j)); % [nd]
end

waypoints2 = linspace(lower_bound2,upper_bound2,100);

alpha_at_wp = zeros(n+1,size(waypoints2,2));
for j = 1:size(waypoints2,2)
    for i = 1:n+1
        alpha_at_wp(i,j) = polyval(p9(i,:),waypoints2(1,j));
    end
    p11(j,:) = polyfit(alpha_at_wp((hyp.good+1),j),hyp.coes(hyp.good,2),2);
    mapping.ecc_arc(j,2) = polyval(p11(j,:),alpha_at_wp(1,j)); % [nd]
end

mapping.ecc_from_arc = mean((mapping.ecc_arc(:,1)+mapping.ecc_arc(:,2))/2);

% Plot verifying arc (alpha vs. alpha rate)
figure(19)
hold on
linecolors = jet(n);
legend_label_common = '\boldmath$e=';
plot(Target.alpha_true(index(3,1):index(4,1),1)*180/pi, ...
    abs(Target.alpha_dot2(index(3,1):index(4,1),1))*180/pi*3600,'-k', ...
    'MarkerSize',16,'LineWidth',3)
legend_label{1}=strcat(legend_label_common,sprintf('%.4f', ...
    input.coes(2,2)),'$');
for i=1:n
    plot(Target_hyp.alpha(index(3,i+1):index(4,i+1),i)*180/pi, ...
        abs(Target_hyp.alpha_dot(index(3,i+1):index(4,i+1),i))*180/pi*3600,...
        '-k','color',linecolors(i,:),'MarkerSize',16,'LineWidth',3)
    legend_label{i+1}=strcat(legend_label_common,sprintf('%.4f', ...
        hyp.coes(i,2)),'$');
end
% Waypoints
xLimits = get(gca,'XLim');
for i=1:size(waypoints1,2)
```

```
    plot(xLimits,abs([waypoints1(1,i),waypoints1(1,i)])*180/pi*3600, ...
        '--k','LineWidth',1.5)
end
grid off
box on
xlabel('\boldmath$\alpha$\textbf{ (deg)}','Interpreter','latex','FontSize',33)
ylabel('\boldmath$|\dot{\alpha}|$\textbf{ (arc sec/sec)}','Interpreter', ...
    'latex','FontSize',33)
ind = 1:1:n; % Indices for legend labels
h = legend(legend_label{1,ind},'location','NorthEast');
set(h,'Interpreter','latex','FontSize',32); %17
set(gca,'FontSize',32,'FontWeight','bold') %17
axis normal;
saveas(gcf,strcat(codeDir,'\Cases\',sprintf('%s',input.runCase),'\Fig19.fig'))

figure(20)
hold on
linecolors = jet(n);
legend_label_common = '\boldmath$e=';
plot(Target.alpha(index(1,1):index(2,1),1)*180/pi, ...
    abs(Target.alpha_dot(index(1,1):index(2,1),1))*180/pi*3600,'-k', ...
    'MarkerSize',16,'LineWidth',3)
legend_label{1}=strcat(legend_label_common,sprintf('%.4f', ...
    input.coes(2,2)),'$');
for i=1:n
    plot(Target_hyp.alpha(index(1,i+1):index(2,i+1),i)*180/pi, ...
        abs(Target_hyp.alpha_dot(index(1,i+1):index(2,i+1),i))*180/pi*3600,...
        '-k','color',linecolors(i,:),'MarkerSize',16,'LineWidth',3)
    legend_label{i+1}=strcat(legend_label_common,sprintf('%.4f', ...
        hyp.coes(i,2)),'$');
end
% Waypoints
xLimits = get(gca,'XLim');
for i=1:size(waypoints2,2)
    plot(xLimits,abs([waypoints2(1,i),waypoints2(1,i)])*180/pi*3600, ...
        '--k','LineWidth',1.5)
end
grid off
box on
xlabel('\boldmath$\alpha$\textbf{ (deg)}','Interpreter','latex','FontSize',33)
ylabel('\boldmath$|\dot{\alpha}|$\textbf{ (arc sec/sec)}','Interpreter', ...
    'latex','FontSize',33)
ind = 1:1:n; % Indices for legend labels
h = legend(legend_label{1,ind},'location','NorthWest');
set(h,'Interpreter','latex','FontSize',32);
set(gca,'FontSize',32,'FontWeight','bold')
axis normal;
saveas(gcf,strcat(codeDir,'\Cases\',sprintf('%s',input.runCase),'\Fig20.fig'))

%% Estimate Target Eccentricity
if eclipse_support == 1
    if geo_degree == 0
        mapping.ecc = mapping.ecc_from_arc
    else
        mapping.ecc = (mapping.ecc_from_min_alphadot + ...
            mapping.ecc_from_arc + mapping.ecc_from_eclipse)/3
    end
else
    if geo_degree == 0
        mapping.ecc = mapping.ecc_from_arc
    else
        mapping.ecc = (mapping.ecc_from_min_alphadot + ...
            mapping.ecc_from_arc)/2
    end
end
mapping.coes(1,2) = mapping.ecc;

%% Estimate Target Inclination
p12 = polyfit(hyp.coes(hyp.good,2)',hyp.coes(hyp.good,3)',2);
mapping.coes(1,3) = polyval(p12,mapping.coes(1,2)); % [deg]

%% Estimate Target RAAN
p13 = polyfit(hyp.coes(hyp.good,2)',hyp.coes(hyp.good,4)',2);
mapping.coes(1,4) = polyval(p13,mapping.coes(1,2)); % [deg]

%% Estimate Target Semi-Major Axis
if abs(alpha_disp) > 2
    p14 = polyfit(hyp.coes(hyp.good,2)',hyp.coes(hyp.good,1)',2);
    mapping.coes(1,1) = polyval(p14,mapping.coes(1,2)); % [deg]
else
    mapping.coes(1,1) = input.coes(1,1);
end

% Estimate Target Argument of Perigee
mapping.coes(1,5) = hyp.coes(1,5); % Constant for all hypotheses

%% Estimate Target True Anomaly
% If 20x20 gravity model, then find nu from skew
if geo_degree > 0 && use_skew2nu == 1
    options = [0,0,0,0,0]; % [prop_time,dt,torus,output,orbitview]
    SatDat_s = gmatOrbitProp(input.state_type,input.coord_sys, ...
        input.elements(1,:),input.epoch,'UTCGregorian',options);
    [mapping.coes(1,6),d_hyp_nu,~] = skew2nu(SatDat_s,...
        mapping.coes(1,1),mapping.coes(1,2),mapping.coes(1,3),...
        mapping.coes(1,4),mapping.coes(1,5),hyp.u_hyp,input,d, ...
        [bias,stdev,1]);
else
    p15 = polyfit(hyp.coes(hyp.good,2)',hyp.coes(hyp.good,6)',2);
    mapping.coes(1,6) = polyval(p15,mapping.coes(1,2)); % [deg]
end

%% Alpha vs. Alpha-Rate Analysis & Plots
figOn = [0,  % alpha profile comparison with baseline               FIGURE 02
         1,  % alpha vs. alpha rate (-180 deg to 180 deg)           FIGURE 03
         1,  % alpha vs. |alpha rate| @ min peak near 90 deg        FIGURE 04
         0,  % alpha vs. |alpha rate| @ min peak near -90 deg       FIGURE 05
         0,  % alpha vs. |alpha rate| @ first max peak              FIGURE 06
         0,  % alpha vs. |alpha rate| @ second max peak             FIGURE 07
         1,  % rs-plane motion with location of extrema             FIGURE 08
         1,  % sw-plane motion with location of extrema             FIGURE 09
         1,  % delta alpha at min alpha_dot vs. range               FIGURE 10
         0,  % alpha at max alpha_dot (first peak) vs. range        FIGURE 11
         0]; % alpha at max alpha_dot (second peak) vs. range       FIGURE 12

close all; clc;

n = size(hyp.coes,1); % number of hypotheses

%% Analyze alpha profiles with a baseline hypothesis
%  COMPARE ALPHA PROFILES to a baseline alpha profile to show where range is
%  observable. The eccentricity of the target, in all cases, must be either
%  greater than or less than the surveyor (no mixing where some are greater
%  than or some are less than, otherwise the alpha profiles are 180deg apart.

b = 1; % which hypothesis to use as baseline for comparing alpha profiles
linecolors = jet(n);
for i=1:n
    % The sine of alpha can be searched for zero crossings. No zero crossings
    % indicates the relative trajectory does not enclose the Surveyor.
    % Get sign of data
    signum = sign(sin(Target_hyp.alpha(:,i)));
    % Set sign of exact data zeros to positive
    signum(sin(Target_hyp.alpha(:,i))==0) = 1;
    % Get zero crossings by diff ~= 0
    zero_crossings = find(diff(signum)~=0);
    if zero_crossings > 0 % only enclosed relative orbits
        Target_hyp.delta_alpha(:,i) = Target_hyp.alpha(:,i) - ...
            Target_hyp.alpha(:,b);
        if Target_hyp.target_at_perigee(1,i) == ...
            Target_hyp.target_at_perigee(1,b)
            figure(2) % delta alpha from some baseline
            hold on
            % skip first and last couple points because of spikes
            plot(Target_hyp.alpha(3:end-2,b)*180/pi, ...
                Target_hyp.delta_alpha(3:end-2,i)*180/pi*3600,'k.', ...
                'color',linecolors(i,:),'MarkerSize',16,'LineWidth',1.5)
            xlim([-180 180])
            ylim([-4000 4000])
            grid on; grid minor;
            box on
            xlabel('\textbf{Baseline }\boldmath$\alpha$\textbf{ (deg)}', ...
                'Interpreter','latex','FontSize',33)
            ylabel('\boldmath$\delta\alpha$\textbf{ (arc sec)}', ...
                'Interpreter','latex','FontSize',33)
            set(gca,'FontSize',32,'FontWeight','bold')
            axis normal;
        end
    end
end
saveas(gcf,strcat(codeDir,'\Cases\',sprintf('%s',input.runCase),'\Fig02.fig'))
if figOn(1)==0
    close
end

%% Alpha vs. Alpha-Rate
figure(3)
hold on
linecolors = jet(n);
```

```
legend_label_common = '\boldmath$e=';
for i=1:n
    % skip plotting the last three points
    plot(Target_hyp.alpha(1:end-3,i)*180/pi, ...
        abs(Target_hyp.alpha_dot(1:end-3,i))*180/pi*3600,'k.','color', ...
        linecolors(i,:),'MarkerSize',16,'LineWidth',1.5)
    legend_label{i}=strcat(legend_label_common,sprintf('%.4f', ...
        hyp.coes(i,2)),'$');
end
xlim([-180 180])
grid on; grid minor;
box on
xlabel('\boldmath$\alpha$\textbf{ (deg)}','Interpreter','latex', ...
    'FontSize',33)
ylabel('\boldmath$|\dot{\alpha}|$\textbf{ (arc sec/sec)}','Interpreter', ...
    'latex','FontSize',33)
ind = 1:1:n; % Indices for legend labels
h = legend(legend_label{1,ind},'location','South');
set(h,'Interpreter','latex','FontSize',32);
set(gca,'FontSize',32,'FontWeight','bold')
axis normal;
saveas(gcf,strcat(codeDir,'\Cases\',sprintf('%s',input.runCase),'\Fig03.fig'))
if figOn(2)==0
    close
end

%% Alpha vs. |Alpha-Rate| @ Min Peak Near 90 deg
if isnan(Target.min_alphadot(1,1)) ~= 1
    figure(4)
    hold on
    linecolors = jet(n);
    legend_label_common = ...
        '\boldmath$\rho\ ($\textbf{at }\boldmath$\dot{\alpha}_{\min})=';
    for i=1:n
        plot((Target_hyp.alpha_at_min_alphadot(1,i)/3600+90),...
            abs(Target_hyp.alpha_dot(Target_hyp.min_alphadot_idx(1,i), ...
            i))*180/pi*3600, 'o','MarkerSize',25,'MarkerEdgeColor', ...
            linecolors(i,:),'MarkerFaceColor',linecolors(i,:))
    end
    for i=1:n
        % Skip plotting the last three points
        plot(Target_hyp.alpha(1:end-3,i)*180/pi, ...
            abs(Target_hyp.alpha_dot(1:end-3,i))*180/pi*3600,'color', ...
            linecolors(i,:),'MarkerSize',2,'LineWidth',3)
        legend_label{i}=strcat(legend_label_common,sprintf('%.1f', ...
            Target_hyp.range_at_min_alphadot(1,i)),'$\textbf{ km}');
    end
    xlim([min(Target_hyp.alpha_at_min_alphadot(1,:)/3600+90)-0.1, ...
        max(Target_hyp.alpha_at_min_alphadot(1,:)/3600+90)+0.1])
    rate = abs(Target_hyp.alpha_dot(Target_hyp.min_alphadot_idx(1,i), ...
        i))*180/pi*3600;
    ylim([rate-0.5 rate+0.5])
    grid on; grid minor;
    box on
    xlabel('\boldmath$\alpha$\textbf{ (deg)}','Interpreter','latex', ...
        'FontSize',33)
```

```
ylabel('\boldmath$|\dot{\alpha}|$\textbf{ (arc sec/sec)}', ...
    'Interpreter','latex','FontSize',33)
ind = 1:1:n; % Indices for legend labels
h = legend(legend_label{1,ind},'location','EastOutside');
set(h,'Interpreter','latex','FontSize',32);
set(gca,'FontSize',32,'FontWeight','bold')
axis normal;
% Add analytical predictions for low-delta energy cases
if abs(alpha_disp) < 2 % disparity_threshold
    for i=1:n
        % Mean motion from Surveyor
        N = sqrt(MU/input.coes(1,1)^3);
        % Find initial eccentric anomaly
        Et0 = 2*atan(tand(hyp.coes(i,6)/2)/sqrt((1+hyp.coes(i,2))/(1- ...
            hyp.coes(i,2)))); % [rad]
        Es0 = 2*atan(tand(input.coes(1,6)/2)/sqrt((1+ ...
            input.coes(1,2))/(1-input.coes(1,2)))); % [rad]
        % Find initial mean anomaly
        Mt0 = Et0 - hyp.coes(i,2)*sin(Et0); % [rad]
        Ms0 = Es0 - input.coes(1,2)*sin(Es0); % [rad]
        syms t et es
        Dnu = Mt0 + 2*hyp.coes(i,2)*sin(Mt0+N*t) + ...
            5/4*(hyp.coes(i,2)^2)*sin(2*Mt0+2*N*t) -...
            Ms0 - 2*input.coes(1,2)*sin(Ms0+N*t) - ...
            5/4*(input.coes(1,2)^2)*sin(2*Ms0+2*N*t);
        dDnudt = diff(Dnu,t);
        f = matlabFunction(dDnudt);
        if Target_hyp.Xrel(1,i)<0
            interval = [0 pi/N];
        else
            interval = [pi/N 2*pi/N];
        end
        t = fzero(f, interval);
        rs = input.coes(1,1)*(1+0.5*(input.coes(1,2)^2)- ...
            input.coes(1,2)*cos(Ms0+N*t)- ...
            0.5*(input.coes(1,2)^2)*cos(2*Ms0+2*N*t));
        rt = input.coes(1,1)*(1+0.5*(hyp.coes(i,2)^2)- ...
            hyp.coes(i,2)*cos(Mt0+N*t)- ...
            0.5*(hyp.coes(i,2)^2)*cos(2*Mt0+2*N*t));
        Dnu = eval(Dnu); % [rad]
        rt_rproj = rt*cos(Dnu);
        x = -(rs - rt_rproj);
        y = rt*sin(Dnu);
        alpha = atan2(y,x)*180/pi;
        % store alpha
        Target_hyp.analytical_alpha_at_min_alphadot(1,i) = ...
            (alpha-90)*3600; % [arc secs]
        Target_hyp.analytical_range_at_min_alphadot(1,i) = ...
            sqrt(x^2 + y^2); % [km]

        yLimits = get(gca,'YLim');  % Get the range of the y axis
        plot([alpha alpha],yLimits,'--k','Color',linecolors(i,:), ...
            'LineWidth',3)
    end
    % Analytical Resolvability - range resolution referenced to rs-plane
    % (larger ranges if beta was calculated) [km/arc sec]
```

```
        Target_hyp.analytical_observability(1,1) = ...
            (max(Target_hyp.analytical_range_at_min_alphadot(1,:)) -...
            min(Target_hyp.analytical_range_at_min_alphadot(1,:)))/ ...
            abs(max(Target_hyp.analytical_alpha_at_min_alphadot(1,:)) -...
            min(Target_hyp.analytical_alpha_at_min_alphadot(1,:)));
        % Numerical Resolvability - range resolution referenced to rs-plane
        [max_rho ind] = max(Target_hyp.range_at_min_alphadot(1,:));
        beta_at_max_rho = Target_hyp.beta(Target_hyp.min_alphadot_idx(1, ...
            ind),ind);
        [min_rho ind] = min(Target_hyp.range_at_min_alphadot(1,:));
        beta_at_min_rho = Target_hyp.beta(Target_hyp.min_alphadot_idx(1, ...
            ind),ind);
        Target_hyp.numerical_observability(1,1) = (max_rho*...
            cos(beta_at_max_rho) - min_rho*cos(beta_at_min_rho))/ ...
            abs(max(Target_hyp.alpha_at_min_alphadot(1,:)) -...
            min(Target_hyp.alpha_at_min_alphadot(1,:))); % [km/arc sec]
    end

saveas(gcf,strcat(codeDir,'\Cases\',sprintf('%s',input.runCase),'\Fig04.fig'))
    if figOn(3)==0
        close
    end
end

%% Alpha vs. |Alpha-Rate| @ Min Peak Near -90 deg
if isnan(Target.min_alphadot(2,1)) ~= 1
    figure(5)
    hold on
    linecolors = jet(n);
    legend_label_common = ...
        '\boldmath$\rho\ ($\textbf{at }\boldmath$\dot{\alpha}_{\min})=';
    for i=1:n
        plot((Target_hyp.alpha_at_min_alphadot(2,i)/3600-90),...
            abs(Target_hyp.alpha_dot(Target_hyp.min_alphadot_idx(2,i), ...
            i))*180/pi*3600,'o','MarkerSize',25,'MarkerEdgeColor', ...
            linecolors(i,:),'MarkerFaceColor',linecolors(i,:))
    end
    for i=1:n
        % Skip plotting the last three points
        plot(Target_hyp.alpha(1:end-3,i)*180/pi, ...
            abs(Target_hyp.alpha_dot(1:end-3,i))*180/pi*3600,'color',...
            linecolors(i,:),'MarkerSize',2,'LineWidth',3)
        legend_label{i}=strcat(legend_label_common,sprintf('%.1f', ...
            Target_hyp.range_at_min_alphadot(2,i)),'$\textbf{ km}');
    end
    xlim([min(Target_hyp.alpha_at_min_alphadot(2,:)/3600-90)-0.1, ...
        max(Target_hyp.alpha_at_min_alphadot(2,:)/3600-90)+0.1])
    rate = abs(Target_hyp.alpha_dot(Target_hyp.min_alphadot_idx(2,i), ...
        i))*180/pi*3600;
    ylim([rate-0.5 rate+0.5])
    grid on; grid minor;
    box on
    xlabel('\boldmath$\alpha$\textbf{ (deg)}','Interpreter','latex', ...
        'FontSize',33)
    ylabel('\boldmath$|\dot{\alpha}|$\textbf{ (arc sec/sec)}', ...
        'Interpreter','latex','FontSize',33)
```

```
ind = 1:1:n; % Indices for legend labels
h = legend(legend_label{1,ind},'location','EastOutside');
set(h,'Interpreter','latex','FontSize',32);
set(gca,'FontSize',32,'FontWeight','bold')
axis normal;
% Add analytical predictions for low-delta energy cases
if abs(alpha_disp) < 2 % disparity_threshold
    for i=1:n
        % Mean motion from Surveyor
        N = sqrt(MU/input.coes(1,1)^3);
        % Find initial eccentric anomaly
        Et0 = 2*atan(tand(hyp.coes(i,6)/2)/sqrt((1+hyp.coes(i,2))/(1-...
            hyp.coes(i,2))));
        Es0 = 2*atan(tand(input.coes(1,6)/2)/sqrt((1+input.coes(1,2))/...
            (1-input.coes(1,2))));
        % Find initial mean anomaly
        Mt0 = Et0 - hyp.coes(i,2)*sin(Et0);
        Ms0 = Es0 - input.coes(1,2)*sin(Es0);
        syms t et es
        Dnu = Mt0 + 2*hyp.coes(i,2)*sin(Mt0+N*t) + ...
            5/4*(hyp.coes(i,2)^2)*sin(2*Mt0+2*N*t) - ...
            Ms0 - 2*input.coes(1,2)*sin(Ms0+N*t) - ...
            5/4*(input.coes(1,2)^2)*sin(2*Ms0+2*N*t);
        dDnudt = diff(Dnu,t);
        f = matlabFunction(dDnudt);
        if Target_hyp.Xrel(1,i)<0
            interval = [pi/N 2*pi/N];
        else
            interval = [0 pi/N];
        end
        t = fzero(f, interval);
        rs = input.coes(1,1)*(1+0.5*(input.coes(1,2)^2) - ...
            input.coes(1,2)*cos(Ms0+N*t) - ...
            0.5*(input.coes(1,2)^2)*cos(2*Ms0+2*N*t));
        rt = input.coes(1,1)*(1+0.5*(hyp.coes(i,2)^2) - ...
            hyp.coes(i,2)*cos(Mt0+N*t) - ...
            0.5*(hyp.coes(i,2)^2)*cos(2*Mt0+2*N*t));
        Dnu = eval(Dnu);
        rt_rproj = rt*cos(Dnu);
        x = -(rs - rt_rproj);
        y = rt*sin(Dnu);
        alpha = atan2(y,x)*180/pi;
        % store alpha
        Target_hyp.analytical_alpha_at_min_alphadot(2,i) = ...
            (alpha+90)*3600; % [arc secs]
        Target_hyp.analytical_range_at_min_alphadot(2,i) = ...
            sqrt(x^2 + y^2); % [km]

        yLimits = get(gca,'YLim');  % Get the range of the y axis
        plot([alpha alpha],yLimits,'--k','Color',linecolors(i,:), ...
           'LineWidth',3)
    end
    % Analytical Resolvability - range resolution referenced to rs-plane
    % (larger ranges if beta was calculated) [km/arc sec]
    Target_hyp.analytical_observability(2,1) = ...
        (max(Target_hyp.analytical_range_at_min_alphadot(2,:)) - ...
            min(Target_hyp.analytical_range_at_min_alphadot(2,:)))/ ...
            abs(max(Target_hyp.analytical_alpha_at_min_alphadot(2,:)) - ...
            min(Target_hyp.analytical_alpha_at_min_alphadot(2,:)));
        % Numerical Resolvability - range resolution referenced to rs-plane
        [max_rho ind] = max(Target_hyp.range_at_min_alphadot(2,:));
        beta_at_max_rho = ...
            Target_hyp.beta(Target_hyp.min_alphadot_idx(2,ind),ind);
        [min_rho ind] = min(Target_hyp.range_at_min_alphadot(2,:));
        beta_at_min_rho = ...
            Target_hyp.beta(Target_hyp.min_alphadot_idx(2,ind),ind);
        Target_hyp.numerical_observability(2,1) = ...
            (max_rho*cos(beta_at_max_rho) - min_rho*cos(beta_at_min_rho))/...
            abs(max(Target_hyp.alpha_at_min_alphadot(2,:)) - ...
            min(Target_hyp.alpha_at_min_alphadot(2,:))); % [km/arc sec]
    end

saveas(gcf,strcat(codeDir,'\Cases\',sprintf('%s',input.runCase),'\Fig05.fig'))
    if figOn(4)==0
        close
    end
end

%% Alpha vs. |Alpha-Rate| @ First Max Peak
if isnan(Target.max_alphadot(1,1)) ~= 1
    figure(6)
    hold on
    linecolors = jet(n);
    legend_label_common = ...
        '\boldmath$\rho\ ($\textbf{at }\boldmath$\dot{\alpha}_{\max})=';
    for i=1:n
        plot(Target_hyp.alpha_at_max_alphadot(1,i),...
            abs(Target_hyp.alpha_dot(Target_hyp.max_alphadot_idx(1,i), ...
            i))*180/pi*3600,'o','MarkerSize',25,'MarkerEdgeColor', ...
            linecolors(i,:),'MarkerFaceColor',linecolors(i,:))
    end
    for i=1:n
        % Skip plotting the last three points
        plot(Target_hyp.alpha(1:end-3,i)*180/pi, ...
            abs(Target_hyp.alpha_dot(1:end-3,i))*180/pi*3600,'color',...
            linecolors(i,:),'MarkerSize',2,'LineWidth',3) %1
        legend_label{i}=strcat(legend_label_common,sprintf('%.1f', ...
            Target_hyp.range_at_max_alphadot(1,i)),'$\textbf{ km}');
    end
    xlim([min(Target_hyp.alpha_at_max_alphadot(1,:))-0.1, ...
        max(Target_hyp.alpha_at_max_alphadot(1,:))+0.1])
    rate = abs(Target_hyp.alpha_dot(Target_hyp.max_alphadot_idx(1,i), ...
        i))*180/pi*3600;
    ylim([rate-2 rate+2])
    grid on; grid minor;
    box on
    xlabel('\boldmath$\alpha$\textbf{ (deg)}','Interpreter','latex', ...
        'FontSize',33)
    ylabel('\boldmath$|\dot{\alpha}|$\textbf{ (arc sec/sec)}', ...
        'Interpreter','latex','FontSize',33)
    ind = 1:1:n; % Indices for legend labels
    h = legend(legend_label{1,ind},'location','EastOutside');
    set(h,'Interpreter','latex','FontSize',32);
    set(gca,'FontSize',32,'FontWeight','bold')
    axis normal;

saveas(gcf,strcat(codeDir,'\Cases\',sprintf('%s',input.runCase),'\Fig06.fig'))
    if figOn(5)==0
        close
    end
end

%% Alpha vs. |Alpha-Rate| @ Second Max Peak
if isnan(Target.max_alphadot(2,1)) ~= 1
    figure(7)
    hold on
    linecolors = jet(n);
    legend_label_common = ...
        '\boldmath$\rho\ ($\textbf{at }\boldmath$\dot{\alpha}_{\max})=';
    for i=1:n
        plot(Target_hyp.alpha_at_max_alphadot(2,i),...
            abs(Target_hyp.alpha_dot(Target_hyp.max_alphadot_idx(2,i), ...
            i))*180/pi*3600,'o','MarkerSize',25,'MarkerEdgeColor', ...
            linecolors(i,:),'MarkerFaceColor',linecolors(i,:))
    end
    for i=1:n
        % Skip plotting the last three points
        plot(Target_hyp.alpha(1:end-3,i)*180/pi, ...
            abs(Target_hyp.alpha_dot(1:end-3,i))*180/pi*3600,'color',...
            linecolors(i,:),'MarkerSize',2,'LineWidth',3)
        legend_label{i}=strcat(legend_label_common,sprintf('%.1f', ...
            Target_hyp.range_at_max_alphadot(2,i)),'$\textbf{ km}');
    end
    xlim([min(Target_hyp.alpha_at_max_alphadot(2,:))-0.1, ...
        max(Target_hyp.alpha_at_max_alphadot(2,:))+0.1])
    rate = abs(Target_hyp.alpha_dot(Target_hyp.max_alphadot_idx(2,i), ...
        i))*180/pi*3600;
    ylim([rate-2 rate+2])
    grid on; grid minor;
    box on
    xlabel('\boldmath$\alpha$\textbf{ (deg)}','Interpreter','latex', ...
        'FontSize',33)
    ylabel('\boldmath$|\dot{\alpha}|$\textbf{ (arc sec/sec)}', ...
        'Interpreter','latex','FontSize',33)
    ind = 1:1:n; % Indices for legend labels
    h = legend(legend_label{1,ind},'location','EastOutside');
    set(h,'Interpreter','latex','FontSize',32);
    set(gca,'FontSize',32,'FontWeight','bold')
    axis normal;

saveas(gcf,strcat(codeDir,'\Cases\',sprintf('%s',input.runCase),'\Fig07.fig'))
    if figOn(6)==0
        close
    end
end

%% RS-plane Motion with Location of Extrema
figure(8)
```

```
hold on
linecolors = jet(n);
legend_label_common = ...
    '\boldmath$\rho\ ($\textbf{at }\boldmath$\dot{\alpha}_{\min})=';
legend_label{1}=strcat(legend_label_common,sprintf('%.1f', ...
    Target.range_at_min_alphadot(1,1)),'$\textbf{ km}');
% MINIMA
plot(Target.Yrel(Target.min_alphadot_idx(1,1),1),...
    Target.Xrel(Target.min_alphadot_idx(1,1),1),'o','MarkerSize',10, ...
    'MarkerEdgeColor','k','MarkerFaceColor','k')
for i=1:n
    plot(Target_hyp.Yrel(Target_hyp.min_alphadot_idx(1,i),i),...
        Target_hyp.Xrel(Target_hyp.min_alphadot_idx(1,i),i),'o', ...
        'MarkerSize',10,'MarkerEdgeColor',linecolors(i,:), ...
        'MarkerFaceColor',linecolors(i,:))
    legend_label{i+1} = strcat(legend_label_common,sprintf('%.1f', ...
        Target_hyp.range_at_min_alphadot(1,i)),'$\textbf{ km}');
end
if isnan(Target.min_alphadot(2,1)) ~= 1
    plot(Target.Yrel(Target.min_alphadot_idx(2,1),1),...
        Target.Xrel(Target.min_alphadot_idx(2,1),1),...
        'o','MarkerSize',10,'MarkerEdgeColor','k','MarkerFaceColor','k')
    for i=1:n
        plot(Target_hyp.Yrel(Target_hyp.min_alphadot_idx(2,i),i), ...
            Target_hyp.Xrel(Target_hyp.min_alphadot_idx(2,i),i),'o', ...
            'MarkerSize',10,'MarkerEdgeColor',linecolors(i,:), ...
            'MarkerFaceColor',linecolors(i,:))
    end
end
% MAXIMA
plot(Target.Yrel(Target.max_alphadot_idx(1,1),1),...
        Target.Xrel(Target.max_alphadot_idx(1,1),1),...
        's','MarkerSize',10,'MarkerEdgeColor','k','MarkerFaceColor','k')
for i=1:n
    if isnan(Target_hyp.max_alphadot_idx(1,i))==0
        plot(Target_hyp.Yrel(Target_hyp.max_alphadot_idx(1,i),i),...
            Target_hyp.Xrel(Target_hyp.max_alphadot_idx(1,i),i),'s', ...
            'MarkerSize',10,'MarkerEdgeColor',linecolors(i,:), ...
            'MarkerFaceColor',linecolors(i,:))
    end
end
if isnan(Target.max_alphadot(2,1)) ~= 1
    plot(Target.Yrel(Target.max_alphadot_idx(2,1),1), ...
        Target.Xrel(Target.max_alphadot_idx(2,1),1), ...
        's','MarkerSize',10,'MarkerEdgeColor','k','MarkerFaceColor','k')
    for i=1:n
        if isnan(Target_hyp.max_alphadot_idx(2,i))==0
            plot(Target_hyp.Yrel(Target_hyp.max_alphadot_idx(2,i),i), ...
                Target_hyp.Xrel(Target_hyp.max_alphadot_idx(2,i),i),'s', ...
                'MarkerSize',10,'MarkerEdgeColor',linecolors(i,:), ...
                'MarkerFaceColor',linecolors(i,:))
        end
    end
end
% TRAJECTORIES
for i=1:n
```

```
    plot(Target_hyp.Yrel(:,i),Target_hyp.Xrel(:,i),'-','color', ...
        linecolors(i,:),'LineWidth',3)
end
plot(Target.Yrel(:,1),Target.Xrel(:,1),'-k','LineWidth',3)
grid on; grid minor;
box on
xlabel('\textbf{Along-Track (km)}','Interpreter','latex','FontSize',33)
ylabel('\textbf{Radial (km)}','Interpreter','latex','FontSize',33)
set(gca,'FontSize',32,'FontWeight','bold')
axis normal; axis equal;
saveas(gcf,strcat(codeDir,'\Cases\',sprintf('%s',input.runCase),'\Fig08.fig'))
if figOn(7)==0
    close
end

%% SW-plane Motion with Location of Extrema
figure(9)
hold on
linecolors = jet(n);
legend_label_common = ...
    '\boldmath$\rho\ ($\textbf{at }\boldmath$\dot{\alpha}_{\min})=';
legend_label{1} = strcat(legend_label_common,sprintf('%.1f', ...
    Target.range_at_min_alphadot(1,1)),'$\textbf{ km}');
% MINIMA
plot(Target.Yrel(Target.min_alphadot_idx(1,1),1), ...
        Target.Zrel(Target.min_alphadot_idx(1,1),1),'o','MarkerSize',10, ...
        'MarkerEdgeColor','k','MarkerFaceColor','k')
for i=1:n
    plot(Target_hyp.Yrel(Target_hyp.min_alphadot_idx(1,i),i), ...
        Target_hyp.Zrel(Target_hyp.min_alphadot_idx(1,i),i),'o', ...
        'MarkerSize',10,'MarkerEdgeColor',linecolors(i,:), ...
        'MarkerFaceColor',linecolors(i,:))
    legend_label{i+1} = strcat(legend_label_common,sprintf('%.1f', ...
        Target_hyp.range_at_min_alphadot(1,i)),'$\textbf{ km}');
end
if isnan(Target_hyp.min_alphadot(2,1)) ~= 1
    plot(Target.Yrel(Target.min_alphadot_idx(2,1),1), ...
        Target.Zrel(Target.min_alphadot_idx(2,1),1),'o','MarkerSize',10, ...
        'MarkerEdgeColor','k','MarkerFaceColor','k')
    for i=1:n
        plot(Target_hyp.Yrel(Target_hyp.min_alphadot_idx(2,i),i), ...
            Target_hyp.Zrel(Target_hyp.min_alphadot_idx(2,i),i),'o', ...
        'MarkerSize',10,'MarkerEdgeColor',linecolors(i,:), ...
        'MarkerFaceColor',linecolors(i,:))
    end
end
% MAXIMA
plot(Target.Yrel(Target.max_alphadot_idx(1,1),1), ...
        Target.Zrel(Target.max_alphadot_idx(1,1),1),'s','MarkerSize',10, ...
        'MarkerEdgeColor','k','MarkerFaceColor','k')
for i=1:n
    if isnan(Target_hyp.max_alphadot_idx(1,i))==0
        plot(Target_hyp.Yrel(Target_hyp.max_alphadot_idx(1,i),i),...
            Target_hyp.Zrel(Target_hyp.max_alphadot_idx(1,i),i),'s',...
        'MarkerSize',10,'MarkerEdgeColor',linecolors(i,:), ...
        'MarkerFaceColor',linecolors(i,:))
    end
end
if isnan(Target_hyp.max_alphadot(2,1)) ~= 1
    plot(Target.Yrel(Target.max_alphadot_idx(2,1),1), ...
        Target.Zrel(Target.max_alphadot_idx(2,1),1),'s','MarkerSize',10, ...
        'MarkerEdgeColor','k','MarkerFaceColor','k')
    for i=1:n
        if isnan(Target_hyp.max_alphadot_idx(2,i))==0
            plot(Target_hyp.Yrel(Target_hyp.max_alphadot_idx(2,i),i), ...
                Target_hyp.Zrel(Target_hyp.max_alphadot_idx(2,i),i),'s', ...
                'MarkerSize',10,'MarkerEdgeColor',linecolors(i,:), ...
                'MarkerFaceColor',linecolors(i,:))
        end
    end
end
% TRAJECTORIES
for i=1:n
    plot(Target_hyp.Yrel(:,i),Target_hyp.Zrel(:,i),'-','color', ...
        linecolors(i,:),'LineWidth',3)
end
plot(Target.Yrel(:,1),Target.Zrel(:,1),'-k','LineWidth',3)
grid on; grid minor;
box on
xlabel('\textbf{Along-Track (km)}','Interpreter','latex','FontSize',33)
ylabel('\textbf{Cross-Track (km)}','Interpreter','latex','FontSize',33)
set(gca,'FontSize',32,'FontWeight','bold')
axis normal; axis equal;
saveas(gcf,strcat(codeDir,'\Cases\',sprintf('%s',input.runCase),'\Fig09.fig'))
if figOn(8)==0
    close
end

%% Delta Alpha at Min Alpha-Rate vs. Range
figure(10)
hold on
plot(Target_hyp.alpha_at_min_alphadot(1,:), ...
    Target_hyp.range_at_min_alphadot(1,:),'-k','LineWidth',5);
if isnan(Target_hyp.min_alphadot(2,1)) ~= 1
    plot(Target_hyp.alpha_at_min_alphadot(2,:), ...
        Target_hyp.range_at_min_alphadot(2,:),'--k','LineWidth',5);
end
linecolors = jet(n);
for i=1:n
    plot(Target_hyp.alpha_at_min_alphadot(1,i), ...
        Target_hyp.range_at_min_alphadot(1,i),'o','MarkerSize',20, ...
        'MarkerEdgeColor',linecolors(i,:),'MarkerFaceColor',linecolors(i,:))
end
if isnan(Target_hyp.min_alphadot(2,1)) ~= 1
    for i=1:n
        plot(Target_hyp.alpha_at_min_alphadot(2,i), ...
        Target_hyp.range_at_min_alphadot(2,i),'o','MarkerSize',20,...
            'MarkerEdgeColor',linecolors(i,:),'MarkerFaceColor', ...
            linecolors(i,:))
    end
end
% Overlay the dots
```

```
plot(Target_hyp.alpha_at_min_alphadot(1,:), ...
    Target_hyp.range_at_min_alphadot(1,:),'-k','LineWidth',5);
if isnan(Target_hyp.min_alphadot(2,1)) ~= 1
    plot(Target_hyp.alpha_at_min_alphadot(2,:), ...
        Target_hyp.range_at_min_alphadot(2,:),'--k','LineWidth',5);
end
grid on; grid minor;
box on
xlabel('\boldmath$\Delta\alpha$\textbf{ (arc sec) at
}\boldmath$\dot{\alpha}_{\min}$','Interpreter','latex','FontSize',33)
ylabel('\textbf{Range (km)}','Interpreter','latex','FontSize',33)
if isnan(Target_hyp.min_alphadot(2,1)) ~= 1
    h = legend('\boldmath$\Delta\alpha$\textbf{ from
}\boldmath$\alpha=90$\textbf{ deg}','\boldmath$\Delta\alpha$\textbf{ from
}\boldmath$\alpha=-90$\textbf{ deg}','location','North');
elseif d<1
    h = legend('\boldmath$\Delta\alpha$\textbf{ from
}\boldmath$\alpha=90$\textbf{ deg}','location','NorthWest');
elseif d>1
    h = legend('\boldmath$\Delta\alpha$\textbf{ from }\boldmath$\alpha=-
90$\textbf{ deg}','location','NorthEast');
end
set(h,'Interpreter','latex','FontSize',32);
set(gca,'FontSize',32,'FontWeight','bold')
axis normal;
saveas(gcf,strcat(codeDir,'\Cases\',sprintf('%s',input.runCase),'\Fig10.fig'))
if figOn(9)==0
    close
end

%% Alpha at Max Alpha-Rate (First Peak) vs. Range
if isnan(Target_hyp.max_alphadot(1,1)) ~= 1
    figure(11)
    hold on
    plot(Target_hyp.alpha_at_max_alphadot(1,:), ...
        Target_hyp.range_at_max_alphadot(1,:),'--b','LineWidth',5);
    linecolors = jet(n);
    for i=1:n
        plot(Target_hyp.alpha_at_max_alphadot(1,i), ...
            Target_hyp.range_at_max_alphadot(1,i),'o','MarkerSize',25,...
            'MarkerEdgeColor',linecolors(i,:),'MarkerFaceColor', ...
            linecolors(i,:))
    end
    grid on; grid minor;
    box on
    xlabel('\boldmath$\alpha$\textbf{ (deg) at
}\boldmath$\dot{\alpha}_{max}$','Interpreter','latex','FontSize',33)
    ylabel('\textbf{Range (km)}','Interpreter','latex','FontSize',33)
    set(gca,'FontSize',32,'FontWeight','bold')
    axis normal;

saveas(gcf,strcat(codeDir,'\Cases\',sprintf('%s',input.runCase),'\Fig11.fig'))
    if figOn(10)==0
        close
    end
end
```

```
%% Alpha at Max Alpha-Rate (Second Peak) vs. Range
if isnan(Target_hyp.max_alphadot(2,1)) ~= 1
    figure(12)
    hold on
    plot(Target_hyp.alpha_at_max_alphadot(2,:), ...
        Target_hyp.range_at_max_alphadot(2,:),'--r','LineWidth',5);
    linecolors = jet(n);
    for i=1:n
        plot(Target_hyp.alpha_at_max_alphadot(2,i), ...
            Target_hyp.range_at_max_alphadot(2,i),'o','MarkerSize',25, ...
            'MarkerEdgeColor',linecolors(i,:),'MarkerFaceColor', ...
            linecolors(i,:))
    end
    grid on; grid minor;
    box on
    xlabel('\boldmath$\alpha$\textbf{ (deg) at
}\boldmath$\dot{\alpha}_{max}$','Interpreter','latex','FontSize',33)
    ylabel('\textbf{Range (km)}','Interpreter','latex','FontSize',33)
    set(gca,'FontSize',32,'FontWeight','bold')
    axis normal;

saveas(gcf,strcat(codeDir,'\Cases\',sprintf('%s',input.runCase),'\Fig12.fig'))
    if figOn(11)==0
        close
    end
end

%% Review Hypotheses
% Propagate hypotheses for one orbital period
options = [0,0,0,0,0]; % [prop_time,dt,torus,output,orbitview]
SatDat_review = gmatOrbitProp('Keplerian',input.coord_sys, ...
    [input.coes(1,:);hyp.coes;input.coes(2,:);mapping.coes],input.epoch, ...
    'UTCGregorian',options);
% Generate LOS observations from hypotheses
[Surveyor,Target_review] = losProc(SatDat_review,input,[0,0,1]);

% Hypotheses
figure(1)
clf
hold on
for i=1:size(hyp.coes,1)
    plot(Target_review.Yrel(:,i),Target_review.Xrel(:,i),'-k','LineWidth',3)
    xlabel('\textbf{Along-Track (km)}','Interpreter','latex','FontSize',33)
    ylabel('\textbf{Radial (km)}','Interpreter','latex','FontSize',33)
    set(gca,'FontSize',32,'FontWeight','bold')
end
axis normal; axis equal;
grid on; grid minor;
box on
saveas(gcf,strcat(codeDir,'\Cases\',sprintf('%s',input.runCase),'\Fig01.fig'))

% Shadow Plot of Truth and IOD Estimate
p = size(SatDat_review.time(:,1),1);
figure(21)
clf;
```

```
hold on
plot3(Target_review.Yrel(:,end-1),Target_review.Zrel(:,end-1), ...
    Target_review.Xrel(:,end-1),'-k','LineWidth',3); % truth
plot3(Target_review.Yrel(:,end),Target_review.Zrel(:,end), ...
    Target_review.Xrel(:,end),':r','LineWidth',3); % estimate
% Surveyor location
plot3(0,0,0,'ko','MarkerSize',8,'MarkerEdgeColor','w','MarkerFaceColor','k');
view([45 -45])
axis equal fill
sl = xlim;
wl = ylim;
rl = zlim;
% Projection on rw-plane  (constant s limit)
plot3(sl(1)*ones(p,1),Target_review.Zrel(:,end-1), ...
    Target_review.Xrel(:,end-1),'-k','LineWidth',2); % truth
plot3(sl(1)*ones(p,1),Target_review.Zrel(:,end), ...
    Target_review.Xrel(:,end),':r','LineWidth',2); % estimate
plot3(sl(1),0,0,'ko','MarkerSize',8,'MarkerEdgeColor','w', ...
    'MarkerFaceColor','k');
% Projection on rs-plane (constant w limit)
plot3(Target_review.Yrel(:,end-1),wl(2)*ones(p,1), ...
    Target_review.Xrel(:,end-1),'-k','LineWidth',2); % truth
plot3(Target_review.Yrel(:,end),wl(2)*ones(p,1), ...
    Target_review.Xrel(:,end),':r','LineWidth',2); % estimate
plot3(0,wl(2),0,'ko','MarkerSize',8,'MarkerEdgeColor','w', ...
    'MarkerFaceColor','k');
% Projection on sw-plane  (constant -r limit)
plot3(Target_review.Yrel(:,end-1),Target_review.Zrel(:,end-1), ...
    rl(1)*ones(p,1),'-k','LineWidth',2); % truth
plot3(Target_review.Yrel(:,end),Target_review.Zrel(:,end),rl(1)*ones(p,1),...
    ':r','LineWidth',2); % estimate
plot3(0,0,rl(1),'ko','MarkerSize',8,'MarkerEdgeColor','w', ...
    'MarkerFaceColor','k');
view([30 25]);
set(gca,'FontSize',32,'FontWeight','bold')
grid on; grid minor;
box on;  % standard black box wireframe
set(gca,'Color',[0.5 0.5 0.5],'GridLineStyle','-','MinorGridLineStyle',...
    ':','LineWidth',1)
xlabel('\textbf{S (km)}','Interpreter','latex','FontSize',33)
ylabel('\textbf{W (km)}','Interpreter','latex','FontSize',33)
zlabel('\textbf{R (km)}','Interpreter','latex','FontSize',33)
set(gca,'FontSize',32,'FontWeight','bold')
pos=get(gca,'Position');
h = legend('\textbf{Truth}','\textbf{Hypothesis}','Location','NorthEast');
set(h,'Interpreter','latex','FontSize',32);
saveas(gcf,strcat(codeDir,'\Cases\',sprintf('%s',input.runCase),'\Fig21.fig'))

save(strcat(codeDir,'\Cases\',sprintf('%s_iod.mat',input.runCase)))
```

E.9 Nonlinear Least Squares

```
%% NLSQ.m

format long g
clear r_RMS alpha_res_mean beta_res_mean

%% INPUTS
% Specify the number of observations which will determine the number
% of times through the data processing loop
num_obs = size(SatDat.time(:,1),1)-1;
% Set maximum number of least squares iterations
max_iter = 10;
% Rejection standard deviations
rs = 10000; % Set high to keep everything
% Measurement Uncertainty
ang_res = (10)/3600*pi/180; % approximate sigma = 10 arc sec
rho_unc = 1000000000; % set it high!

% Set index limits for time
t1=1;
t2=t1+num_obs;

% Set initial correction to reference trajectory
dX0 = [0;0;0;0;0;0];
X0 = [0;0;0;0;0;0];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Output IOD error at epoch to screen
x0 = (SatDat.Xef(1,2)-SatDat_iod.Xef(1,2))*1000;
y0 = (SatDat.Yef(1,2)-SatDat_iod.Yef(1,2))*1000;
z0 = (SatDat.Zef(1,2)-SatDat_iod.Zef(1,2))*1000;
vx0 = (SatDat.VXef(1,2)-SatDat_iod.VXef(1,2))*1000 - Omegae*y0;
vy0 = (SatDat.VYef(1,2)-SatDat_iod.VYef(1,2))*1000 + Omegae*x0;
vz0 = (SatDat.VZef(1,2)-SatDat_iod.VZef(1,2))*1000;
[x0;y0;z0;vx0;vy0;vz0]

% Instrument covariance matrix.
Q = [ang_res^2,0,0;0,ang_res^2,0;0,0,rho_unc^2];
Q_inv = [1/ang_res^2,0,0;0,1/ang_res^2,0;0,0,1/rho_unc^2];

% Noise on measurements
noise = zeros(3,num_obs);
for i=1:num_obs
    noise(:,i) = [ang_res;ang_res;0].*randn(3,1); % mean zero, 1 sigma
end

% Observation times
t=SatDat.time(t1:t2,1); % secs

% Set up z, the total data (observation) vector % 3xn matrix
z = [Target.alpha_meas(t1+1:t2,1)';Target.beta_meas(t1+1:t2,1)'; ...
    zeros(num_obs,1)'];
```

```
%% Begin iteration loop
% Initialize iteration so the 'while' loop will begin processing.
iteration = 1;
P_inv_prev = zeros(3);
while iteration <= max_iter
    % Write iteration to screen for progress monitoring.
    fprintf('Iteration %d of %d\n',iteration,max_iter)
    fprintf('\n') % Blank line for spacing on screen.

    %% Initialize buffers for populating vectors/matrices
    Xref = zeros(6,num_obs+1); % propagated state at obs time, ti
    Xref_rel = zeros(3,num_obs+1); % ref relative coordinates at obs time, ti
    rho_est = zeros(num_obs+1,1); % ref range at obs time, ti
    zref = zeros(3,num_obs); % obs prediction at obs time, ti
    r = zeros(3,num_obs); % residual
    rejected = zeros(1,num_obs); % rejection matrix
    % For product of (T transpose)(Q inverse)(r)
    % Dimensions: (6 x n) (n x n) (n x 1) = (6 x 1)
    T_tran_Q_inv_r = zeros(6,1);
    % Initialize covariance matrix inverse (6 x 6)
    P_inv = zeros(6);

    %% Epoch
    X0plus = X0+dX0
    dX_dX0_cum = eye(6);
    Xref(:,1) = X0plus; % [m,m/s]
    % Calculate Xref_rel at epoch
    Xref_ecef = [SatDat_iod.Xef(t1,2);SatDat_iod.Yef(t1,2); ...
        SatDat_iod.Zef(t1,2)] + Xref(1:3,1)./1000; % [km]
    % Convert ECEF to ECI
    JD = SatDat.UTCMJ(t1,2) + 2430000.0 + deltaJD; % Julian Date
    dummy = [0;0;0]; % use for v_ECI and a_ECI in ECItoECEF
    [Xref_eci ~] = ECEFtoECI(JD,Xref_ecef(1:3,1),dummy,dummy,Omegae,dtheta);
    % Convert ECI to RSW
    Xref_rsw = Surveyor.rotECI2RSW(:,:,t1)*Xref_eci; % [km]
    Xref_rel(1,1) = Xref_rsw(1,1) - Surveyor.Xrsw(1,1); % [km]
    Xref_rel(2,1) = Xref_rsw(2,1) - Surveyor.Yrsw(1,1); % [km]
    Xref_rel(3,1) = Xref_rsw(3,1) - Surveyor.Zrsw(1,1); % [km]
    rho_est(1,1)=sqrt(Xref_rel(1,1)^2+Xref_rel(2,1)^2+Xref_rel(3,1)^2); %[km]

    %% Observation processing loop
    for iob = t1:t2-t1
        % Propagate the state vector to the observation time, ti, and
        % obtain the state transition matrix, dX_dX0(ti,t0). Use IOD torus
        % or IOD integration to get Target's estimated position & velocity
        % in ECEF frame for generating the A matrix.
        if use_torus == 1
            q = zeros(1,3); % preallocate
            qdot = zeros(1,3); % preallocate
            Cj = zeros(length(C2)-1,3); % preallocate
            Sj = zeros(length(C2)-1,3); % preallocate
            % Sum Fourier Series
            % start with q
            Q = Omega*t(iob)/60/13.446852+Q0;
            JQ=J(2:end,:)*Q';
            cJQ=cos(JQ);
            sJQ=sin(JQ);
            Cj(:,1)=C2(2:end,1).*cJQ;
            Sj(:,1)=S2(2:end,1).*sJQ;
            Cj(:,2)=C2(2:end,2).*cJQ;
            Sj(:,2)=S2(2:end,2).*sJQ;
            Cj(:,3)=C2(2:end,3).*cJQ;
            Sj(:,3)=S2(2:end,3).*sJQ;
            q(1,1)=C2(1,1)+sum(Cj(:,1))+sum(Sj(:,1));
            q(1,2)=C2(1,2)+sum(Cj(:,2))+sum(Sj(:,2));
            q(1,3)=C2(1,3)+sum(Cj(:,3))+sum(Sj(:,3));
            % now for qdot
            Jw=J(2:end,:)*Omega';
            Cj(:,1)=C2(2:end,1).*sJQ;
            Sj(:,1)=S2(2:end,1).*cJQ;
            Cj(:,2)=C2(2:end,2).*sJQ;
            Sj(:,2)=S2(2:end,2).*cJQ;
            Cj(:,3)=C2(2:end,3).*sJQ;
            Sj(:,3)=S2(2:end,3).*cJQ;
            qdot(1,1)=sum(Jw.*(-Cj(:,1)+Sj(:,1)));
            qdot(1,2)=sum(Jw.*(-Cj(:,2)+Sj(:,2)));
            qdot(1,3)=sum(Jw.*(-Cj(:,3)+Sj(:,3)));
            r_ECEF = q'*6378135; % [m]
            v_ECEF = qdot'*6378135/60/13.446852; % [m/sec]
        elseif use_torus == 0
            r_ECEF = [SatDat_iod.Xef(iob,2);SatDat_iod.Yef(iob,2); ...
                SatDat_iod.Zef(iob,2)]*1000; % [m]
            v_ECEF = [SatDat_iod.VXef(iob,2);SatDat_iod.VYef(iob,2); ...
                SatDat_iod.VZef(iob,2)]*1000; % [m/sec]
        end
        [A,Ham,~] = Amatrix(r_ECEF,v_ECEF,SatDat_iod.UTCMJ(iob,2),method);
        % state transition matrix
        dt=t(iob+1)-t(iob); % [sec]
        dX_dX0 = eye(6) + A*dt + (A^2)*(dt^2)/2 + (A^3)*(dt^3)/(2*3) + ...
            (A^4)*(dt^4)/(2*3*4) + (A^5)*(dt^5)/(2*3*4*5) + ...
            (A^6)*(dt^6)/(2*3*4*5*6);
        dX_dX0 = dX_dX0*dX_dX0_cum;
        Xref(:,iob+1) = dX_dX0*X0plus; % this is actually delta_ECEF
        dX_dX0_cum = dX_dX0;

        % Use IOD torus or IOD integration to get Target's estimated
        % position & velocity in ECEF frame at propagated observation time
        if use_torus == 1
            q = zeros(1,3); % preallocate
            qdot = zeros(1,3); % preallocate
            Cj = zeros(length(C2)-1,3); % preallocate
            Sj = zeros(length(C2)-1,3); % preallocate
            % Sum Fourier Series
            % start with q
            Q = Omega*t(iob+1)/60/13.446852+Q0;
            JQ=J(2:end,:)*Q';
            cJQ=cos(JQ);
            sJQ=sin(JQ);
            Cj(:,1)=C2(2:end,1).*cJQ;
            Sj(:,1)=S2(2:end,1).*sJQ;
            Cj(:,2)=C2(2:end,2).*cJQ;
            Sj(:,2)=S2(2:end,2).*sJQ;
            Cj(:,3)=C2(2:end,3).*cJQ;
            Sj(:,3)=S2(2:end,3).*sJQ;
            q(1,1)=C2(1,1)+sum(Cj(:,1))+sum(Sj(:,1));
            q(1,2)=C2(1,2)+sum(Cj(:,2))+sum(Sj(:,2));
            q(1,3)=C2(1,3)+sum(Cj(:,3))+sum(Sj(:,3));
            % now for qdot
            Jw=J(2:end,:)*Omega';
            Cj(:,1)=C2(2:end,1).*sJQ;
            Sj(:,1)=S2(2:end,1).*cJQ;
            Cj(:,2)=C2(2:end,2).*sJQ;
            Sj(:,2)=S2(2:end,2).*cJQ;
            Cj(:,3)=C2(2:end,3).*sJQ;
            Sj(:,3)=S2(2:end,3).*cJQ;
            qdot(1,1)=sum(Jw.*(-Cj(:,1)+Sj(:,1)));
            qdot(1,2)=sum(Jw.*(-Cj(:,2)+Sj(:,2)));
            qdot(1,3)=sum(Jw.*(-Cj(:,3)+Sj(:,3)));
            r_ECEF = q'*6378135; % [m]
            v_ECEF = qdot'*6378135/60/13.446852; % [m/sec]
        elseif use_torus == 0
            r_ECEF = [SatDat_iod.Xef(iob+1,2);SatDat_iod.Yef(iob+1,2); ...
                SatDat_iod.Zef(iob+1,2)]*1000; % [m]
            v_ECEF = [SatDat_iod.VXef(iob+1,2);SatDat_iod.VYef(iob+1,2); ...
                SatDat_iod.VZef(iob+1,2)]*1000; % [m/sec]
        end

        Xref_ecef = [r_ECEF;v_ECEF] + Xref(:,iob+1); % [m, m/sec]
        % Subtract rotation rate of earth from velocity (gives ECEF velocity)
        Xref_ecef(4,1) = Xref_ecef(4,1) + Omegae*Xref(2,iob+1);
        Xref_ecef(5,1) = Xref_ecef(5,1) - Omegae*Xref(1,iob+1);
        % Convert ECEF to ECI
        JD = SatDat.UTCMJ(iob+1,2) + 2430000.0 + deltaJD; % Julian Date
        [Xref_eci VXref_eci AXref_eci theta] = ECEFtoECI(JD, ...
            Xref_ecef(1:3,1),Xref_ecef(4:6,1),dummy,Omegae,dtheta);
        % Convert ECI to RSW
        Xref_rsw = Surveyor.rotECI2RSW(:,:,iob+1)*(Xref_eci/1000); % [km]
        Xref_rel(1,iob+1) = Xref_rsw(1,1) - Surveyor.Xrsw(iob+1,1); % [km]
        Xref_rel(2,iob+1) = Xref_rsw(2,1) - Surveyor.Yrsw(iob+1,1); % [km]
        Xref_rel(3,iob+1) = Xref_rsw(3,1) - Surveyor.Zrsw(iob+1,1); % [km]
        rho_est(iob+1,1) = sqrt(Xref_rel(1,iob+1)^2 +Xref_rel(2,iob+1)^2 +...
            Xref_rel(3,iob+1)^2); % [km]
        alpha_ref = atan2(Xref_rel(2,iob+1),Xref_rel(1,iob+1)); % rad
        beta_ref = atan2(Xref_rel(3,iob+1),sqrt(Xref_rel(1,iob+1)^2 + ...
            Xref_rel(2,iob+1)^2)); % rad
        zref(:,iob) = [alpha_ref;beta_ref;rho_est(iob+1,1)];

        % Obtain the residual vector, ri=zi-G(Xpred). Calculate Hi(ti) or
        % dG_dX(Xref(ti),ti) for this particular data point. The
        % observation matrix Ti=Hi*dX_dX0 can be calculated.
        H = dG_dX(Xref_ecef(1:3,1),theta*pi/180, ...
            Surveyor.rotECI2RSW(:,:,iob+1),[Surveyor.Xrsw(iob+1,1); ...
            Surveyor.Yrsw(iob+1,1);Surveyor.Zrsw(iob+1,1)]*1000);

        % Form matrix product T = H * dX/dX0
        % Dimensions: (n x 6) = (n x 6)*(6 x 6), where
        % n = number of rows in the observation z matrix = 3
        T = H*dX_dX0; % observation matrix
```

```
    % Residual
    r(:,iob) = z(:,iob) - zref(:,iob);

    % Rejection processing
    % First address the possibility of angles jumping 360 deg
    for i = 1:2 % First two rows in the observation z matrix
        if r(i,iob) > pi
            r(i,iob) = z(i,iob) - zref(i,iob) - 2*pi;
        elseif r(i,iob) < -pi
            r(i,iob) = z(i,iob) - zref(i,iob) + 2*pi;
        end

        % Compare the elements of r(i) with its corresponding
        % diagonal entry of the Q_inv matrix
        if abs(r(i,iob)) > rs*ang_res
            % Set residual rejection flag to sort/omit rejected obs
            rejected(iob) = 1;
        end
    end

    % If the observation is not rejected, add new terms to the running
    % sums of the matrices: (T transpose)*(Q inverse)*(T) and
    % (T transpose)*(Q inverse)*(r).
    if (rejected(iob) ~= 1) % Check if 'rejected' is not equal to 1
        % Form product P_inv = (T transpose)*(Q inverse)*(T)
        % This product is the "observability condition." It
        % must be invertible for an estimate to exist.
        % Dimensions: (6 x 6)= (6 x n)*(n x n) *(n x 6)
        P_inv = P_inv + (T' * Q_inv * T);
        % Form product (T transpose)*(Q inverse)*(r)
        % Dimensions: (6 x 1) = (6 x 1) + (6 x n)*(n x n)*(n x 1)
        T_tran_Q_inv_r = T_tran_Q_inv_r + (T' * Q_inv * r(:,iob));
    end
end

%% Data is processed; improve estimate of dX0 at EPOCH.
% You are NOT calculating dX0 at every time step.

% Invert matrix T transpose Q inverse T to find covariance P
% Dimensions: (6 x 6) = inv((6 x n)*(n x n)*(n x 6))
P = inv(P_inv); % covariance of the correction dX0

% Multiply P by T transpose Q inverse r to get correction
% to the reference state vector at epoch.
% Dimensions: (6 x 1) = (6 x 6)*(6 x 1)
dX0 = (P_inv\T_tran_Q_inv_r)

%% Check convergence.
% RMS Residual
clear tn rn
i=1;
for iob=1:num_obs
    if rejected(iob)==0
        tn(i,1)=t(iob+1,1);
        rn(:,i)=r(:,iob); %filtered residuals
        i=i+1;
        end
    end
    r_RMS(iteration,:)=sqrt(sum(rn.^2,2)/size(rn,2)); % radians
    % Print final RMS residuals to screen
    r_RMS(iteration,1:2)*180/pi*3600

    rho_true = Target.rho_true(t1:t2,1); % [km]
    drho = rho_true - rho_est;
    drho2 = rho_true - Target_iod.rho_true(t1:t2,1);
    drho_percent = (drho./rho_true)*100;
    drho_percent2 = (drho2./rho_true)*100;

    X0 = X0plus;
    P_inv_prev = P_inv;

    if iteration > 1 && sqrt(dX0(1,1)^2 + dX0(2,1)^2 + dX0(3,1)^2) < 0.1
        fprintf('Converged on iteration %d of %d\n',iteration,max_iter);
        break
    elseif iteration == max_iter
        fprintf('Did not converge. :(');
    end
    % Increment iteration value by 1
    iteration = iteration + 1;
end % End statement for the iterations to max_iter loop

%% Generate Final Estimated Trajectory
Xref(:,1) = X0; % [m,m/s]
% Calculate Xref_rel at epoch
Xref_ecef=[SatDat_iod.Xef(t1,2);SatDat_iod.Yef(t1,2);SatDat_iod.Zef(t1,2)]...
    + Xref(1:3,1)./1000; % [km]
% Convert ECEF to ECI
JD = SatDat.UTCMJ(t1,2) + 2430000.0 + deltaJD; % Julian Date
dummy = [0;0;0]; % use for v_ECI and a_ECI in ECItoECEF
[Xref_eci ~] = ECEFtoECI(JD,Xref_ecef(1:3,1),dummy,dummy,Omegae,dtheta);
% Convert ECI to RSW
Xref_rsw = Surveyor.rotECI2RSW(:,:,t1)*Xref_eci; % [km]
Xref_rel(1,1) = Xref_rsw(1,1) - Surveyor.Xrsw(1,1); % [km]
Xref_rel(2,1) = Xref_rsw(2,1) - Surveyor.Yrsw(1,1); % [km]
Xref_rel(3,1) = Xref_rsw(3,1) - Surveyor.Zrsw(1,1); % [km]
rho_est(1,1) =sqrt(Xref_rel(1,1)^2 +Xref_rel(2,1)^2 +Xref_rel(3,1)^2); % [km]

dX_dX0_cum = eye(6);
for iob = t1:t2-t1
    % Propagate the state vector to the observation time, ti, and obtain the
    % state transition matrix, dX_dX0(ti,t0). Use IOD torus or IOD
    % integration to get Target's estimated position & velocity in ECEF frame
    % for generating the A matrix.
    if use_torus == 1
        q = zeros(1,3); % preallocate
        qdot = zeros(1,3); % preallocate
        Cj = zeros(length(C2)-1,3); % preallocate
        Sj = zeros(length(C2)-1,3); % preallocate
        % Sum Fourier Series
        % start with q
        Q = Omega*t(iob)/60/13.446852+Q0;
        JQ=J(2:end,:)*Q';
    cJQ=cos(JQ);
    sJQ=sin(JQ);
    Cj(:,1)=C2(2:end,1).*cJQ;
    Sj(:,1)=S2(2:end,1).*sJQ;
    Cj(:,2)=C2(2:end,2).*cJQ;
    Sj(:,2)=S2(2:end,2).*sJQ;
    Cj(:,3)=C2(2:end,3).*cJQ;
    Sj(:,3)=S2(2:end,3).*sJQ;
    q(1,1)=C2(1,1)+sum(Cj(:,1))+sum(Sj(:,1));
    q(1,2)=C2(1,2)+sum(Cj(:,2))+sum(Sj(:,2));
    q(1,3)=C2(1,3)+sum(Cj(:,3))+sum(Sj(:,3));
    % now for qdot
    Jw=J(2:end,:)*Omega';
    Cj(:,1)=C2(2:end,1).*sJQ;
    Sj(:,1)=S2(2:end,1).*cJQ;
    Cj(:,2)=C2(2:end,2).*sJQ;
    Sj(:,2)=S2(2:end,2).*cJQ;
    Cj(:,3)=C2(2:end,3).*sJQ;
    Sj(:,3)=S2(2:end,3).*cJQ;
    qdot(1,1)=sum(Jw.*(-Cj(:,1)+Sj(:,1)));
    qdot(1,2)=sum(Jw.*(-Cj(:,2)+Sj(:,2)));
    qdot(1,3)=sum(Jw.*(-Cj(:,3)+Sj(:,3)));
    r_ECEF = q'*6378135; % [m]
    v_ECEF = qdot'*6378135/60/13.446852; % [m/sec]
elseif use_torus == 0
    r_ECEF = [SatDat_iod.Xef(iob,2);SatDat_iod.Yef(iob,2); ...
        SatDat_iod.Zef(iob,2)]*1000; % [m]
    v_ECEF = [SatDat_iod.VXef(iob,2);SatDat_iod.VYef(iob,2); ...
        SatDat_iod.VZef(iob,2)]*1000; % [m/sec]
end

[A,Ham,a_ECI] = Amatrix(r_ECEF,v_ECEF,SatDat_iod.UTCMJ(iob,2),method);
% State transition matrix
dt=t(iob+1)-t(iob); % [sec]
dX_dX0 = eye(6) + A*dt + (A^2)*(dt^2)/2 + (A^3)*(dt^3)/(2*3) + ...
    (A^4)*(dt^4)/(2*3*4) + (A^5)*(dt^5)/(2*3*4*5) + ...
    (A^6)*(dt^6)/(2*3*4*5*6);
dX_dX0 = dX_dX0*dX_dX0_cum;
Xref(:,iob+1) = dX_dX0*X0; % this is actually delta_ECEF
dX_dX0_cum = dX_dX0;

% Use IOD torus or IOD integration to get Target's estimated position
% & velocity in ECEF frame at propagated observation time
if use_torus == 1
    q = zeros(1,3); % preallocate
    qdot = zeros(1,3); % preallocate
    Cj = zeros(length(C2)-1,3); % preallocate
    Sj = zeros(length(C2)-1,3); % preallocate
    % Sum Fourier Series
    % start with q
    Q = Omega*t(iob+1)/60/13.446852+Q0;
    JQ=J(2:end,:)*Q';
    cJQ=cos(JQ);
    sJQ=sin(JQ);
    Cj(:,1)=C2(2:end,1).*cJQ;
    Sj(:,1)=S2(2:end,1).*sJQ;
    Cj(:,2)=C2(2:end,2).*cJQ;
    Sj(:,2)=S2(2:end,2).*sJQ;
    Cj(:,3)=C2(2:end,3).*cJQ;
    Sj(:,3)=S2(2:end,3).*sJQ;
    q(1,1)=C2(1,1)+sum(Cj(:,1))+sum(Sj(:,1));
    q(1,2)=C2(1,2)+sum(Cj(:,2))+sum(Sj(:,2));
    q(1,3)=C2(1,3)+sum(Cj(:,3))+sum(Sj(:,3));
    % now for qdot
    Jw=J(2:end,:)*Omega';
    Cj(:,1)=C2(2:end,1).*sJQ;
    Sj(:,1)=S2(2:end,1).*cJQ;
    Cj(:,2)=C2(2:end,2).*sJQ;
    Sj(:,2)=S2(2:end,2).*cJQ;
    Cj(:,3)=C2(2:end,3).*sJQ;
    Sj(:,3)=S2(2:end,3).*cJQ;
    qdot(1,1)=sum(Jw.*(-Cj(:,1)+Sj(:,1)));
    qdot(1,2)=sum(Jw.*(-Cj(:,2)+Sj(:,2)));
    qdot(1,3)=sum(Jw.*(-Cj(:,3)+Sj(:,3)));
    r_ECEF = q'*6378135; % [m]
    v_ECEF = qdot'*6378135/60/13.446852; % [m/sec]
elseif use_torus == 0
    r_ECEF = [SatDat_iod.Xef(iob+1,2);SatDat_iod.Yef(iob+1,2); ...
        SatDat_iod.Zef(iob+1,2)]*1000; % [m]
    v_ECEF = [SatDat_iod.VXef(iob+1,2);SatDat_iod.VYef(iob+1,2); ...
        SatDat_iod.VZef(iob+1,2)]*1000; % [m/sec]
end

Xref_ecef = [r_ECEF;v_ECEF] + Xref(:,iob+1); % [m, m/sec]
% Subtract rotation rate of earth from velocity (gives ECEF velocity)
Xref_ecef(4,1) = Xref_ecef(4,1) + Omegae*Xref(2,iob+1);
Xref_ecef(5,1) = Xref_ecef(5,1) - Omegae*Xref(1,iob+1);
% convert ECEF to ECI
JD = SatDat.UTCMJ(iob+1,2) + 2430000.0 + deltaJD; % Julian Date
[Xref_eci VXref_eci AXref_eci theta] = ECEFtoECI(JD,Xref_ecef(1:3,1), ...
    Xref_ecef(4:6,1),dummy,Omegae,dtheta);
% Convert ECI to RSW
Xref_rsw = Surveyor.rotECI2RSW(:,:,iob+1)*(Xref_eci/1000); % [km]
Xref_rel(1,iob+1) = Xref_rsw(1,1) - Surveyor.Xrsw(iob+1,1); % [km]
Xref_rel(2,iob+1) = Xref_rsw(2,1) - Surveyor.Yrsw(iob+1,1); % [km]
Xref_rel(3,iob+1) = Xref_rsw(3,1) - Surveyor.Zrsw(iob+1,1); % [km]
rho_est(iob+1,1) = sqrt(Xref_rel(1,iob+1)^2 + Xref_rel(2,iob+1)^2 + ...
    Xref_rel(3,iob+1)^2); % [km]
alpha_ref = atan2(Xref_rel(2,iob+1),Xref_rel(1,iob+1)); % rad
beta_ref = atan2(Xref_rel(3,iob+1),sqrt(Xref_rel(1,iob+1)^2 + ...
    Xref_rel(2,iob+1)^2)); % rad
zref(:,iob) = [alpha_ref;beta_ref;rho_est(iob+1,1)];
r(:,iob) = z(:,iob) - zref(:,iob);
% Address the possibility of angles jumping 360 deg
for i = 1:2 % First two rows in the observation z matrix.
    if r(i,iob) > pi
        r(i,iob) = z(i,iob) - zref(i,iob) - 2*pi;
    elseif r(i,iob) < -pi
        r(i,iob) = z(i,iob) - zref(i,iob) + 2*pi;
    end
end
if iob==t1
```

```
        % Estimated COEs
        [p,a,ecc,incl,omega,argp,nu,~] = rv2coe((Xref_eci')/1000, ...
            (VXref_eci')/1000);
        NLS.coes = [a,ecc,incl*180/pi,omega*180/pi,argp*180/pi,nu*180/pi];
    end
end

%% Plot & Save
% RMS Residual
clear tn rn
i=1;
for iob=1:num_obs
    if rejected(iob)==0
        tn(i,1)=t(iob+1,1);
        rn(:,i)=r(:,iob); %filtered residuals
        i=i+1;
    end
end
r_RMS_final=sqrt(sum(rn.^2,2)/size(rn,2)); % radians

rho_true = Target.rho_true(t1:t2,1); % [km]
drho = abs(rho_true - rho_est);
drho2 = abs(rho_true - Target_iod.rho_true(t1:t2,1));
drho_percent = (drho./rho_true)*100;
drho_percent2 = (drho2./rho_true)*100;

% Plot least squares residuals
figure(30)
clf;
plot(tn/60,rn(1,:)*180/pi*3600,'k','LineWidth',3);
hold on
plot(tn/60,rn(2,:)*180/pi*3600,'r','LineWidth',3);
axis normal;
grid on; grid minor;
xlabel('\textbf{Time (min)}','Interpreter','latex','FontSize',33)
ylabel('\textbf{Residual (arc secs)}','Interpreter','latex','FontSize',33)
set(gca,'FontSize',32,'FontWeight','bold')
h = legend('\boldmath$\alpha$','\boldmath$\beta$','Location','SouthWest');
set(h,'Interpreter','latex','FontSize',32);
saveas(gcf,strcat(codeDir,'\Cases\',sprintf('%s',input.runCase),'\Fig30.fig'))

% Plot true range errors
figure(31)
clf;
semilogy(t(:,1)/60,drho*1000,'-b','LineWidth',3);
hold on
semilogy(t(:,1)/60,drho2*1000,'-r','LineWidth',3);
axis normal;
grid on;
xlabel('\textbf{Time (min)}','Interpreter','latex','FontSize',33)
ylabel('\textbf{Range Error (m)}','Interpreter','latex','FontSize',33)
set(gca,'FontSize',32,'FontWeight','bold')
h = legend('\textbf{NLS Estimate}','\textbf{IOD Estimate}','Location', ...
    'SouthWest');
set(h,'Interpreter','latex','FontSize',32);
saveas(gcf,strcat(codeDir,'\Cases\',sprintf('%s',input.runCase),'\Fig31.fig'))
```

```
% Plot true range errors
figure(32)
clf;
semilogy(t(:,1)/60,drho./rho_true*100,'-b','LineWidth',3);
hold on
semilogy(t(:,1)/60,drho2./rho_true*100,'-r','LineWidth',3);
axis normal;
grid on;
xlabel('\textbf{Time (min)}','Interpreter','latex','FontSize',33)
ylabel('\textbf{Range Error (\%)}','Interpreter','latex','FontSize',33)
set(gca,'FontSize',32,'FontWeight','bold')
h = legend('\textbf{NLS Estimate}','\textbf{IOD Estimate}','Location', ...
    'SouthWest');
set(h,'Interpreter','latex','FontSize',32);
saveas(gcf,strcat(codeDir,'\Cases\',sprintf('%s',input.runCase),'\Fig32.fig'))

% Plot relative coordinate errors
figure(33)
clf;
plot(t(:,1)/60,(Xref_rel(1,:)'-Target.Xrel(:,1))*1000,'-r','LineWidth',3);
hold on
plot(t(:,1)/60,(Xref_rel(2,:)'-Target.Yrel(:,1))*1000,'-g','LineWidth',3);
plot(t(:,1)/60,(Xref_rel(3,:)'-Target.Zrel(:,1))*1000,'-b','LineWidth',3);
plot(t(:,1)/60,(rho_true - rho_est)*1000,'-k','LineWidth',3);
axis normal;
grid on;
xlabel('\textbf{Time (min)}','Interpreter','latex','FontSize',33)
ylabel('\textbf{Error (m)}','Interpreter','latex','FontSize',33)
set(gca,'FontSize',32,'FontWeight','bold')
h = legend('\textbf{R}','\textbf{S}','\textbf{W}','\boldmath$\rho$', ...
    'Location','NorthEast');
set(h,'Interpreter','latex','FontSize',32);
saveas(gcf,strcat(codeDir,'\Cases\',sprintf('%s',input.runCase),'\Fig33.fig'))

save(strcat(codeDir,'\Cases\',sprintf('%s_nls.mat',input.runCase)))
```

E.10 Measurement Sensitivity Matrix

```
%  [function]    dG_dX
%
%  [purpose]:    Function generates the 3x6 "H" matrix
%                Hi(ti)=dG_dX(Xref(ti),ti) where G is the observation
%                relationship between the system state at the current time and
%                the observation geometry.
%
%  [useage]:     [H]     = dG_dX(X,theta,R,Xrsw_S)
%
%  [inputs]:     X       = Xref_ecef(1:3,1)
%                theta   = GMST
%                R       = transformation matrix from ECI to RSW
%                Xrsw_S = Surveyor's RSW vector
%
%  [outputs]:    H       = [da/dx, da/dy, da/dz, da/dvx, da/dvy, da/dvz;
%                           db/dx, db/dy, db/dz, db/dvx, db/dvy, db/dvz;
%                           dr/dx, dr/dy, dr/dz, dr/dvx, dr/dvy, dr/dvz];
%
%                where  a = alpha = atan(y/x)
%                       b = beta = atan(z/sqrt(x^2+y^2))
%                       r = rho = sqrt(x^2+y^2+z^2)
%
%  Max Yates, February 28, 2017

function [H] = dG_dX(X,theta,R,Xrsw_S)

x = X(1,1); y = X(2,1); z = X(3,1);
R11 = R(1,1); R12 = R(1,2); R13 = R(1,3);
R21 = R(2,1); R22 = R(2,2); R23 = R(2,3);
R31 = R(3,1); R32 = R(3,2); R33 = R(3,3);
xrsw_S = Xrsw_S(1,1); yrsw_S = Xrsw_S(2,1); zrsw_S = Xrsw_S(3,1);

% Preallocate matrix
H = zeros(3,6);

% Partial G with respect to Xecef
% da/dx
H(1,1) = ((R21*cos(theta) + R22*sin(theta))/(R13*z - xrsw_S + ...
    R11*(x*cos(theta) - y*sin(theta)) + R12*(y*cos(theta) + ...
    x*sin(theta))) - ((R11*cos(theta) + R12*sin(theta))*(R23*z - ...
    yrsw_S + R21*(x*cos(theta) - y*sin(theta)) + R22*(y*cos(theta) + ...
    x*sin(theta))))/(R13*z - xrsw_S + R11*(x*cos(theta) - ...
    y*sin(theta)) + R12*(y*cos(theta) + x*sin(theta)))^2)/((R23*z - ...
    yrsw_S + R21*(x*cos(theta) - y*sin(theta)) + R22*(y*cos(theta) + ...
    x*sin(theta)))^2/(R13*z - xrsw_S + R11*(x*cos(theta) - ...
    y*sin(theta)) + R12*(y*cos(theta) + x*sin(theta)))^2 + 1);

%da/dy
H(1,2) = ((R22*cos(theta) - R21*sin(theta))/(R13*z - xrsw_S + ...
    R11*(x*cos(theta) - y*sin(theta)) + R12*(y*cos(theta) + ...
    x*sin(theta))) - ((R12*cos(theta) - R11*sin(theta))*(R23*z - ...
    yrsw_S + R21*(x*cos(theta) - y*sin(theta)) + R22*(y*cos(theta) + ...
    x*sin(theta))))/(R13*z - xrsw_S + R11*(x*cos(theta) - ...
    y*sin(theta)) + R12*(y*cos(theta) + x*sin(theta)))^2)/((R23*z - ...
    yrsw_S + R21*(x*cos(theta) - y*sin(theta)) + R22*(y*cos(theta) + ...
    x*sin(theta)))^2/(R13*z - xrsw_S + R11*(x*cos(theta) - ...
    y*sin(theta)) + R12*(y*cos(theta) + x*sin(theta)))^2 + 1);

%da/dz
H(1,3) = (R23/(R13*z - xrsw_S + R11*(x*cos(theta) - y*sin(theta)) + ...
    R12*(y*cos(theta) + x*sin(theta))) - (R13*(R23*z - yrsw_S + ...
    R21*(x*cos(theta) - y*sin(theta)) + R22*(y*cos(theta) + ...
    x*sin(theta))))/(R13*z - xrsw_S + R11*(x*cos(theta) - ...
    y*sin(theta)) + R12*(y*cos(theta) + x*sin(theta)))^2)/((R23*z - ...
    yrsw_S + R21*(x*cos(theta) - y*sin(theta)) + R22*(y*cos(theta) + ...
    x*sin(theta)))^2/(R13*z - xrsw_S + R11*(x*cos(theta) - ...
    y*sin(theta)) + R12*(y*cos(theta) + x*sin(theta)))^2 + 1);

%db/dx
H(2,1) = ((R31*cos(theta) + R32*sin(theta))/((R13*z - xrsw_S + ...
    R11*(x*cos(theta) - y*sin(theta)) + R12*(y*cos(theta) + ...
    x*sin(theta)))^2 + (R23*z - yrsw_S + R21*(x*cos(theta) - ...
    y*sin(theta)) + R22*(y*cos(theta) + x*sin(theta)))^2)^(1/2) - ...
    ((2*(R11*cos(theta) + R12*sin(theta))*(R13*z - xrsw_S + ...
    R11*(x*cos(theta) - y*sin(theta)) + R12*(y*cos(theta) + ...
    x*sin(theta))) + 2*(R21*cos(theta) + R22*sin(theta))*(R23*z - ...
    yrsw_S + R21*(x*cos(theta) - y*sin(theta)) + R22*(y*cos(theta) + ...
    x*sin(theta))))*(R33*z - zrsw_S + R31*(x*cos(theta) - ...
    y*sin(theta)) + R32*(y*cos(theta) + x*sin(theta))))/(2*((R13*z - ...
    xrsw_S + R11*(x*cos(theta) - y*sin(theta)) + R12*(y*cos(theta) + ...
    x*sin(theta)))^2 + (R23*z - yrsw_S + R21*(x*cos(theta) - ...
    y*sin(theta)) + R22*(y*cos(theta) + x*sin(theta)))^2)^(3/2)))/...
    ((R33*z - zrsw_S + R31*(x*cos(theta) - y*sin(theta)) + ...
    R32*(y*cos(theta) + x*sin(theta)))^2/((R13*z - xrsw_S + ...
    R11*(x*cos(theta) - y*sin(theta)) + R12*(y*cos(theta) + ...
    x*sin(theta)))^2 + (R23*z - yrsw_S + R21*(x*cos(theta) - ...
    y*sin(theta)) + R22*(y*cos(theta) + x*sin(theta)))^2) + 1);

%db/dy
H(2,2) = ((R32*cos(theta) - R31*sin(theta))/((R13*z - xrsw_S + ...
    R11*(x*cos(theta) - y*sin(theta)) + R12*(y*cos(theta) + ...
    x*sin(theta)))^2 + (R23*z - yrsw_S + R21*(x*cos(theta) - ...
    y*sin(theta)) + R22*(y*cos(theta) + x*sin(theta)))^2)^(1/2) - ...
    ((2*(R12*cos(theta) - R11*sin(theta))*(R13*z - xrsw_S + ...
    R11*(x*cos(theta) - y*sin(theta)) + R12*(y*cos(theta) + ...
    x*sin(theta))) + 2*(R22*cos(theta) - R21*sin(theta))*(R23*z - ...
    yrsw_S + R21*(x*cos(theta) - y*sin(theta)) + R22*(y*cos(theta) + ...
    x*sin(theta))))*(R33*z - zrsw_S + R31*(x*cos(theta) - ...
    y*sin(theta)) + R32*(y*cos(theta) + x*sin(theta))))/(2*((R13*z - ...
    xrsw_S + R11*(x*cos(theta) - y*sin(theta)) + R12*(y*cos(theta) + ...
    x*sin(theta)))^2 + (R23*z - yrsw_S + R21*(x*cos(theta) - ...
    y*sin(theta)) + R22*(y*cos(theta) + x*sin(theta)))^2)^(3/2)))/...
    ((R33*z - zrsw_S + R31*(x*cos(theta) - y*sin(theta)) + ...
    R32*(y*cos(theta) + x*sin(theta)))^2/((R13*z - xrsw_S + ...
    R11*(x*cos(theta) - y*sin(theta)) + R12*(y*cos(theta) + ...
    x*sin(theta)))^2 + (R23*z - yrsw_S + R21*(x*cos(theta) - ...
    y*sin(theta)) + R22*(y*cos(theta) + x*sin(theta)))^2) + 1);

%db/dz
```

```
H(2,3) = (R33/((R13*z - xrsw_S + R11*(x*cos(theta) - y*sin(theta)) + ...
    R12*(y*cos(theta) + x*sin(theta)))^2 + (R23*z - yrsw_S + ...
    R21*(x*cos(theta) - y*sin(theta)) + R22*(y*cos(theta) + ...
    x*sin(theta)))^2)^(1/2) - ((2*R13*(R13*z - xrsw_S + ...
    R11*(x*cos(theta) - y*sin(theta)) + R12*(y*cos(theta) + ...
    x*sin(theta))) + 2*R23*(R23*z - yrsw_S + R21*(x*cos(theta) - ...
    y*sin(theta)) + R22*(y*cos(theta) + x*sin(theta))))*(R33*z - ...
    zrsw_S + R31*(x*cos(theta) - y*sin(theta)) + R32*(y*cos(theta) + ...
    x*sin(theta))))/(2*((R13*z - xrsw_S + R11*(x*cos(theta) - ...
    y*sin(theta)) + R12*(y*cos(theta) + x*sin(theta)))^2 + (R23*z - ...
    yrsw_S + R21*(x*cos(theta) - y*sin(theta)) + R22*(y*cos(theta) + ...
    x*sin(theta)))^2)^(3/2)))/((R33*z - zrsw_S + R31*(x*cos(theta) - ...
    y*sin(theta)) + R32*(y*cos(theta) + x*sin(theta)))^2/((R13*z - ...
    xrsw_S + R11*(x*cos(theta) - y*sin(theta)) + R12*(y*cos(theta) + ...
    x*sin(theta)))^2 + (R23*z - yrsw_S + R21*(x*cos(theta) - ...
    y*sin(theta)) + R22*(y*cos(theta) + x*sin(theta)))^2) + 1);

%dr/dx
H(3,1) = (2*(R11*cos(theta) + R12*sin(theta))*(R13*z - xrsw_S + ...
    R11*(x*cos(theta) - y*sin(theta)) + R12*(y*cos(theta) + ...
    x*sin(theta))) + 2*(R21*cos(theta) + R22*sin(theta))*(R23*z - ...
    yrsw_S + R21*(x*cos(theta) - y*sin(theta)) + R22*(y*cos(theta) + ...
    x*sin(theta))) + 2*(R31*cos(theta) + R32*sin(theta))*(R33*z - ...
    zrsw_S + R31*(x*cos(theta) - y*sin(theta)) + R32*(y*cos(theta) + ...
    x*sin(theta))))/(2*((R13*z - xrsw_S + R11*(x*cos(theta) - ...
    y*sin(theta)) + R12*(y*cos(theta) + x*sin(theta)))^2 + (R23*z - ...
    yrsw_S + R21*(x*cos(theta) - y*sin(theta)) + R22*(y*cos(theta) + ...
    x*sin(theta)))^2 + (R33*z - zrsw_S + R31*(x*cos(theta) - ...
    y*sin(theta)) + R32*(y*cos(theta) + x*sin(theta)))^2)^(1/2));

%dr/dy
H(3,2) = (2*(R12*cos(theta) - R11*sin(theta))*(R13*z - xrsw_S + ...
    R11*(x*cos(theta) - y*sin(theta)) + R12*(y*cos(theta) + ...
    x*sin(theta))) + 2*(R22*cos(theta) - R21*sin(theta))*(R23*z - ...
    yrsw_S + R21*(x*cos(theta) - y*sin(theta)) + R22*(y*cos(theta) + ...
    x*sin(theta))) + 2*(R32*cos(theta) - R31*sin(theta))*(R33*z - ...
    zrsw_S + R31*(x*cos(theta) - y*sin(theta)) + R32*(y*cos(theta) + ...
    x*sin(theta))))/(2*((R13*z - xrsw_S + R11*(x*cos(theta) - ...
    y*sin(theta)) + R12*(y*cos(theta) + x*sin(theta)))^2 + (R23*z - ...
    yrsw_S + R21*(x*cos(theta) - y*sin(theta)) + R22*(y*cos(theta) + ...
    x*sin(theta)))^2 + (R33*z - zrsw_S + R31*(x*cos(theta) - ...
    y*sin(theta)) + R32*(y*cos(theta) + x*sin(theta)))^2)^(1/2));

%dr/dz
H(3,3) = (2*R13*(R13*z - xrsw_S + R11*(x*cos(theta) - y*sin(theta)) + ...
    R12*(y*cos(theta) + x*sin(theta))) + 2*R23*(R23*z - yrsw_S + ...
    R21*(x*cos(theta) - y*sin(theta)) + R22*(y*cos(theta) + ...
    x*sin(theta))) + 2*R33*(R33*z - zrsw_S + R31*(x*cos(theta) - ...
    y*sin(theta)) + R32*(y*cos(theta) + x*sin(theta))))/(2*((R13*z - ...
    xrsw_S + R11*(x*cos(theta) - y*sin(theta)) + R12*(y*cos(theta) + ...
    x*sin(theta)))^2 + (R23*z - yrsw_S + R21*(x*cos(theta) - ...
    y*sin(theta)) + R22*(y*cos(theta) + x*sin(theta)))^2 + (R33*z - ...
    zrsw_S + R31*(x*cos(theta) - y*sin(theta)) + R32*(y*cos(theta) + ...
    x*sin(theta)))^2)^(1/2));
```

E.11 Equations of Variation A Matrix

```
%  [function]    Amatrix
%
%  [purpose]:    Calculates the Hamiltonian and its partials for
%                constructing the A matrix.
%
%  [useage]:     [a_ECI,A,Ham] = Amatrix(r_ECEF,v_ECEF,utcmjd,method)
%
%  [inputs]:     r_ECEF = 3x1 array of cartesian coordinates in earth
%                         centered rotating (ECEF) frame. Currently setup
%                         to accept units of m.
%
%                v_ECEF = 3x1 array of cartesian velocities in earth
%                         centered rotating (ECEF) frame. Currently setup
%                         to accept units of m/sec.
%
%                utcmjd = UTC Modified Julian date (from GMAT)
%
%                method = '2body', 'direct' or 'numerical'
%
%  [outputs]:    A      = A matrix
%
%                Ham    = Hamiltonian
%
%                a_ECI  = 3x1 array of cartesian acceleration in earth
%                         centered inertial (ECI) frame. m/sec^2.
%
%  [reference]:
%
%  [notes]:      gravity function is dictating units for r_ECEF & v_ECEF
%
%  Max Yates, February 12, 2017

function [A,Ham,a_ECI] = Amatrix(r_ECEF,v_ECEF,utcmjd,method)

global DU TU Omegae omegae geo_degree deltaJD

mu = 3986004.415E+8; % [m^3/s^2] EGM96
R = 6378136.3; % [m] EGM96

JD = utcmjd + 2430000.0 + deltaJD; % Julian Date
nm = geo_degree;
x = r_ECEF(1,1); y = r_ECEF(2,1); z = r_ECEF(3,1); % [m]
r = sqrt(x^2 + y^2 + z^2); % [m]

a_ECI = NaN; % pre-assign (not output by all methods)
Ham = NaN;

%% First & Second Order Partial Derivatives of r, lambda (longitude), phi
%% (geocentric latitude), and theta (colatitude)
% dr/dx
drdx = x/r;
% d2r/dx2
d2rdx2 = (y^2 + z^2)/(r^3);
```

```
% d2r/dxdy
d2rdxdy = -x*y/(r^3);
% d2r/dxdz
d2rdxdz = -x*z/(r^3);
% dr/dy
drdy = y/r;
% d2r/dy2
d2rdy2 = (x^2 + z^2)/(r^3);
% d2r/dydx
d2rdydx = d2rdxdy;
% d2r/dydz
d2rdydz = -y*z/(r^3);
% dr/dz
drdz = z/r;
% d2r/dz2
d2rdz2 = (x^2 + y^2)/(r^3);
% d2r/dzdx
d2rdzdx = d2rdxdz;
% d2r/dzdy
d2rdzdy = d2rdydz;

% dlambda/dx
dldx = -y/(x^2 + y^2);
% d2lambda/dx2
d2ldx2 = 2*x*y/(x^2 + y^2)^2;
% d2lambda/dxdy
d2ldxdy = (y^2 - x^2)/(x^2 + y^2)^2;
% d2lambda/dxdz
d2ldxdz = 0;
% dlambda/dy
dldy = x/(x^2 + y^2);
% d2lambda/dy2
d2ldy2 = -2*x*y/(x^2 + y^2)^2;
% d2lambda/dydx
d2ldydx = d2ldxdy;
% d2lambda/dydz
d2ldydz = 0;
% dlambda/dz
dldz = 0;
% d2lambda/dz2
d2ldz2 = 0;
% d2lambda/dzdx
d2ldzdx = 0;
% d2lambda/dzdy
d2ldzdy = 0;

% dphi/dx
dfdx = -(x*z)/((r^3)*sqrt(1 - (z^2)/(r^2)));
% d2phi/dx2
d2fdx2 = (x^2)*(z^3)/((r^7)*(1 - (z^2)/(r^2))^(3/2)) + ...
    (3*(x^2)*z)/((r^5)*sqrt(1 - (z^2)/(r^2))) - z/((r^3)*sqrt(1 - ...
    (z^2)/(r^2)));
% d2phi/dxdy
d2fdxdy = x*y*(z^3)/((r^7)*(1 - (z^2)/(r^2))^(3/2)) + ...
    3*x*y*z/((r^5)*sqrt(1 - (z^2)/(r^2)));
% d2phi/dxdz
```

```
d2fdxdz = x*z*(2*(z^3)/(r^4) - 2*z/(r^2))/(2*(r^3)*(1 - ...
    (z^2)/(r^2))^(3/2)) + 3*x*(z^2)/((r^5)*sqrt(1 - (z^2)/(r^2))) - ...
    x/((r^3)*sqrt(1 - (z^2)/(r^2)));
% dphi/dy
dfdy = -(y*z)/((r^3)*sqrt(1 - (z^2)/(r^2)));
% d2phi/dy2
d2fdy2 = (y^2)*(z^3)/((r^7)*(1 - (z^2)/(r^2))^(3/2)) + ...
    (3*(y^2)*z)/((r^5)*sqrt(1 - (z^2)/(r^2))) - z/((r^3)*sqrt(1 - ...
    (z^2)/(r^2)));
% d2phi/dydx
d2fdydx = d2fdxdy;
% d2phi/dydz
d2fdydz = y*z*(2*(z^3)/(r^4) - 2*z/(r^2))/(2*(r^3)*(1 - ...
    (z^2)/(r^2))^(3/2)) + 3*y*(z^2)/((r^5)*sqrt(1 - (z^2)/(r^2))) - ...
    y/((r^3)*sqrt(1 - (z^2)/(r^2)));
% dphi/dz
dfdz = (1/r - (z^2)/(r^3))/sqrt(1 - (z^2)/(r^2));
% d2phi/dz2
d2fdz2 = ((3*z^3)/(r^5) - (3*z)/(r^3))/sqrt(1 - (z^2)/(r^2)) - ...
    (((2*z^3)/(r^4) - (2*z)/(r^2))*(1/r - (z^2)/(r^3)))/(2*(1 - ...
    (z^2)/(r^2))^(3/2));

% dtheta/dx
dtdx = (x*z)/(sqrt(1 - (z^2)/(r^2))*(r^3));
% dtheta/dy
dtdy = (y*z)/(sqrt(1 - (z^2)/(r^2))*(r^3));
% dtheta/dz
dtdz = -(1/r - (z^2)/(r^3))/sqrt(1 - (z^2)/(r^2));

if strcmp(method,'2body')
    H_px_x = 0;
    H_px_y = Omegae;
    H_px_z = 0;
    H_px_px = 1;
    H_px_py = 0;
    H_px_pz = 0;
    H_py_x = -Omegae;
    H_py_y = 0;
    H_py_z = 0;
    H_py_px = 0;
    H_py_py = 1;
    H_py_pz = 0;
    H_pz_x = 0;
    H_pz_y = 0;
    H_pz_z = 0;
    H_pz_px = 0;
    H_pz_py = 0;
    H_pz_pz = 1;
    H_x_x = mu/(x^2 + y^2 + z^2)^(3/2) - (3*mu*x^2)/(x^2 + y^2 + z^2)^(5/2);
    H_x_y = -(3*mu*x*y)/(x^2 + y^2 + z^2)^(5/2);
    H_x_z = -(3*mu*x*z)/(x^2 + y^2 + z^2)^(5/2);
    H_x_px = 0;
    H_x_py = -Omegae;
    H_x_pz = 0;
    H_y_x = -(3*mu*x*y)/(x^2 + y^2 + z^2)^(5/2);
    H_y_y = mu/(x^2 + y^2 + z^2)^(3/2) - (3*mu*y^2)/(x^2 + y^2 + z^2)^(5/2);
    H_y_z = -(3*mu*y*z)/(x^2 + y^2 + z^2)^(5/2);
    H_y_px = Omegae;
    H_y_py = 0;
    H_y_pz = 0;
    H_z_x = -(3*mu*x*z)/(x^2 + y^2 + z^2)^(5/2);
    H_z_y = -(3*mu*y*z)/(x^2 + y^2 + z^2)^(5/2);
    H_z_z = mu/(x^2 + y^2 + z^2)^(3/2) - (3*mu*z^2)/(x^2 + y^2 + z^2)^(5/2);
    H_z_px = 0;
    H_z_py = 0;
    H_z_pz = 0;

    A = [H_px_x, H_px_y, H_px_z, H_px_px, H_px_py, H_px_pz;
         H_py_x, H_py_y, H_py_z, H_py_px, H_py_py, H_py_pz;
         H_pz_x, H_pz_y, H_pz_z, H_pz_px, H_pz_py, H_pz_pz;
         -H_x_x, -H_x_y, -H_x_z, -H_x_px, -H_x_py, -H_x_pz;
         -H_y_x, -H_y_y, -H_y_z, -H_y_px, -H_y_py, -H_y_pz;
         -H_z_x, -H_z_y, -H_z_z, -H_z_px, -H_z_py, -H_z_pz];
end

if strcmp(method,'direct')
    %% GrafLab
    % Latitudes must be entered within the interval <-90°,90°> and longitudes
    % in the range <0°,360°> or <-180°,180°>
    lat = asin(z/sqrt(x^2 + y^2 + z^2))*180/pi; % geocentric latitude [deg]
    lon = atan2(y,x)*180/pi; % [deg]
    [P]=GrafLab('OK',mu,R,0,geo_degree,2,'EGM96.mat',1,2,[],[],[],[],[], ...
        [],[],[],lat,lon,r,'my_output',0,[11;12;13;16],1,[],0,0,0,[],[], ...
        [],[],[],0);

    %% Gravitational Second-Order Partials in Spherical Coordinates
    V1 = P(1); % gravitational potential
    V_rr = P(2)*1e-9;
    V_ff = P(3)*1e-9;
    V_ll = P(4)*1e-9;
    V_rf = P(5)*1e-9; V_fr = V_rf;
    V_rl = P(6)*1e-9; V_lr = V_rl;
    V_fl = P(7)*1e-9; V_lf = V_fl;

    % Tensor corrections. See page 2 in Bucha's functional definitions
    % document.
    V_rf = r*V_rf; V_fr = V_rf;
    V_rl = r*cosd(lat)*V_rl; V_lr = V_rl;
    V_ff = r*r*V_ff;
    V_fl = r*r*cosd(lat)*V_fl; V_lf = V_fl;
    V_ll = r*r*cosd(lat)*cosd(lat)*V_ll;

    %% Gravity Gradients
    % Gravity vector in the local north-oriented reference frame
    gx = P(8);
    gy = P(9);
    gz = P(10);

    % Corrections see page 6 in Bucha's functional definitions document.
    dVcdr = (Omegae^2)*r*(cosd(lat)^2)*1e5;
    dVcdf = -(Omegae^2)*(r^2)*cosd(lat)*sind(lat)*1e5;
    dVcdl = 0;
```

```
vr = (gz - dVcdr)*1e-5;
vl = (gy*(-r*cosd(lat)) - dVcdl)*1e-5;
vf = (gx*r - dVcdf)*1e-5;

%% Calculate Gravitational Second-Order Partials in Cartesian Coordinates
%% with Multivariable Calculus
d2Vdxdx = V_rr*(drdx^2) + V_ll*(dldx^2) + V_ff*(dfdx^2) + ...
    2*V_rl*dldx*drdx + 2*V_rf*dfdx*drdx + 2*V_fl*dfdx*dldx + ...
    vr*d2rdx2 + vl*d2ldx2 + vf*d2fdx2;
d2Vdydy = V_rr*(drdy^2) + V_ll*(dldy^2) + V_ff*(dfdy^2) + ...
    2*V_rl*dldy*drdy + 2*V_rf*dfdy*drdy + 2*V_fl*dfdy*dldy + ...
    vr*d2rdy2 + vl*d2ldy2 + vf*d2fdy2;
d2Vdzdz = V_rr*(drdz^2) + V_ll*(dldz^2) + V_ff*(dfdz^2) + ...
    2*V_rl*dldz*drdz + 2*V_rf*dfdz*drdz + 2*V_fl*dfdz*dldz + ...
    vr*d2rdz2 + vl*d2ldz2 + vf*d2fdz2;
d2Vdxdy = V_rr*drdx*drdy + V_lr*dldx*drdy + V_fr*dfdx*drdy + ...
    V_rl*drdx*dldy + V_ll*dldx*dldy + V_fl*dfdx*dldy + ...
    V_rf*drdx*dfdy + V_lf*dldx*dfdy + V_ff*dfdx*dfdy + ...
    vr*d2rdxdy + vl*d2ldxdy + vf*d2fdxdy;
d2Vdydx = d2Vdxdy;
d2Vdxdz = V_rr*drdx*drdz + V_lr*dldx*drdz + V_fr*dfdx*drdz + ...
    V_rl*drdx*dldz + V_ll*dldx*dldz + V_fl*dfdx*dldz + ...
    V_rf*drdx*dfdz + V_lf*dldx*dfdz + V_ff*dfdx*dfdz + ...
    vr*d2rdxdz + vl*d2ldxdz + vf*d2fdxdz;
d2Vdzdx = d2Vdxdz;
d2Vdydz = V_rr*drdy*drdz + V_lr*dldy*drdz + V_fr*dfdy*drdz + ...
    V_rl*drdy*dldz + V_ll*dldy*dldz + V_fl*dfdy*dldz + ...
    V_rf*drdy*dfdz + V_lf*dldy*dfdz + V_ff*dfdy*dfdz + ...
    vr*d2rdydz + vl*d2ldydz + vf*d2fdydz;
d2Vdzdy = d2Vdydz;

d2HdX2 = [-d2Vdxdx, -d2Vdxdy, -d2Vdxdz,      0, -Omegae, 0;
          -d2Vdydx, -d2Vdydy, -d2Vdydz, Omegae,       0, 0;
          -d2Vdzdx, -d2Vdzdy, -d2Vdzdz,      0,       0, 0;
                 0,   Omegae,        0,      1,       0, 0;
           -Omegae,        0,        0,      0,       1, 0;
                 0,        0,        0,      0,       0, 1];
% Symplectic group matrix
Z = [zeros(3,3),     eye(3);
        -eye(3), zeros(3,3)];

A = Z*d2HdX2;

% Check conservation of the Hamiltonian in canonical units
x = r_ECEF(1,1)/1000/DU; % [DU]
y = r_ECEF(2,1)/1000/DU; % [DU]
z = r_ECEF(3,1)/1000/DU; % [DU]
vx = v_ECEF(1,1)/1000/DU*60*TU; % [DU/TU]
vy = v_ECEF(2,1)/1000/DU*60*TU; % [DU/TU]
vz = v_ECEF(3,1)/1000/DU*60*TU; % [DU/TU]
% Momenta
px = vx-y*omegae;
py = vy+x*omegae;
pz = vz;
% Hamiltonian
```

```
    Ham = 0.5*(px*px+py*py+pz*pz) + omegae*(y*px-x*py) - ...
        V1/(3986004.415E+8/6378136.3);
end

if strcmp(method,'numerical')
    %% Calculate Gravity Gradients
    [a_ECI,Vgrav,vl,vt,vr,a_ECEF] = gravity([r_ECEF;v_ECEF],nm,JD);

    % Multivariable Chain Rule
    dVdx = vr*drdx + vl*dldx + vt*dtdx;
    dVdy = vr*drdy + vl*dldy + vt*dtdy;
    dVdz = vr*drdz + vl*dldz + vt*dtdz;

    %% Numerically calculate gravitational second-order partial derivatives
    %% in Cartesian coordinates
    % Perturb x by dx
    dx = 0.00001*DU*1000; % [m]
    X = [r_ECEF;v_ECEF]+[dx;0;0;0;0;0];
    [a_ECI2,Vgrav2,vl2,vt2,vr2,a_ECEF2] = gravity(X,nm,JD);
    r = sqrt(X(1,1)^2 + X(2,1)^2 + X(3,1)^2);
    % dr/dx
    drdx_pert = X(1,1)/r;
    % dr/dy
    drdy_pert = X(2,1)/r;
    % dr/dz
    drdz_pert = X(3,1)/r;
    % dlambda/dx
    dldx_pert = -X(2,1)/(X(1,1)^2 + X(2,1)^2);
    % dlambda/dy
    dldy_pert = X(1,1)/(X(1,1)^2 + X(2,1)^2);
    % dlambda/dz
    dldz_pert = 0;
    % dtheta/dx
    dtdx_pert = (X(1,1)*X(3,1))/(sqrt(1 - (X(3,1)^2)/(r^2))*(r^3));
    % dtheta/dy
    dtdy_pert = (X(2,1)*X(3,1))/(sqrt(1 - (X(3,1)^2)/(r^2))*(r^3));
    % dtheta/dz
    dtdz_pert = -(1/r - (X(3,1)^2)/(r^3))/sqrt(1 - (X(3,1)^2)/(r^2));
    % Multivariable Chain Rule
    dVdx2 = vr2*drdx_pert + vl2*dldx_pert + vt2*dtdx_pert;
    dVdy2 = vr2*drdy_pert + vl2*dldy_pert + vt2*dtdy_pert;
    dVdz2 = vr2*drdz_pert + vl2*dldz_pert + vt2*dtdz_pert;

    d2Vdxdx_num = (dVdx2-dVdx)/dx;
    d2Vdydx_num = (dVdy2-dVdy)/dx;
    d2Vdzdx_num = (dVdz2-dVdz)/dx;

    % Perturb y by dy
    dy = 0.00001*DU*1000; % [m]
    X = [r_ECEF;v_ECEF]+[0;dy;0;0;0;0];
    [a_ECI2,Vgrav2,vl2,vt2,vr2,a_ECEF2] = gravity(X,nm,JD);
    r = sqrt(X(1,1)^2 + X(2,1)^2 + X(3,1)^2);
    % dr/dx
    drdx_pert = X(1,1)/r;
    % dr/dy
    drdy_pert = X(2,1)/r;
```

```
% dr/dz
drdz_pert = X(3,1)/r;
% dlambda/dx
dldx_pert = -X(2,1)/(X(1,1)^2 + X(2,1)^2);
% dlambda/dy
dldy_pert = X(1,1)/(X(1,1)^2 + X(2,1)^2);
% dlambda/dz
dldz_pert = 0;
% dtheta/dx
dtdx_pert = (X(1,1)*X(3,1))/(sqrt(1 - (X(3,1)^2)/(r^2))*(r^3));
% dtheta/dy
dtdy_pert = (X(2,1)*X(3,1))/(sqrt(1 - (X(3,1)^2)/(r^2))*(r^3));
% dtheta/dz
dtdz_pert = -(1/r - (X(3,1)^2)/(r^3))/sqrt(1 - (X(3,1)^2)/(r^2));
% Multivariable Chain Rule
dVdx2 = vr2*drdx_pert + vl2*dldx_pert + vt2*dtdx_pert;
dVdy2 = vr2*drdy_pert + vl2*dldy_pert + vt2*dtdy_pert;
dVdz2 = vr2*drdz_pert + vl2*dldz_pert + vt2*dtdz_pert;

d2Vdxdy_num = (dVdx2-dVdx)/dy;
d2Vdydy_num = (dVdy2-dVdy)/dy;
d2Vdzdy_num = (dVdz2-dVdz)/dy;

% Perturb z by dz
dz = 0.00001*DU*1000; % [m]
X = [r_ECEF;v_ECEF]+[0;0;dz;0;0;0];
[a_ECI2,Vgrav2,vl2,vt2,vr2,a_ECEF2] = gravity(X,nm,JD);
r = sqrt(X(1,1)^2 + X(2,1)^2 + X(3,1)^2);
% dr/dx
drdx_pert = X(1,1)/r;
% dr/dy
drdy_pert = X(2,1)/r;
% dr/dz
drdz_pert = X(3,1)/r;
% dlambda/dx
dldx_pert = -X(2,1)/(X(1,1)^2 + X(2,1)^2);
% dlambda/dy
dldy_pert = X(1,1)/(X(1,1)^2 + X(2,1)^2);
% dlambda/dz
dldz_pert = 0;
% dtheta/dx
dtdx_pert = (X(1,1)*X(3,1))/(sqrt(1 - (X(3,1)^2)/(r^2))*(r^3));
% dtheta/dy
dtdy_pert = (X(2,1)*X(3,1))/(sqrt(1 - (X(3,1)^2)/(r^2))*(r^3));
% dtheta/dz
dtdz_pert = -(1/r - (X(3,1)^2)/(r^3))/sqrt(1 - (X(3,1)^2)/(r^2));
% Multivariable Chain Rule
dVdx2 = vr2*drdx_pert + vl2*dldx_pert + vt2*dtdx_pert;
dVdy2 = vr2*drdy_pert + vl2*dldy_pert + vt2*dtdy_pert;
dVdz2 = vr2*drdz_pert + vl2*dldz_pert + vt2*dtdz_pert;

d2Vdxdz_num = (dVdx2-dVdx)/dz;
d2Vdydz_num = (dVdy2-dVdy)/dz;
d2Vdzdz_num = (dVdz2-dVdz)/dz;

d2HdX2_num = [-d2Vdxdx_num, -d2Vdxdy_num, -d2Vdxdz_num,     0,-Omegae, 0;
```

```
                  -d2Vdydx_num, -d2Vdydy_num, -d2Vdydz_num,Omegae,      0, 0;
                  -d2Vdzdx_num, -d2Vdzdy_num, -d2Vdzdz_num,     0,      0, 0;
                             0,       Omegae,            0,     1,      0, 0;
                       -Omegae,            0,            0,     0,      1, 0;
                             0,            0,            0,     0,      0, 1];
    % Symplectic group matrix
    Z = [zeros(3,3),     eye(3);
            -eye(3), zeros(3,3)];

    A = Z*d2HdX2_num;

    % Check conservation of the Hamiltonian in canonical units
    x = r_ECEF(1,1)/1000/DU; % [DU]
    y = r_ECEF(2,1)/1000/DU; % [DU]
    z = r_ECEF(3,1)/1000/DU; % [DU]
    vx = v_ECEF(1,1)/1000/DU*60*TU; % [DU/TU]
    vy = v_ECEF(2,1)/1000/DU*60*TU; % [DU/TU]
    vz = v_ECEF(3,1)/1000/DU*60*TU; % [DU/TU]
    % Momenta
    px = vx-y*omegae;
    py = vy+x*omegae;
    pz = vz;
    % Hamiltonian
    Ham = 0.5*(px*px+py*py+pz*pz) + omegae*(y*px-x*py) - ...
        Vgrav/(3986004.415E+8/6378136.3);
end
```

BIOGRAPHY

UNITED STATES AIR FORCE

MAJOR MAX W. YATES, PhD

Major Max W. Yates is a student at the United States Naval War College, Newport, RI. He previously served as Program Manager for Rapid Acquisitions within the Space Vehicles Directorate, Air Force Research Laboratory (AFRL), Kirtland Air Force Base, New Mexico, where he led a 79 member team in support of a $166 million dollar technology portfolio to respond to urgent warfighter needs. These included a space situational awareness program, a C-130 deployed UAV for Special Operators and a UAV missile.

Major Yates was born in Butte, Montana. He received a doctorate from the Department of Aeronautics and Astronautics at Massachusetts Institute of Technology in 2017, a master's degree in astronautical engineering from the Air Force Institute of Technology in 2011 and a dual bachelor of science degree in aeronautical and mechanical engineering from Rensselaer Polytechnic Institute in 2006 where he commissioned through the Reserve Officer Training Corps. Major Yates has served as a flight test engineer at the Air Force Test Center as well as NASA Armstrong Flight Research Center in support of Global Hawk and the Orion spacecraft, respectively. He was also Deputy Program Manager for AFRL's Strategic Systems and Launch Technologies group where he managed the development of advanced navigation and ballistic missile technology.

EDUCATION
2006 Dual Bachelor of Science degree in Aeronautical and Mechanical Engineering, Rensselaer Polytechnic Institute, Troy, NY
2006 Distinguished Graduate, Reserve Officer Training Corp, Rensselaer Polytechnic Institute, Troy, NY
2006 Top Performer (no DG program), Air and Space Basic Course, Maxwell Air Force Base, AL
2011 Master of Science degree in Astronautical Engineering, Air Force Institute of Technology (AFIT), Wright Patterson AFB, Ohio
2013 Distinguished Graduate, Squadron Officer School, Maxwell Air Force Base, AL
2017 Doctor of Philosophy degree in Astronautical Engineering, Massachusetts Institute of Technology, Cambridge, MA

ASSIGNMENTS
1. July 2006 – September 2007, Global Hawk Performance, Propulsion and Flying Qualities Engineer, 773rd Test Squadron, Edwards AFB, CA.

2. September 2007 – July 2008, Global Hawk Lead Modeling and Simulation Engineer, 773rd Test Squadron, Edwards AFB, CA.
3. July 2008 – August 2009, Orion Controls and Dynamics Engineer, NASA Armstrong Flight Research Center, CA.
4. August 2009 – April 2011, Student, Air Force Institute of Technology, Wright Patterson AFB, OH.
5. April 2011 – March 2013, Deputy Program Manager, Strategic Systems and Launch Technologies, Space Vehicles Directorate, Air Force Research Laboratory, Kirtland AFB, NM.
6. March 2013 – August 2014, Program Manager, Rapid Acquisitions, Space Vehicles Directorate, Air Force Research Laboratory, Kirtland AFB, NM.
7. August 2014 – July 2017, Student, Massachusetts Institute of Technology, Cambridge, MA.
8. July 2017 – present, Student, College of Naval Command and Staff, Naval War College, Newport, RI.

MAJOR AWARDS AND DECORATIONS

Air Force Meritorious Service Medal
Air Force Commendation Medal

2014	MIT-Lincoln Laboratory Military Fellowship
2013	Company Grade Officer of the Year, AFRL
2012	Company Grade Officer of the Year, AFRL/RV (Space Vehicles Directorate)
2011	NASA Group Achievement Award (Orion Abort Flight Test Pad Abort-1 Team)
2007	Company Grade Officer of the Year, 412 TW/EN (Test Engineering Group)

PROFESSIONAL MEMBERSHIPS AND AFFILIATIONS

Member, American Institute of Aeronautics and Astronautics
Member, Society of Flight Test Engineers

OTHER ACHIEVEMENTS

2011	Scientist, Engineer & Technical Management Award, AFRL/RVS (Spacecraft Technology Division)
2011	Top 20 percent graduate, Air Force Institute of Technology

PUBLICATIONS

Yates, Max, *Angles-Only Navigation Technique for Maneuver-Free Spacecraft Proximity Operations*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, September 2017.
Yates, Max, *Stochastic Orbit Prediction Using KAM Tori*, MS Thesis, Air Force Institute of Technology, Wright-Patterson AFB OH, March 2011.
Idicula, J., Williams-Hayes, P., Stillwater, R., and Yates, M., “A Flight Dynamics Perspective of the Orion Pad Abort One Flight Test,” AIAA Atmospheric Flight Mechanics Conference, 2009.

EFFECTIVE DATES OF PROMOTION

Second Lieutenant June 11, 2006
First Lieutenant June 11, 2008
Captain June 11, 2010
Major October 1, 2016

(Current as of July 2017)