Space-Based Laser Guide Stars for Astronomical Observatories

by

James R. Clark

S.B., Massachusetts Institute of Technology (2014)
S.M., Massachusetts Institute of Technology (2016)

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

Doctor of Philosophy in Aerospace Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2020

Author . . . .
Department of Aeronautics and Astronautics
August 18, 2020

Certified by . . . .
Kerri Cahoy
Associate Professor
Thesis Supervisor

Accepted by . . . .
Zoltán S. Spakovszky
Chairman, Department Committee on Graduate Theses

THIS PAGE INTENTIONALLY LEFT BLANK

Space-Based Laser Guide Stars for Astronomical Observatories

by

James R. Clark

Submitted to the Department of Aeronautics and Astronautics
on August 18, 2020, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Aerospace Engineering

Abstract

The Laser Guide Star (LGS) concept is proposed to enable reductions in cost of next-generation space telescopes, by providing reference targets (as bright as apparent magnitude -7) to enable wavefront stability and control (WFSC) to compensate for high-rate motions of mirror segments. This will relax the requirements on the stability of the telescope and flow down to metrology, construction, and control.

In this work, we present the detailed design of an LGS small satellite (and constellation of LGSs) that would fly in formation with a large space observatory that uses adaptive optics (AO) for wavefront sensing and control, or orbit around the Earth to support ground-based telescopes. We find that an LGS small satellite using the 12U CubeSat standard can accommodate a propulsion system sufficient to enable the LGS satellite to formation fly near the targets in the telescope boresight and to meet exoplanet direct imaging mission requirements on number of targets and duration. We simulate the formation flight for an LGS/telescope system at L2 to assess the precision required to enable the wavefront sensing and control during observation, and find that commercial off-the-shelf attitude control hardware can easily satisfy the pointing needs (error $< 14°$) and that the telescope needs to update the LGS no more than once every five minutes. We compare and recommend commercial off-the-shelf (COTS) propulsion and attitude determination and control systems (ADCS) for controlling the LGS spacecraft. We develop a constellation design tool for assessing the number of LGS spacecraft required to support a desired rate and quantity of observations at L2, and for trading that quantity against the parameters of the LGS spacecraft and the telescope(s) they support. We present a design reference mission (DRM) for deploying up to 19 LGS spacecraft to L2 to assist the Large Ultraviolet Optical Infrared Surveyor (LUVOIR). The L2 LGS DRM covers 259 exoplanet target systems with 5 or more revisits to each system over a 5-year mission. We also identify a series of technology demonstration missions for deploying an LGS satellite to geostationary orbit and other Earth orbits for use with 6.5+ meter ground telescopes with AO to observe Wolf 1061, 40 Eridani, and other near-equatorial targets.

Thesis Supervisor: Kerri Cahoy
Title: Associate Professor

Acknowledgments

The author wishes to acknowledge and thank Kerri Cahoy for her mentorship in engineering and education, and for her patience in coaching the author's flights of fancy into real research.

The author also wishes to acknowledge and thank Richard Binzel, John Conklin, Ewan Douglas, John Mather, and Rhonda Morgan for their valued technical and applied perspectives.

The author also wishes to acknowledge and thank his colleagues in STAR Lab and the Laser Guide Star team: Weston, Ashley, Akshata, Greg, Yinzi, Jared, Jhen, Will, Ezinne, Rachel, Ondrej, Amelia, Cadence, and Christian.

Finally, the author wishes to acknowledge and thank his family for their support, during all times, precedented and otherwise.

This research was funded in part by the Wide-Field Infrared Survey Telescope/Roman Space Telescope program (award number NNG16PJ24C/61238711-122362) and by the JPL Strategic University Research Partnership program (RSA No. 1646345).

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

Chapter 1

Introduction

The concept of a space-based laser guide star (LGS) was first articulated by Greenaway and Clark (no particular relation to the author) in 1994 as "Pharos" [26]. They envisioned a satellite in a highly-elliptical orbit to illuminate ground-based telescopes, which would serve as a reference source for wavefront stabilization and control (WFSC) and photometric calibration, to allow astronomers to correct for the effects of atmospheric turbulence and fading. In practice, ground-based telescopes have instead used lasers shining up from the ground that excite and illuminate the upper atmosphere for their adaptive optics systems. As space-based telescope diameters grow larger and larger, from Hubble (2.4 m, monolithic) to James Webb Space Telescope (6.5 m, segmented) and beyond (*e.g.* LUVOIR, 9.2 m or 16 m, segmented), we make the case for deploying companion spacecraft to augment their capabilities and relax the expensive and technically challenging construction stability requirements.

1.1 Motivation

The brightness of a celestial object is described by its apparent magnitude. The brightest stars are assigned to be "first magnitude" or magnitude 1, and the scale continues logarithmically, with a difference in five magnitudes corresponding to a factor of 100 in absolute brightness. The human eye can see stars as dim as magnitude 6 (under ideal sky conditions), and telescopes have seen individual stars and asteroids as dim as magnitude 28. The Hubble Space Telescope conducted Extreme Deep Field observations with over twenty days of integration time, and imaged galaxies as dim as magnitude 31. However, imaging Earth-like planets will require new classes of telescopes; Earth is one ten-billionth as bright as the Sun (10^{-10}), or a difference of twenty-five magnitudes, so a star between magnitudes 5 and 10 will have Earth-like planets between magnitudes 30 and 35, and planets will not obligingly remain still for a twenty-day observation.[1]

Advancements in adaptive optics and coronagraph technology offer the capability to suppress a star's light and see its surrounding planets, including those like Earth. [42] Pueyo and N'Diaye [47, 49] simulated what our Solar System might look like through such a telescope, reproduced here in Figure 1-1; the goal of LUVOIR, the Large Ultraviolet, Optical, and Infrared Surveyor telescope, is to achieve that level of performance and produce similar images of other star systems. A list of 259 stars that are key candidates for exoplanet direct imaging has been distributed by Chris Stark *et al.* [57]

This dissertation, a detailed assessment and development of tools for the space-based laser guide star concept has three factors: enhancing the capabilities and reducing the costs of space telescopes, advances in formation-flying small satellite capabilities at costs low enough to consider as an alternative to the pricetag of new technology on a large space telescope capability, and the recent demonstration of small-satellite technology beyond low Earth orbit by the MarCO spacecraft..

1.1.1 Space telescope capabilities and costs

To achieve contrast high enough for direct imaging of exoplanets on its own, the 9 or 16 m diameter LUVOIR would have to achieve wavefront stability on the order of picometers, smaller than the radius of a hydrogen atom. [44] This is not to say that the spacecraft must be still on the scale of atomic lengths, but that, statistically, the RMS uncertainty of the telescope's optical measurements is below 10 picometers. The thermal cycling, atmospheric drag, and perturbations of low Earth orbit would

[1] Even if they did, devoting a telescope to a single star for so long may be difficult to justify.

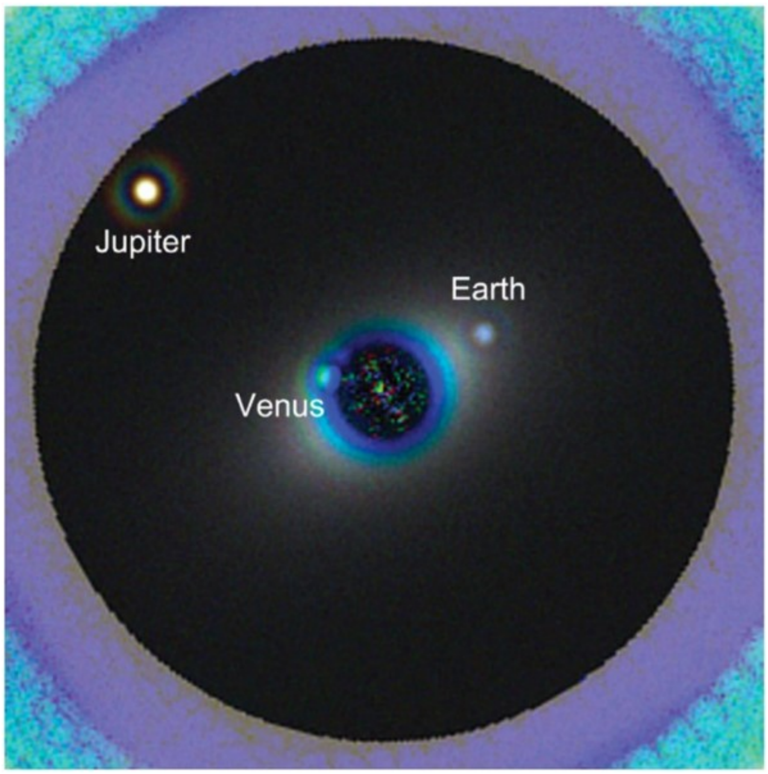


Figure 1-1: A simulated view of the Solar System through a LUVOIR-like space telescope. Figure by Pueyo and N'Diaye. [47, 49]

be impossible obstacles for such a large telescope to overcome, and so LUVOIR will be deployed to the Sun-Earth L2 Lagrange point. [28] This is close enough to communicate relatively easily with Earth, but far enough away so that it will not be disturbed by the Earth's thermal, atmospheric, and gravitational disturbances.

Devices with picometer stability have been built and demonstrated in the laboratory [62, 51] and in space [35], but not at the scale of tens of meters, nor are these devices capable of surviving launch and then unfolding in space *and then* stabilizing. Of all of LUVOIR's technical systems, it is the ultra-stable optical platform that is least mature. [10] L2 is nearly 1.5 million kilometers away from Earth, or four times the distance of the Moon, and so LUVOIR will be far more challenging to repair than Hubble was, assuming robotic servicing is possible. A scale drawing of the Sun-Earth L1 and L2 points, as well as the Moon's orbit and an L2 halo orbit, is shown in Figure 1-2, and a close-up view of the Earth/Moon system and Sun-Earth L2 halo orbit is shown in Figure 2-1.

The cost of building such a large telescope to have picometer stability would be significant and possibly cost-prohibitive; the James Webb Space Telescope (JWST) is "only" 6.5 meters across, "only" observes the red and infrared spectrum, and must have "only" 51 nm RMS of wavefront stability [34], but is experiencing chronic delays, and the final mission budget is expected to reach nearly 10 billion dollars. [43] LUVOIR is expected to have a similar architecture, but to be 1.5 to 2.5 times the size and with a minimum wavelength one-sixth of Webb's. Empirically, space telescope costs are estimated to scale with diameter raised to the 1.12th power, which would put LUVOIR's cost in the ballpark of 16 to 28 billion dollars. [56]

1.1.2 Telescope companions

To relax the stability requirements and reduce complexity on the space telescope, multiple mission concepts have proposed flying external companions in formation with the space telescope. The architecture which has received considerable investigation and support is the Starshade concept, where the coronagraph element is taken outside of the telescope and instead becomes a separate formation-flying spacecraft. [60, 25]

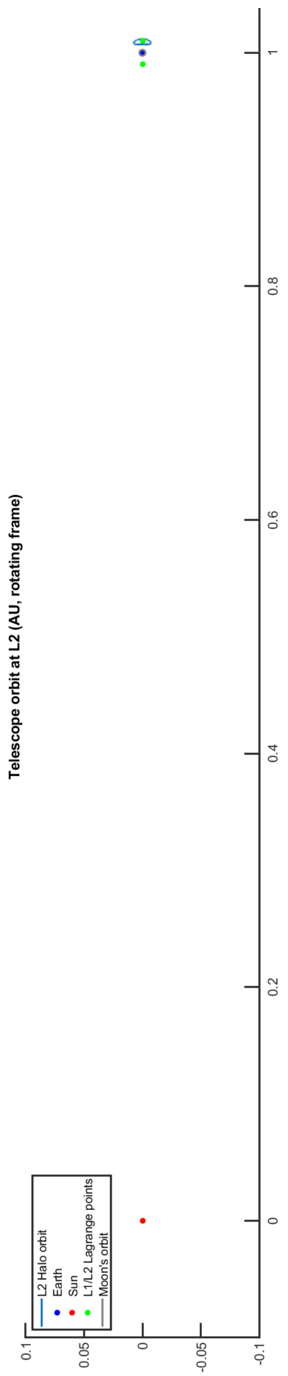


Figure 1-2: To-scale drawing of the Sun-Earth L1 and L2 Lagrange points. The light grey circle surrounding the dark blue dot at the top of the image is the Moon's orbit.

This spacecraft will block light from a target star achieve the contrast of 10^{-10} and remove the technical complexity from the space telescope, which can then be built more conventionally. However, the Starshade will also require being built to optical tolerances, and its size (at least 16 meters, with 30-, 70-, and 100-meter options considered for larger telescopes) means that it is impractically costly to field more than one or two. Even a single Starshade prototype operating for one year (rather than the five years desired by LUVOIR) would cost approximately 400 million dollars, and a three-year mission paired with a commercial telescope (1-meter class or smaller) would cost approximately a billion dollars. [52] Two Starshades operational limits the pace of observations to one every thirty days or so [25] while LUVOIR's desired pace of observations is to conduct one nearly every day. [57]

Another possible companion spacecraft, which is studied in detail in this work, is the laser guide star (LGS). Like Starshade, LGS aims to reduce the technical complexity of the telescope structure. While Starshade works by "disassembling" the mission into an occulter plus a (simpler) telescope, LGS works by providing enough photons to the telescope's wavefront stability and control system (WFSC) to control enough actuators at a high-enough update rate to correct for the high-frequency flutter of large space telescope segments due to onboard vibration sources (pumps, reaction wheels, *etc.*) and other phenomena.

Douglas *et al.* 2019 [19] developed the optical requirements for such a paired LGS mission: a laser at a range of tens of thousands of kilometers, as bright as magnitude -7, would allow the telescope's stability requirement to be relaxed by two orders of magnitude (*i.e.* the WFSC can produce wavefront stability of 10 pm at the focal plane even if the telescope's mirror is only stable to 1 nm). Marlow *et al.* 2017 [38] developed some small-satellite bus options that could support such a laser in geostationary orbit. However, that LGS spacecraft was designed to support space situational awareness of other satellites in the GEO belt, which differs in many ways from the astronomical observations studied here. First, the observation durations are short, measured in minutes at most, when the shortest of LUVOIR's scheduled observations is over five hours long. Second, the SSA LGS would use propulsion to place itself on an orbit that would pass in front of or behind the target of observation, but would not use its thruster during observation, as will be required to support astronomy at L2. Finally, the mission operation will be more challenging, five years at L2 (over a million km from Earth) vs. two years in GEO ("only" 36,000 km away).

In this work, the systems engineering for a small-satellite to serve at GEO and beyond is developed in more detail, and tools are made to compute just how many LGS spacecraft are require to service a mission.

1.1.3 Small satellites in deep space

The goal adopted in this work of the LGS concept is to relax telescope requirements *and* support a high pace of observations *without* modifying the proposed space telescope[2], and our proposed way to do this (as will be discussed in Chapter 4: Constellation Design) is with a constellation of CubeSats.[3] This is achievable because CubeSat technology has matured to allow complex operations, including multi-satellite operation and propulsion, even in deep space. This has been demonstrated with the Mars Cube One ("MarCO") mission, a pair of CubeSats with cold-gas propulsion systems which were deployed along with the InSight mission, to serve (successfully) as radio relays for InSight during its entry, descent, and landing (EDL) on Mars. [31, 59] In the coming years, The Lunar Polar Hydrogen Mapper (LunaH-Map) and Lunar IceCube missions, among others, will be deployed from the first launch of the Space Launch System to fly to the Moon; [24, 27] they will use electric propulsion systems and fly as far as (approximately) one million kilometers from Earth before arriving in Lunar orbit from a weak-stability boundary (WSB) orbit – not quite as far as Sun-Earth L2, but experiencing the same environment. [12]

[2]In Section 2.2.1, we will consider what new options are available if LUVOIR is modified.

[3]The mission could also be completed by a handful of larger satellites with monopropellant propulsion systems, but that would be a much more expensive course of action. Even granting the monopropellant thruster a generous 230 seconds of specific impulse [2], a monopropellant-powered LGS spacecraft would have to be over two-thirds fuel by mass to match the delta-V capability of a CubeSat with electric propulsion (2480 m/s, see Table 2.2). This is not impossible to construct, but by virtue of its larger size and greater complexity than the CubeSat option, it will be much more expensive, when the whole point of this concept is to reduce the cost of the overall mission.

LGS location	**Telescope location**		
	Ground	GEO	Sun-Earth L2
Ground	Sodium AO	Lasercom-like LGS	-
LEO	Lasercom-like LGS	Lasercom-like LGS	-
GEO	**Pathfinder 1**	**Pathfinder 2**	-
HEO	Long dwell, **PF3**	**Pathfinder 2** (con't)	-
Sun-Earth L2	-	-	**LUVOIR-LGS**

Table 1.1: A matrix of possible telescope-LGS architectures. LEO: Low Earth Orbit. GEO: Geostationary Earth Orbit. HEO: High Elliptical Orbit.
Sodium AO: state of the art [18]
Lasercom-like LGS: cf. OCSD [17] and CLICK [14] in LEO, LCRD [21] in GEO
Pathfinder 1: LGS in GEO, cf. Marlow *et al.* 2017 [38] but for astronomy, Ch. 5
Pathfinder 2: LGS and telescope in GEO, then LGS boosts up to higher orbit. Ch. 5
Long dwell: proposed by Pharos [26], ORCAS [45], Pathfinder 3. Ch. 5
LUVOIR-LGS: discussed in Ch. 2, 3, 4

1.2 Laser Guide Star roadmap

There are many possible architectures for using laser guide stars, depending on the location of the telescope and LGS. Table 1.1 shows a "matrix" of some possible LGS architectures, including those which will be discussed in this work (space-based LGS for ground- and space-based telescopes). In the state of the art, ground-based lasers are used by ground-based telescopes, and the goal for the future is to use laser guide stars at L2 to support space telescopes orbiting at L2. It's a long way from here to there, and so while most of the work in this dissertation is focused on the L2 case, three pathfinder missions are studied in Chapter 5, placing the laser guide star first at GEO, and then at higher orbits to validate the operation of the LGS-telescope system.

The concept of deploying a laser guide star satellite to geostationary orbit (GEO) was previously studied in Marlow *et al.* 2017 [38], but the goal of that project was to fly close in front or behind other satellites in geostationary orbit, to support space surveillance efforts of the GEO belt. In this work, the GEO-LGS architecture is studied for its application to astronomy, in particular, exoplanet direct imaging. Most of the exoplanet mission application discussion is in Chapter 5. To briefly motivate the study of deploying LGS spacecraft to geostationary and higher orbits, we can consider

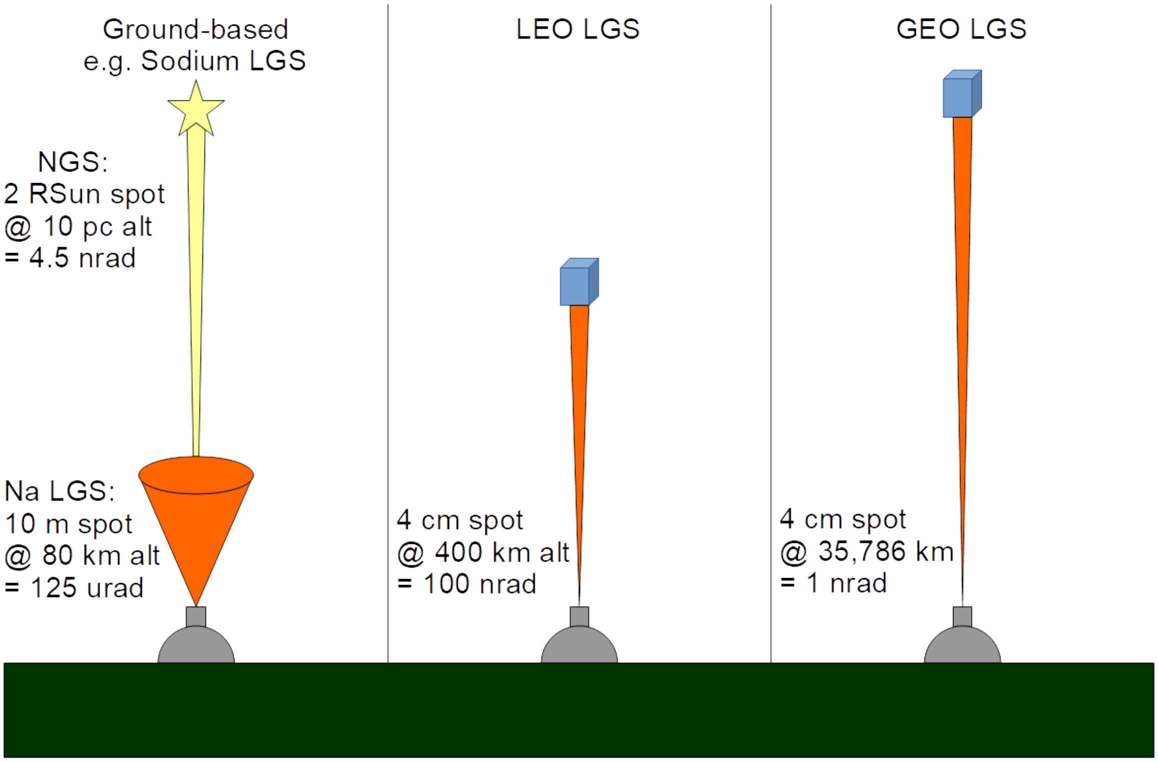


Figure 1-3: A comparison of the angular widths of natural stars and laser guide stars, as seen by the telescope (the laser beams will spread around the telescope). Ground-based guide stars (far left) produce spots that are much broader than actual stars, but a space-based laser guide star would produce light that is much more star-like (middle and right).

the sketch of different LGS placement options shown in Figure 1-3. Conventional ground-based *e.g.* sodium laser guide stars produce spots that are diffuse objects of much greater angular width than natural stars, which means that information gained from the sodium LGS is not perfectly representative of the corrections required to improve the image of the target star. Additionally, sodium laser guide stars are subject to a phenomenon called focal anisoplanatism or the "cone effect"; essentially, because they are produced within the atmosphere at such close focal distance (compared to the distant object being sensed), the air above and surrounding the laser's path is not sensed, which limits the modes that the WFSC system can compensate for. This is illustrated in Figure 1-4.

Finally, because the laser light is passing both up and down through the atmosphere before reaching the telescope, it carries no information on atmospheric tip/tilt

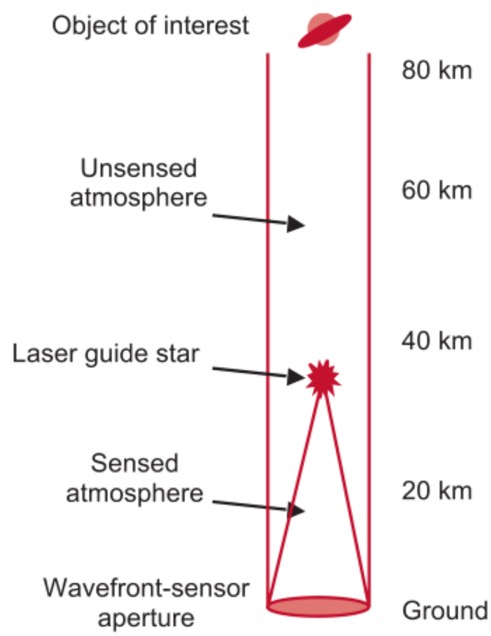


Figure 1-4: An illustration of focal anisoplanatism experienced by sodium laser guide stars. Figure by Tyson and Frasier. [63, p. 41]

error. Telescopes can use natural guide stars to perform tip/tilt correction, but this is only possible if there is a bright enough star close enough to the target (within 25" at Keck, for example), and that limits the sky coverage of such instruments – 60% when looking within the Milky Way galaxy, but less than 10% when looking outside of it. [37] Placing the LGS at geostationary orbit or higher will result in a spot that appears much more star-like, samples the entire atmospheric column, and will provide tip/tilt information at any point in the sky.

1.3 Contributions

The four contributions developed in this dissertation are:

1. Constellation planner for space-space LGS and ground-space LGS
   Develop a scheduling tool that considers LGS capability, dynamics of Earth orbit and L2, and target stars and observatories. The scheduling tool enables performance trades based on metrics such as: number and propulsion capability of LGS spacecraft, number and duration of target observations supported, and

the pace of observations.

2. Telescope-LGS formation flight

   Develop a model of the formation flight of a LGS and a ground or space telescope during target observation. This model supports trades on metrics such as: sensor, thruster, and pointing uncertainty, telescope-LGS vector stability, and considers some of the possible negative impact(s) by the LGS on observations (glint, etc.).

3. LGS architecture tool

   Develop LGS configuration(s) that satisfy performance goals: a factor of 2 improvement in accessible star targets (from 259 to 518) without impacting pace of observation (1 observation per 1.2 days, even when adding external companions).

4. Technology development and demonstration plan

   Develop pathfinder missions for LGS and telescope not necessarily at L2. Compare options for different LGS orbits (LEO, GTO, GEO, HEO) based on operational and systems engineering cost and scientific utility.

1.4 Organization of sections

Each of the above contributions are addressed by the following chapters:

1. Constellation planner: Chapter 4
2. Formation flight: Chapter 3
3. LGS architecture: Chapter 2
4. Technology demonstration plan: Chapter 5

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Approach

2.1 Concept of operations

As discussed in Table 1.1, the LGS concept of operations studied for most of this work is for one or more LGS spacecraft to fly in formation with a large space telescope at L2, nominally represented by the LUVOIR mission. [9] LUVOIR, like many past and planned space observatories[1], will fly a halo orbit at the Sun-Earth L2 Lagrange point, which orbits the Sun with a period of one year while remaining approximately 1.5 million km (0.01 AU) away from Earth. This keeps the telescope close enough to Earth for monitoring and control, without subjecting the telescope to the challenges of the varying thermal environment of low Earth orbit. LUVOIR is being designed for launch in the late 2030s, with several potential architectures under trade; in this work, we will focus on the 9.2-meter "Architecture B" option. [23]

The non-dimensionalized circular restricted three-body problem is used to simulate the orbits of the telescope and laser guide star at L2. The calculations were performed in MATLAB, with code available in Appendix B.3.6. The starting parameters of the simulation are presented in Table 2.1, and the resulting halo orbit is plotted in Figure 2-1.

When the telescope is making an observation, a laser guide star satellite will fly

[1] For example: the Wilkinson Microwave Anisotropy Probe (WMAP) and Gaia, and the upcoming James Webb Space Telescope (JWST) and Nancy Grace Roman Space Telescope (RST, previously known as the Wide Field Infrared Survey Telescope or WFIRST)

Table 2.1: Parameters used for the circular restricted three-body problem simulation.

Parameter	**Value** (non-dim.)	**Notes**
μ	3.036×10^{-6}	Sun-Earth system
x_0	1.00717285919175	Initial position of LUVOIR, AU (1.07×10^6 km from Earth)
$\dot{y}_0$	0.0163636	Initial velocity of LUVOIR, $2\pi\times$AU/yr (487 m/s relative to Earth)

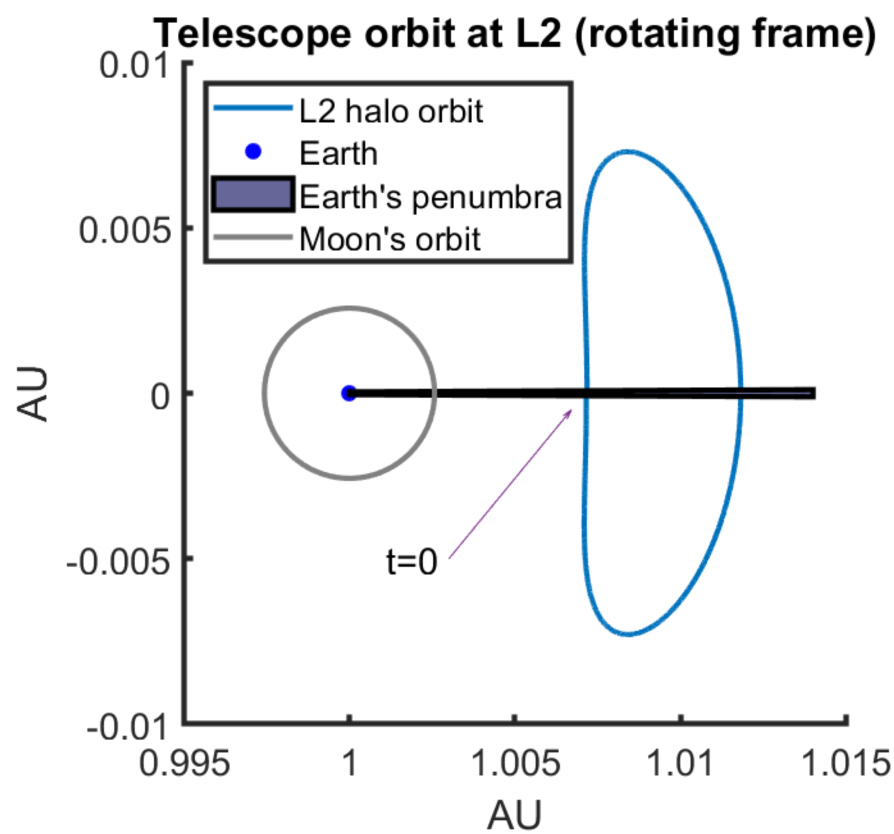


Figure 2-1: Simulation of LUVOIR's six-month halo orbit, shown in the heliocentric reference frame co-rotating with the Earth. Elapsed time $t = 0$ corresponds to the midpoint of the straight segment of the orbit, when LUVOIR is closest to Earth.

on the line of sight between the telescope and the star it is observing, and shine its laser at the telescope to provide a reference source for the telescope's adaptive optics system. This is illustrated in Figure 2-2.

LUVOIR is currently planned to execute a five-year mission of observing a list of 259 stars enumerated by Stark *et al.* [57], with each star being observed five or six times for 1,539 total observations. The stars on the target list are plotted in Figure 2-3.[2] The stars with apparent magnitude of 2 or brighter are also plotted on Figure 2-3, to illustrate why a satellite-based laser guide star is useful; even though some of the target stars appear on the map to be close to bright stars, the smallest separation is over 0.3 degrees (> 1000 arcsec, vs. 25 arcsec for Keck NGS AO [37]), and the majority of bright stars are more than two degrees away from the nearest target star.

2.2 LUVOIR requirements

Many of the key parameters of the LGS spacecraft, such as laser power and pointing, propulsion, and navigation precision, are set by the architecture of the telescope it is supporting. Because the goal of LGS is to reduce the cost of its host mission, we will spend the majority of our analysis considering a LGS architecture that minimizes the changes asked of LUVOIR.

LUVOIR's main instrument for observing exoplanets is ECLIPS, the Extreme Coronagraph for Living Planetary Systems. [28] The wavefront sensor for ECLIPS is currently baselined to be a Zernike wavefront sensor (ZWFS), similar to that baselined for the Roman Space Telescope. [9, 68] The ZWFS works by focusing incoming light through a $\pi/2$ phase mask; this converts changes in the phase of light into changes in intensity at the instrument image plane. However, while the ZWFS is sensitive to small changes in phase, its use can also impose a tight requirement on the alignment

[2] All full-sky maps in this work use the Robinson projection, a compromise with some area distortion at the poles but which preserves shapes in an aesthetically pleasing way. [69] The National Geographic Society now uses the Winkel III projection [41], but the Robinson projection is still an excellent projection [15], and MATLAB's Mapping Toolbox doesn't have Winkel III as a built-in option.

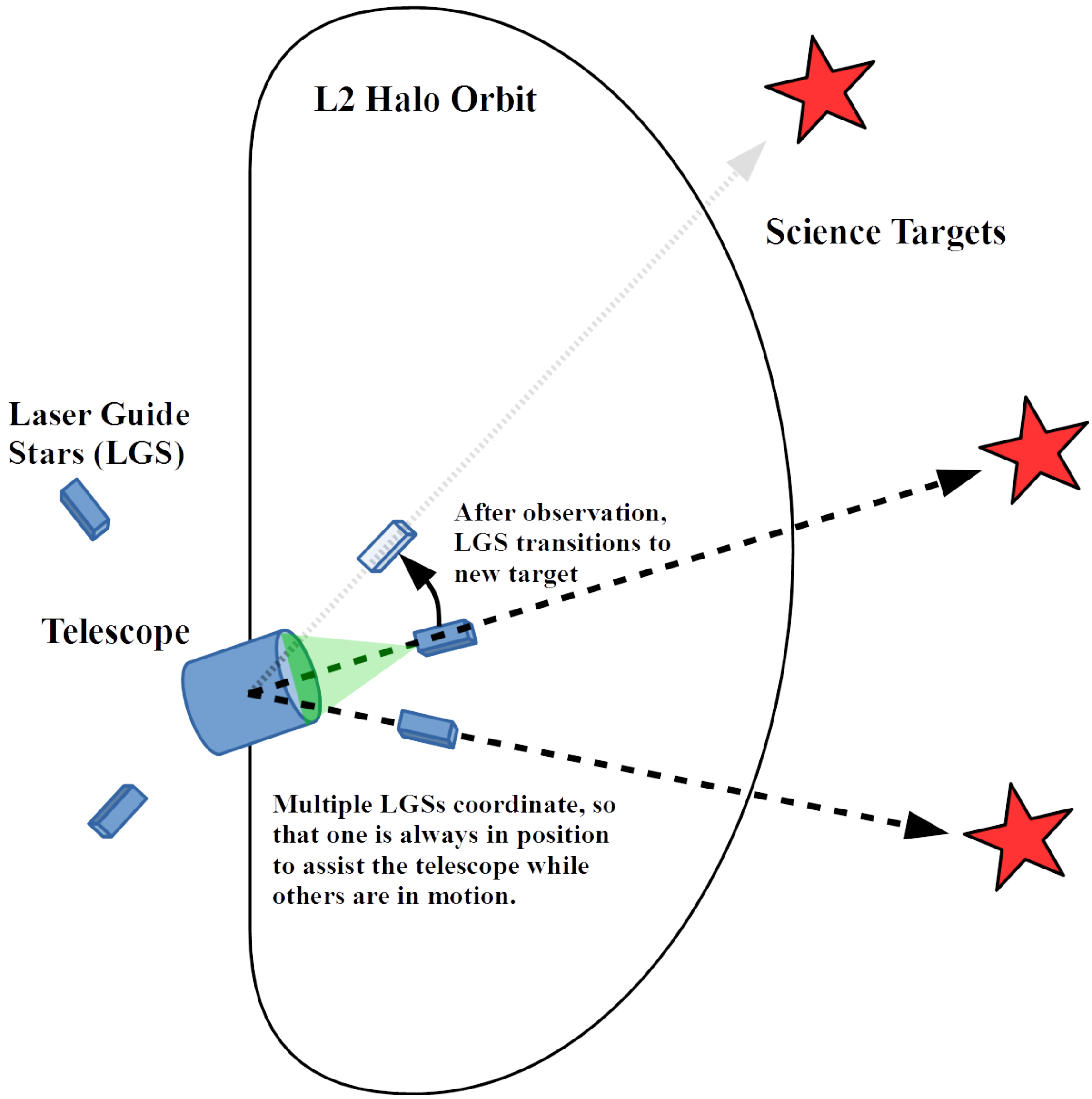


Figure 2-2: A schematic diagram of the satellite laser guide star concept of operations with a space telescope at Sun-Earth L2. (Earth off-panel to the left; see Figures 1-2 and 2-1 for scale.)

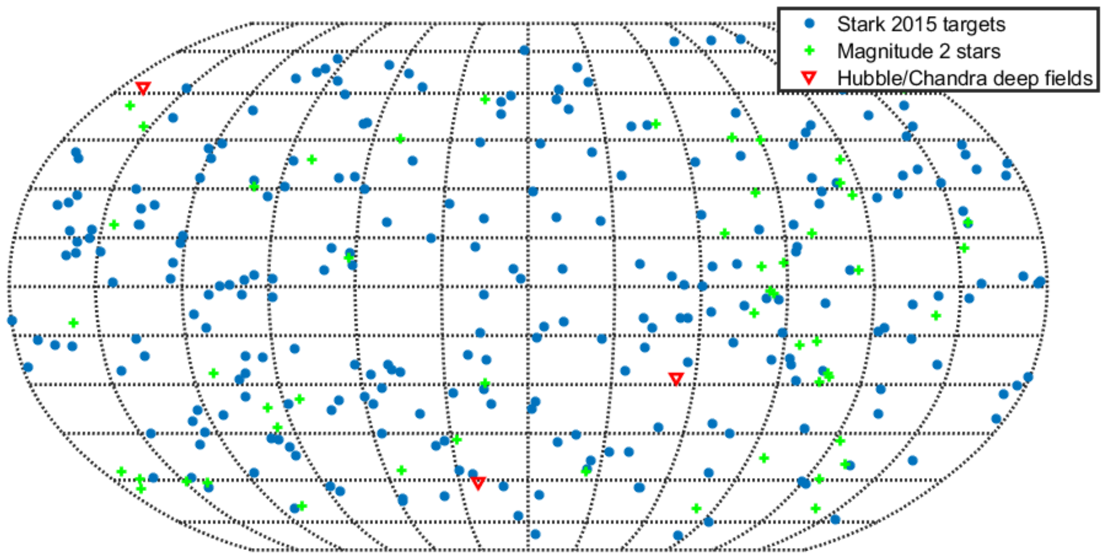


Figure 2-3: A sky map of LUVOIR's targets (blue circles [57]), stars of magnitude 2 or brighter (green crosses), and Hubble and Chandra deep fields (red triangles).

between light from the LGS and from the target star. If the LGS spacecraft is too close, the difference in curvature between the laser wavefronts and the target star's light will make the LGS useless for improving the telescope's image of the target star. It would be like trying to use reading glasses as binoculars.

As calculated by Douglas *et al.* 2019, given a space telescope with aperture radius R_T and a laser of wavelength λ, the range R_C between the laser guide star and telescope that will keep the laser's wavefronts flat to within $\pm\pi/4$ is given in Equation 2.1. For the 9.2 meter diameter LUVOIR Architecture B, and the 980 nm near-infrared laser (to stay out of the visible band during observations, see Section 2.3.1), this imposes a minimum separation of 43,184 km (sometimes referred to in this work as "43,000 km") between the telescope and LGS spacecraft.[3]

$$R_C = 2R_T^2/\lambda \tag{2.1}$$

Having assessed the required separation between the LGS spacecraft and the telescope, we can calculate the limits on how far off-axis the LGS spacecraft may fly. To prevent ambiguity, the angle error across each mirror segment (each with diameter

[3]This is comparable to the altitude of geostationary orbit above the Equator, 35,786 km.

D_S) should be less than one wavelength, so the maximum permitted off-axis error Δx_{max} is given in Equation 2.2. Given that LUVOIR-B's mirror segments are 1.15 m across [47], and the range and laser wavelengths used above, the maximum off-axis error is 37 meters.

$$\Delta x_{max} = R_C \frac{\lambda}{D_S} \tag{2.2}$$

However, there are other reasons to keep the LGS spacecraft on-center, which will impose tighter requirements. The LGS spacecraft will of course be emitting a bright laser beam, which must be kept away from the science instruments. The way accomplish this with the least modifications to LUVOIR (i.e. no modifications at all) is to require the LGS spacecraft to fly inside the inner working angle (IWA) of LUVOIR's coronagraph, that is, the central region of its field-of-view which is masked from the science sensor. ECLIPS's coronagraph will suppress the light coming from the central star so that the surrounding planets can be seen, and the LGS can hide its laser beam there as well, as shown in Figure 2-4. The angular width of the inner working angle of a coronagraph (and also the outer working angle (OWA), the size of its field of view) is often measured by units of λ/D: the wavelength in which the observation is made, divided by the diameter of the telescope. The IWA of ECLIPS is presented as $4\lambda/D$ [47], which results in an off-axis error of 9 meters when multiplied by the telescope-LGS separation, but that is the region in which magnitude-25 *planets* are suppressed from view. To suppress a bright laser of magnitude -7, the required angle is $0.25\lambda/D$ [50], or a distance of 0.59 meters off-center.

2.2.1 LUVOIR modifications

If modifying the LUVOIR design is an option, there are several minor modifications that could be made to LUVOIR to reduce the number of LGS spacecraft required or otherwise relax some of the formation-flight requirements.

As discussed in Section 4.1.2, the LGS constellation design is very sensitive to the required separation between the telescope and LGS spacecraft. From a propellant-

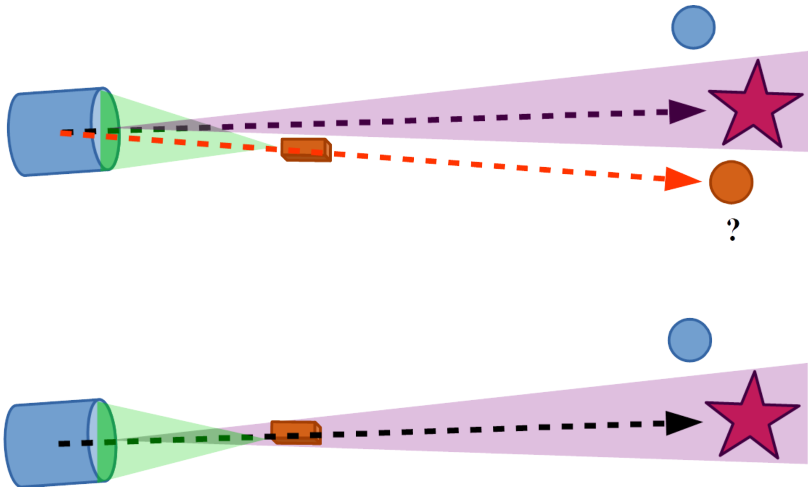


Figure 2-4: An illustration of the operation of LUVOIR's coronagraph. LUVOIR (left) will look at a solar system (right) and use its coronagraph to prevent the star's light from reaching the science sensor (purple triangle), which will allow it to see the planets around that star. However, if the LGS spacecraft flies outside of that IWA, even if the laser is filtered out, it would be about as bright as a planet, which would disrupt the observation (orange "planet", top right).

management perspective, it is desirable to fly closer, but this would result in additional wavefront curvature. LUVOIR would have to use a different wavefront sensor, or have an additional focus-correction stage in front of it, in order to tolerate greater than $\lambda/4$ curvature.

That change would be fairly invasive (the ZWFS was chosen *specifically because* it is so sensitive), but a change that would be more impactful in relaxing the constraints on LGS without impacting LUVOIR's observations would be to add some line filters to LUVOIR's optical paths to reject the wavelengths used by the laser. As will be discussed in Section 2.3.1, these can be chosen to lie outside of the science band being observed, so that the observation's quality is not affected. Commercial notch filters can be purchased with with optical density (OD) 8 [4], that is, passing only 1 part in 10^8 of light whose wavelength falls within its filter bandwidth. That is equivalent to a reduction in brightness by 20 magnitudes, which would take the laser from magnitude -7 to magnitude 13, comparable in brightness to a star, which can be suppressed as far out as $1\lambda/D$. Both limits ($0.25\lambda/D$ and $1\lambda/D$) will be considered in later calculations.

2.3 Laser Payload

The main payload of the LGS spacecraft is its laser. Douglas *et al.* 2019 [19] simulated the closed-loop wavefront control over a range of guide star brightness, and found that, if the guide star is of magnitude -3, the wavefront control system could hold wavefront error to 10 pm or less, even if the telescope's mirror segments are moving up to 100 pm RMS, or a factor of 10 relaxation from optical stability to mechanical stability requirement. If the laser guide star is as bright as magnitude -7, that relaxation factor can be between 100 and 1,000 (depending on the power law of the segment-motion power spectral density). Therefore, the required magnitude for an effective LGS in this analysis shall be -3, with a goal of -7.

To calculate the brightness of the laser as seen by LUVOIR, a Gaussian link budget is used. The link budget code is printed in Appendix B.2, but the key assumptions used in the calculations are that the Gaussian beam waist is 1/3 of the physical aperture radius, to minimize diffraction losses, and that there is a -3 dB pointing loss for the sake of margin. Under these assumptions, a 5 W laser at 980 nm (see below for the wavelength selection) transmitted through a 3.5 cm diameter aperture will have a FWHM beamwidth of 63 μrad (13 arcsec), and will be received at LUVOIR at magnitude -7. At the expected distance between the LGS spacecraft and host telescope calculated in Section 2.2, 43,000 km, that corresponds to a laser spot size of diameter 2700 meters, or nearly 300 times the diameter of LUVOIR itself.

2.3.1 Wavelength selection

The LGS payload will have at least two wavelengths available, tentatively selected to be 532 nm (green) and 980 nm (infrared). This is done so that there is always at least one wavelength available outside of whatever band LUVOIR is using to observe. The bright laser light can then be diverted to a fast WFSC system upstream of the science detector, so that it does not affect the observation. This is illustrated in Figure 2-5.

For some applications, it may be desirable to use a laser wavelength that lies within the science band, such as for measuring atmospheric disturbances over a ground-based

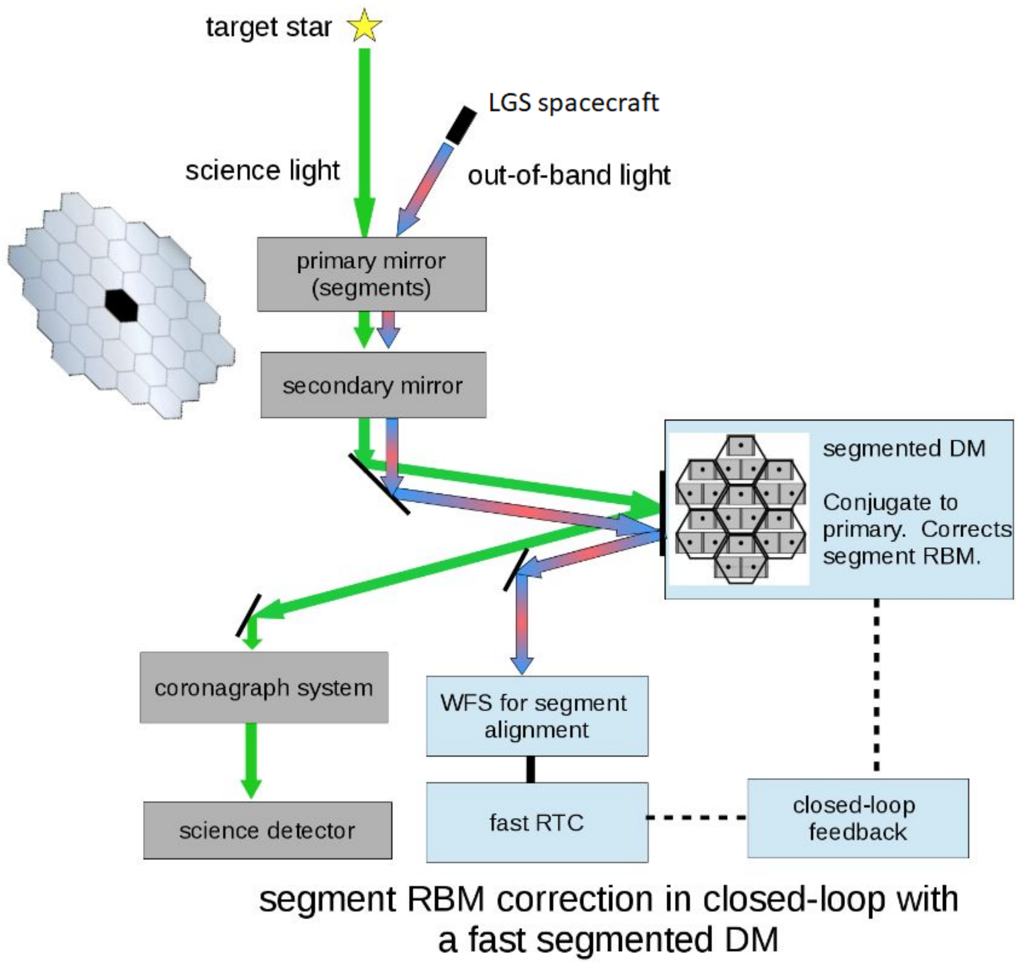


Figure 2-5: A schematic representation of how the LGS could work with telescope optics. A laser wavelength outside of the science band can be used and diverted upstream of the science camera to a fast WFSC system. RBM: Rigid Body Motion (of primary mirror segments). RTC: Real-Time Controller. Figure by Jared Males.

telescope or for co-phasing an interferometer. However, that would require the science payload(s) to "sacrifice" that wavelength plus some filter bandwidth to either side; further analysis of such configurations are application-specific and left for future work.

2.3.2 Beam pointing

The laser payload for the LGS spacecraft is based on that developed for CLICK, the CubeSat Lasercom Infrared CrosslinK. [14] CLICK is intended to enable CubeSats to share data at 20 Mbps at a range over 500 km. Like the proposed LGS, CLICK carries two lasers, including a beacon laser with a role analogous to the LGS primary function at 980 nm. CLICK's communications laser is in the 1550 nm band, not visible, but the beam pointing mechanism has been tested to have a 3-sigma pointing error of 1.6 arcsec, much narrower than the desired beamwidth of 13 arcsec.

2.4 12U Space-Space LGS/LUVOIR L2 point design

The LGS spacecraft studied in this work is built using the 12U CubeSat standard, based on previous MIT efforts designing a flexible high-power (>60 W) small-satellite bus. The ADCS and electrical power systems nominally use the GOMspace Nanopower P60 and BCT XACT, respectively, while the laser subsystem is based on MIT's Cubesat Laser Infrared CrosslinK (CLICK) mission. [72] The current best estimate masses of all of the parts of the satellite, excluding the propulsion system, add up to 11.5 kg (limit: 24 kg). A preliminary arrangement of the parts of the satellite is shown in Figure 2-6.

The parts are arranged such that a thruster up to 3U in volume (1.5 x 2 x 1)[4] can be mounted with its thrust axis perpendicular to the axis of the laser beam. This will allow the LGS spacecraft to resist the cross-track acceleration at L2 during a formation flight, which is discussed more thoroughly in Chapter 3. Propulsion systems that are offered for sale or advertised as in development were considered for this study. Their properties are summarized in Table 2.2.

The "parent" bus design developed by the STAR team from which the LGS design is derived was estimated to cost $11 million for one unit to complete the design, construction and testing process, and $5 million for subsequent identical units. Even if the LGS spacecraft cost $10 million each, to account for the additional development needed to assure reliability at L2, then twenty LGS spacecraft[5] would cost $200 million in total. That is still less than the cost of a 1-year Starshade technology

[4]If the spacecraft bus is enlarged to a 16U, then propulsion systems up to 5U in volume, 2.5 x 2 x 1, can be accommodated.

[5]The exact number of LGS spacecraft required to support LUVOIR's mission or others is discussed in Chapter 4: LGS Constellation Design, but twenty is a close round estimate.

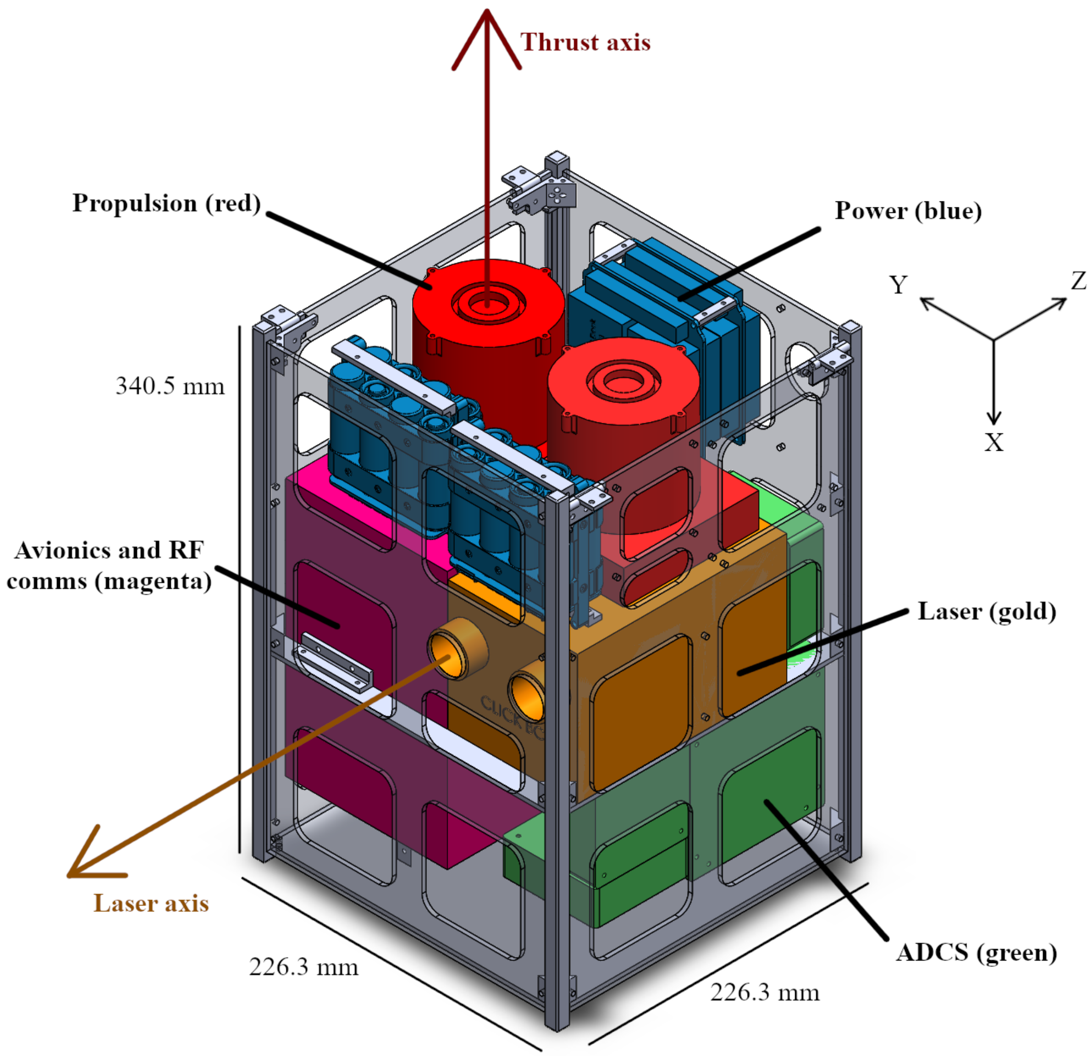


Figure 2-6: CAD rendering of conceptual LGS spacecraft design, in a 12U CubeSat bus. Not shown: body panels or deployable solar panels. Design by W. Kammerer, rev 1, 2019 Mar 21.

Prop system	**Mass** (kg)	I_{sp} (sec)	**Thrust** (mN)	**dV** (m/s)	**Vol.** (U)
Accion TILE 5000 x2 [1]	2.2 + 0.64	1500	3	672	2.5
Apollo Constellation [5]	4.5 + 1	1500	33	892	4
Busek BIT-3 [13]	1.4 + 1.5	2300	1.25	2480	2
Enpulsion IFM Nano x2 [22]	1.44 + 0.46	3000	0.7	1030	2
IFM Nano (max I_{sp}) x2 [22]	1.44 + 0.46	6000	0.5	2060	2
Phase Four x2 [46]	3 + 1	900	2	589	2
VACCO Green MiPS [67]	3 + 2	170	0.4	215	3
VACCO/JPL MarCO [66]	2.5 + 1	80	0.1	52	3

Table 2.2: Table of propulsion system options considered for a 12U LGS spacecraft. Mass is shown as propulsion dry mass + propellant mass. Delta-V capacity is calculated with a spacecraft mass of 11.5 kg plus propulsion system dry mass.

demonstrator mission (est. $400 million [52]), and considerably smaller than the nearly $10 billion total cost of the James Webb Space Telescope. [43] LUVOIR is intended to be over 50% larger than JWST, which, if all else were equal, would raise the mission price tag to nearly $15 billion (per Smart's parametric cost model of space telescope missions [56]), but JWST is at Technology Readiness Level 6[6] at same time that LUVOIR's optical system is TRL 2[7] [10], which would put the cost to develop LUVOIR at nearly $39 billion dollars. Both Starshade and LGS could more than pay for themselves by relaxing the requirements levied upon LUVOIR, but LGS can do so even at constellation scales.

2.4.1 Propulsion needs

With five years to complete 1,539 observations for LUVOIR's exoplanet detection and characterization mission, and with the average observation lasting 9 hours, the telescope-LGS formation will only have 19 hours to re-align from one target to another. We will see in Chapter 4 that multiple LGS spacecraft will be required to sustain this pace of observations.

The LGS spacecraft will spend the majority of its propellant in transiting between targets. If two stars are a separated by a small angle $\theta < 14^\circ$, and the LGS spacecraft is at a range R from the telescope, then we can approximate the distance that the LGS spacecraft must travel as $R\theta$ with error less than one percent. If the spacecraft has mass m and a thruster with thrust T (thus acceleration $a = T/m$), the way to cover that distance in the least time possible is to accelerate toward the new target point until it reaches the midpoint, and then turn over and decelerate to a stop at the new target point. This maneuver is illustrated in Figure 2-7, and the total time t required for this maneuver is given by Equation 2.3.

$$t = 2\sqrt{R\theta/a} = 2\sqrt{R\theta m/T} \tag{2.3}$$

[6]TRL 6: system/subsystem model or prototype demonstration in a relevant environment (ground or space). [54]

[7]TRL 2: Technology concept and/or application formulated, research underway to prove feasibility. [54]

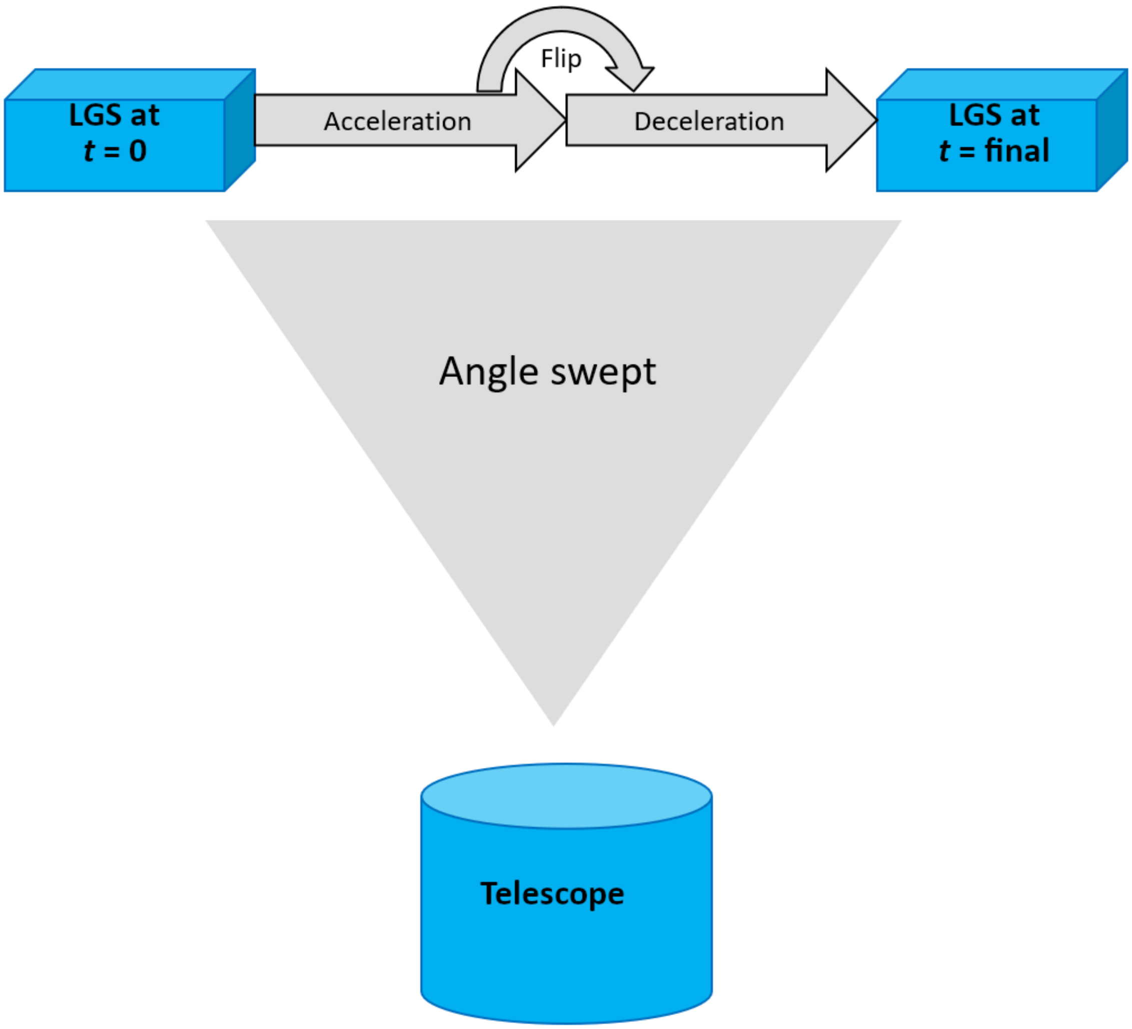


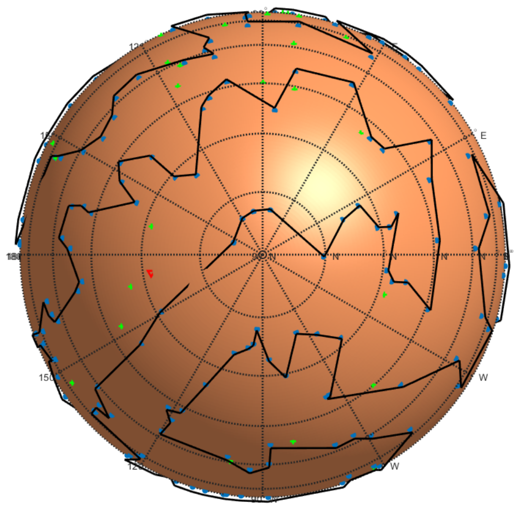
Figure 2-7: An illustration of a transit maneuver, with the LGS moving from in front of one target star to another.

The delta-V cost is that time multiplied by the acceleration, as shown in Equation 2.4.[8]

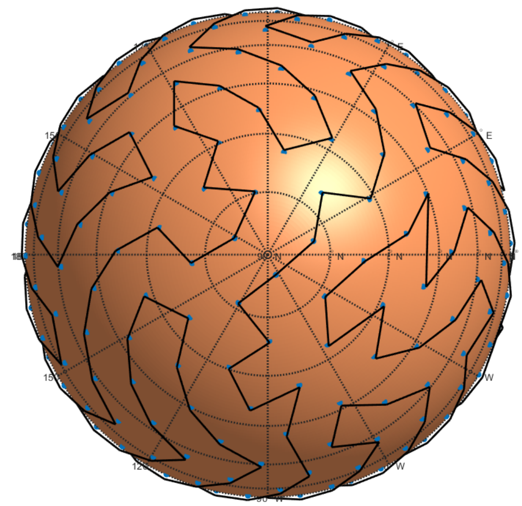
$$\Delta V = at = 2\sqrt{R\theta a} = 2\sqrt{R\theta T/m} \tag{2.4}$$

From this, we can see that we can complete any number of maneuvers with an arbitrarily small expenditure of delta-V, but only at the cost of decreasing thrust and increasing maneuver time.[9]

Equations 2.3 and 2.4 could be used as cost functions in a traveling-salesman-problem calculation on the LUVOIR target database (which was done to produce the single-salesman path in Figure 2-8a), but to simplify calculations and allow rapid trade studies of observation campaigns with different numbers of targets, we will first assume that all targets are approximately uniformly distributed over the sky by a Fibonacci grid. A Fibonacci grid is a set of points on a sphere constructed based on the golden ratio, which produces a more isotropic distribution of points compared to points which are evenly-spaced in latitude and longitude (or hour angle and declination). [61] To validate this simplification, MATLAB's `intlinprog` solver was applied to the traveling salesman problem for the LUVOIR target database (259 stars), and the average angle to transit between stars was found to be 8.8 degrees. For comparison, the average angle to transit in the Fibonacci grid, also with 259 points, is 11 degrees. The two courses are plotted in Figure 2-8. We can therefore proceed with the uniform-distribution assumption (*i.e.*, constant θ for all maneuvers) and know that any constellation designed on that basis will have margin when investigated in greater detail with the actual target population.



(a) Traveling-salesman path for LUVOIR target population. [57]



(b) Traveling-salesman path for Fibonacci grid with 259 stars.

Figure 2-8: Comparison of actual LUVOIR target population versus simplified uniform distribution. Note that the actual star population has clusters that allow fuel savings, while in the Fibonacci grid, all points are roughly equally distributed.

Prop system	**Min. transit time** (days)	**Maneuver count** Max thrust	**Maneuver count** Equal time
Accion TILE 5000 x2	4.8	7	18
Apollo Constellation	1.6	3	24
Busek BIT-3	7.5	44	**67**
Enpulsion IFM Nano x2	9.6	23	28
IFM Nano (max I_{sp}) x2	11	**56**	56
Phase Four x2	6.1	8	16
VACCO Green MiPS	14	7	-
VACCO/JPL MarCO	27	3	-

Table 2.3: Table of a subset of propulsion system capabilities for a 12U LGS spacecraft. The left entry in the maneuver count column is the number of maneuvers each propulsion system can sustain at maximum thrust, corresponding to the minimum transit time in the column to the far left. The right entry corresponds to the number of maneuvers achievable at reduced thrust, such that all systems complete maneuvers in the same amount of time as the slowest electric propulsion system (*i.e.*, the Enpulsion IFM Nano's maximum specific impulse). The VACCO propulsion systems cannot perform the maneuver in 11 days even at full thrust, and so their entries in the second column are empty.

2.4.2 Propulsion system trade

Using the subset of available propulsion systems suitable for a 12U CubeSat in Table 2.2 and Equations 2.3 and 2.4, we can calculate how many days are required to execute the 11-degree "standard maneuver", and how many of those maneuvers can be achieved by each propulsion system. The complete results are listed in Table 2.3, where we can see that, under the assumption that all thrusters operate at maximum thrust and their design specific impulse (*i.e.*, no performance degradation is modeled and no margin is set aside), the Enpulsion IFM Nano thruster in its maximum-I_{sp} design point is the one capable of the most maneuvers (56). However, it requires 11 days to execute a maneuver of 11 degrees, which is the slowest of all the electric propulsion options. If we allow the thrusters to operate at reduced thrust, to match the acceleration and maneuver time of the IFM Nano thrusters, the Busek BIT-3 has the greatest maneuver capacity (67). We will therefore baseline the Busek BIT-3 in the 12U LGS spacecraft design. From the perspective of spacecraft longevity, deep space operation, and heritage, the BIT-3 has the further advantage of having been selected for use on the upcoming LunaH-Map and Lunar IceCube missions. [24, 12]

The BIT-3 and IFM Nano thrusters both have the disadvantage of using solid propellants, which must be melted before they can produce thrust. This means that they have extra power and thermal loads, and it means that the spacecraft must either keep them on 'hot standby' or schedule all maneuvers with at least an hour in advance for heating. However, it is because they have solid propellants that they can carry so much fuel relative to their dry mass and achieve such high delta-V figures. The Accion TILE thrusters instead have a liquid propellant carried in a porous metal substrate, which leads to a much higher dry mass for the amount of propellant they carry. On the other hand, the TILE thruster does not require any preheating. Smaller TILE units may be useful in a more detailed design as a secondary "reaction control

[8] This is the case for spacecraft where the fuel mass is much less than the total spacecraft mass. For the LGS spacecraft design considered here, with the propulsion system options in Table 2.2, fuel mass is never more than 12% of the total spacecraft mass.

[9] Eventually, the maneuver lasts so long and thrust is so low that the spacecraft is more affected by the dynamics of L2 than its own thruster. As discussed in Section 3.2, the thrust required to counteract the dynamics of L2 is, on average, 7% of the maximum throttle of the Busek BIT-3.

system" (RCS) if the spacecraft needs additional agility.

2.4.3 Communication

As discussed in Section 3.3 (thruster and sensor uncertainty requirements), the LGS spacecraft will not need to check-in frequently with the host telescope, nor exchange large volumes of data – an update on their relative position and velocity once every five minutes will suffice. For the 12U LUVOIR companion design, the LGS spacecraft has been baselined to include the JPL Iris communications system, selected for its heritage in deep-space CubeSat missions such as MarCO and upcoming LunaH-Map and Lunar IceCube. [7] MarCO used a large deployable 60-cm high-gain antenna to close a link to the Deep Space Network (34- and 70-m dishes) at a distance of nearly 2 AU (300 million km); the factor of 200 reduction in distance to L2 (0.01 AU, 1.5 million km) means the LGS will be able to close the same link with a patch antenna onboard and a 5-meter dish on Earth, and the 43,000-km link to the telescope can be closed by a pair of patch antennas.

2.4.4 Power

The two high-power components that will be used regularly by the LGS spacecraft are the propulsion system, with a maximum power usage of 75 W [13] (but, as discussed in Chapters 3 and 4, it need not be used above 30% throttle), and the laser, which will likely use a maximum of 30 W. [70] Taken together, the average power draw of the LGS spacecraft during a worst-case observation will be 73.5 W. To sustain that power output for the multiple hours of an observation, the LGS spacecraft nominally can use a COTS system, such as the GOMspace P60 modular power system, with two BPX battery modules (total capacity: 154 Whr). [6]

With 4 single-deployable solar panels on the 6U faces, plus one set of body solar panels on the 4U face not occupied by the thrusters (the +X face in Figure 2-6), the maximum power that can be generated by the spacecraft is 85 W, if the telescope is observing a target at a right angle to the Sun vector.[10] This means that the LGS spacecraft will be power-positive if the target line-of-sight is angled up to 30 degrees towards or away from the Sun, and capable of sustaining an 11-hour observation (longer the 75% of all desired observations [57]) if the target line-of-sight is angled up to 45 degrees towards or away from the Sun, at least under beginning-of-life conditions.

If, as the design matures, additional power capacity is required, the simplest approach may be to increase the length of the vehicle to a 16U CubeSat. The larger bus size will have more surface area for solar panels, which will enable longer mission lifetimes by compensating for solar panel degradation, and more volume for batteries to enable longer observations or observations where the solar panels must be angled further away from the Sun. The implications for propulsion system capacity are briefly discussed in Appendix A.1.

2.4.5 Attitude Determination and Control

As discussed in Section 3.3 (Thruster and sensor uncertainty requirements), the LGS spacecraft will not be subject to strict body-pointing requirements during an observation, as long as the telescope can provide it with regular updates on its relative position and velocity. The LGS baseline design incorporates a COTS reaction wheel assembly such as the BCT XACT, both for its demonstration of sub-arcminute-class accuracy on-orbit [39] and its deep-space heritage on MarCO. [59]

Because the LGS spacecraft will be flying in a halo orbit at L2 and not low Earth orbit, there will be no disturbance torques from Earth's atmosphere, gravity gradient, or magnetic field, but there will also be no possibility of using Earth's magnetic field to desaturate its reaction wheels. The worst-case solar radiation pressure (SRP) torque will be $0.14\,\mu$Nm, exerted on the deployable solar panels when they are oriented at 45° to the Sun. While transiting between targets, the LGS spacecraft will flip over halfway and can choose its orientation to cancel out any torque accumulation, but

[10] As discussed in Figure 3-3, this is the preferred configuration for minimum fuel cost during an observation.

this will not be an option during an observation. The longest desired observation is nearly 18 hours [57], which would result in accumulating up to 0.0092 Nms of angular momentum; BCT's RWP-100 reaction wheels can store up to 0.1 Nms of angular momentum, so there is no need to desaturate during any observation (even up to 8 days in duration), but that momentum must be shed eventually. The LGS spacecraft will do this with the BIT-3 thruster, which can gimbal up to 10 degrees; at that angle, and at the nominal thrust for transiting between stars (19%, see 4.2.2), the thruster can produce 6.1 μNm of torque, or over 40 times the worst-case SRP torque, for a loss of propulsion efficiency of less than 2%. The accumulated angular momentum will be dissipated within the first minutes of a many-day transit maneuver.

2.4.6 12U LGS Space-Space L2/LUVOIR point design assumptions

- Telescope: 9.2 m diameter, observing at $\lambda = 500$ nm
- Range to telescope: 43,184 km (sometimes referred to as "43,000 km")
- Total mass: 14.4 kg (incl. 1.5 kg iodine propellant in BIT-3)
- Volume: 12U ($22.6 \times 22.6 \times 34.1$ cm)
- RF communication system: *e.g.* JPL Iris (X-band), crosslink to telescope and downlink to DSN[11]
- ADCS: COTS, such as BCT XACT-100 (+3x RWP-100 reaction wheels)
- Propulsion system: COTS, such as Busek BIT-3, 1.24 mN, 2.48 km/s delta-V.
- Power: COTS, such as GOMspace P60
- Laser system: 5 W, 3.5 cm aperture. Two wavelengths selected, adjust for instrument applications to offer out-of-band options:
  - 532 nm: 34 μrad = 7" beamwidth (FWHM), apparent magnitude -5
  - 980 nm: 63 μrad = 13" beamwidth (FWHM), apparent magnitude -7

[11]The rate of communication is not likely to require DSN specifically; NEN dishes can close the link to alleviate DSN usage.

Chapter 3

Formation Flight of a LGS with Space Telescope at L2

In this chapter, we will consider the formation flight of a laser guide star spacecraft with a space telescope at L2. Architectures with the laser guide star in an Earth orbit (GEO LGS with ground-based telescope or GEO space telescope) are discussed in Chapter 5. Architectures where the "laser guide star" is not actually carrying its own laser, but passively reflects a laser generated elsewhere, are discussed in Appendix A.2.

We will use the frame of reference illustrated in Figure 3-1, with its origin at the telescope, z-axis towards the target star and x-axis aligned with the difference in gravitational acceleration between the telescope and the LGS. In an ideal observation, the LGS will have its thrust vector aligned against the net L2 disturbance force, and the LGS spacecraft will be exactly on the line of sight from the space telescope to the target star.

The main line-of-sight requirements were previously calculated in Section 2.2, but as a brief reminder, they are reprinted in Table 3.1.

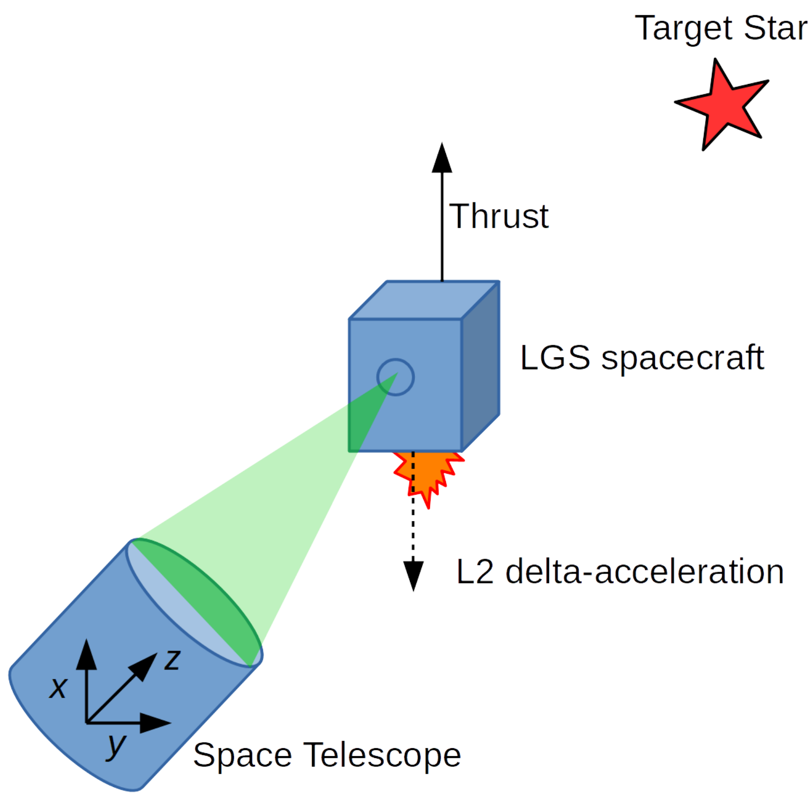


Figure 3-1: The frame of reference used in discussing the telescope-LGS formation flight. Origin at the telescope, z-axis towards the target star, x-axis aligned with the difference in gravitational acceleration between the telescope and the LGS.

Parameter	**Value**	**Notes**
Range	43,000 km	"At infinity", flat wavefronts on the telescope
Off-axis error	37 m	No waves across a segment
	2.3 m	Stay inside coronagraph IWA (λ/D)
	0.6 m	Stay **deep** inside coronagraph IWA ($0.25\lambda/D$)

Table 3.1: Top-level formation flight requirements for LGS in front of LUVOIR.

3.1 Mitigating LGS impacts on observation

Beyond the needs of the wavefront sensor, another reason to stay as close to the telescope-star axis as possible is to remain behind the coronagraph mask to mitigate negative effects on the observation. We will consider an observation in the visible part of the spectrum (center wavelength 500 nm). Besides using a laser which lies outside the science band (the 980 nm laser), as described in Section 2.3.1, we will evaluate the LGS spacecraft's thruster plumes, sunlight glinting from its body, and thermal emissions.

3.1.1 Plumes

For the 12U CubeSat LGS design, the top thruster candidates are electric propulsion systems with exhaust velocities in excess of 15 km/s. If the thruster is shut off, the plume will leave the outer working angle of the coronagraph ($24\lambda/D = 1.3$ μrad [47]) in less than 4 ms. As the worst-case thrust required to sustain formation flight is less than 1/3 of the BIT-3's maximum thrust (calculated in Section 3.2), and on average requires less than 7% thrust, it will be straightforward for the LGS to pulse its thrusters and coordinate with the telescope to integrate between impulses.

If the LGS spacecraft is built to a larger form factor with monopropellant thrusters, then its exhaust products will remain in the coronagraph's field of view for over 30 ms, even though the LGS spacecraft should be able to use shorter pulses to hold the formation.

3.1.2 Reflected sunlight

During an observation, the sides of the LGS spacecraft will either be facing the telescope aperture directly, or at right angles to it. As shown in Figure 3-2, there will be no direct reflections from the Sun into the telescope. This still leaves the question of scattered light from the LGS spacecraft's edges. Steeves *et al.* [58] have measured the light glinting from sharp aluminum edges and found that the total glinting from the Starshade will be between 22nd and 26th magnitude, depending on the angle to the Sun. Scaling from the perimeter of Starshade (≈400 m of edges) down to a 12U bus (up to 5 m of edges, with dual-deployed solar panels), and moving from 48,800 km inwards to 43,000 km, LGS would have a glint between 26th and 30th magnitude (23rd-27th magnitude at 10,000 km). This is comparable in brightness to an Earth-like planet around a 5th-magnitude star; even though the telescope should know where the LGS spacecraft is located and be able to ignore a spurious signal, it could obscure the observation of an actual planet, so it would be desirable to fly within the inner working angle of the coronagraph.

If the LGS spacecraft happens to lose attitude control and turns so that one of its faces reflects a sunbeam directly into LUVOIR, the resulting glint will be magnitude 1.. This could be potentially disturbing to the observation if the LGS also loses positional control and leaves the coronagraph mask, but it would not be damaging to the telescope.

3.1.3 Thermal emission

If the LGS spacecraft is to assist during an infrared observation, its thermal emission may be a source of confusion. Based on an estimated maximum power budget of 100 W, and assuming that the face with the laser is covered in aluminum (for its low infrared emissivity), the spacecraft would appear as bright as magnitude 26 in the K band. The average star in the LUVOIR target list has magnitude 4 in K band, so the "contrast" between the LGS spacecraft and the target star will be approximately 10^{-9}, comparable to the contrast between Jupiter and the Sun; this will be well within the capabilities of the coronagraph to suppress.

3.2 Thrust requirements for formation flight

The non-dimensionalized circular restricted three-body problem is used to calculate the thrust required to sustain the telescope-LGS formation flight at L2. After a nominal orbit was computed for LUVOIR (see Figure 2-1), a test particle was placed at the desired range (43,000 km) in all directions, at all times in the six-month

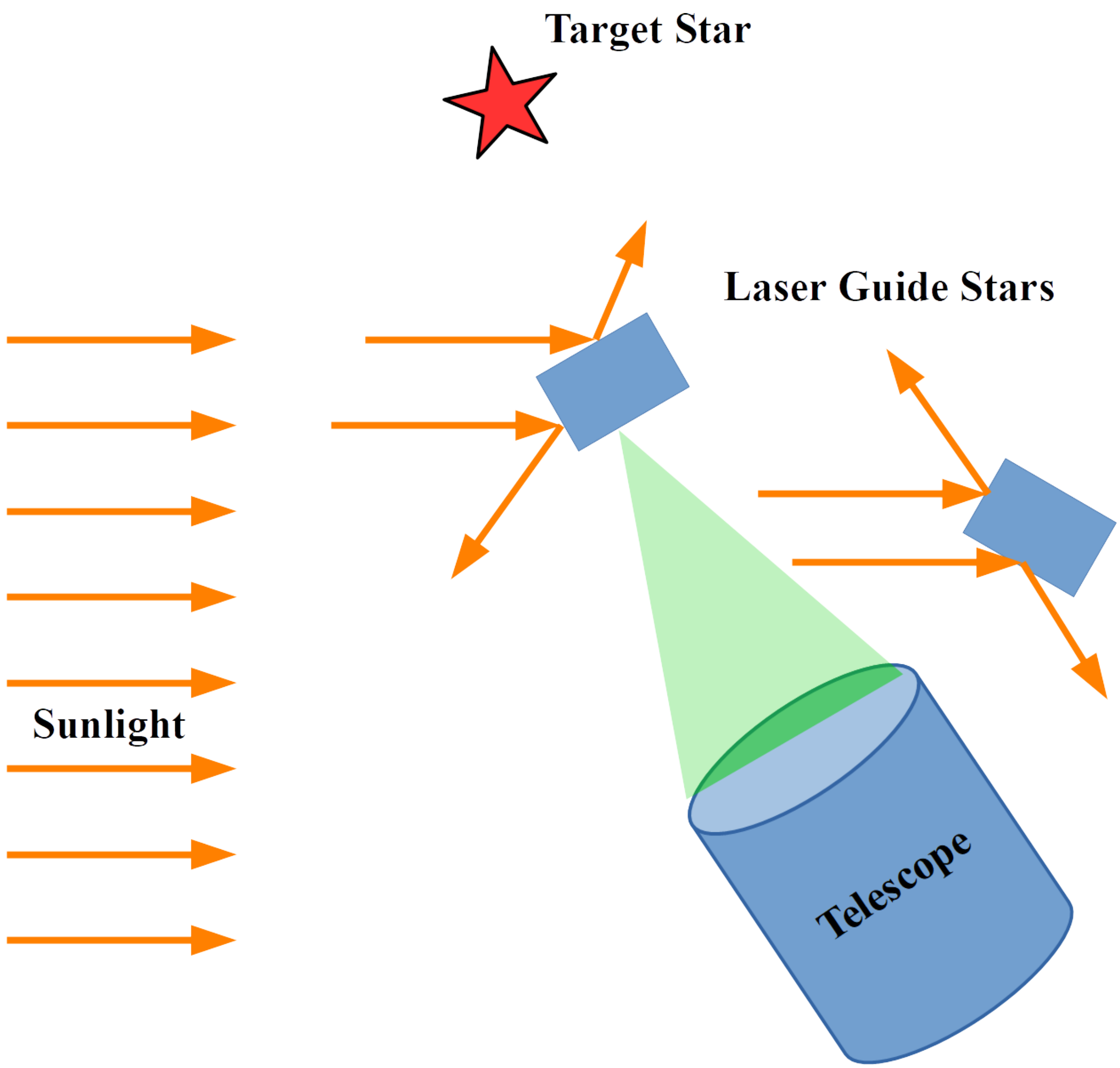


Figure 3-2: As long as the telescope-LGS line of sight is not facing directly towards or away from the Sun, there will be no direct reflections from the Sun into the telescope from any of the LGS's faces.

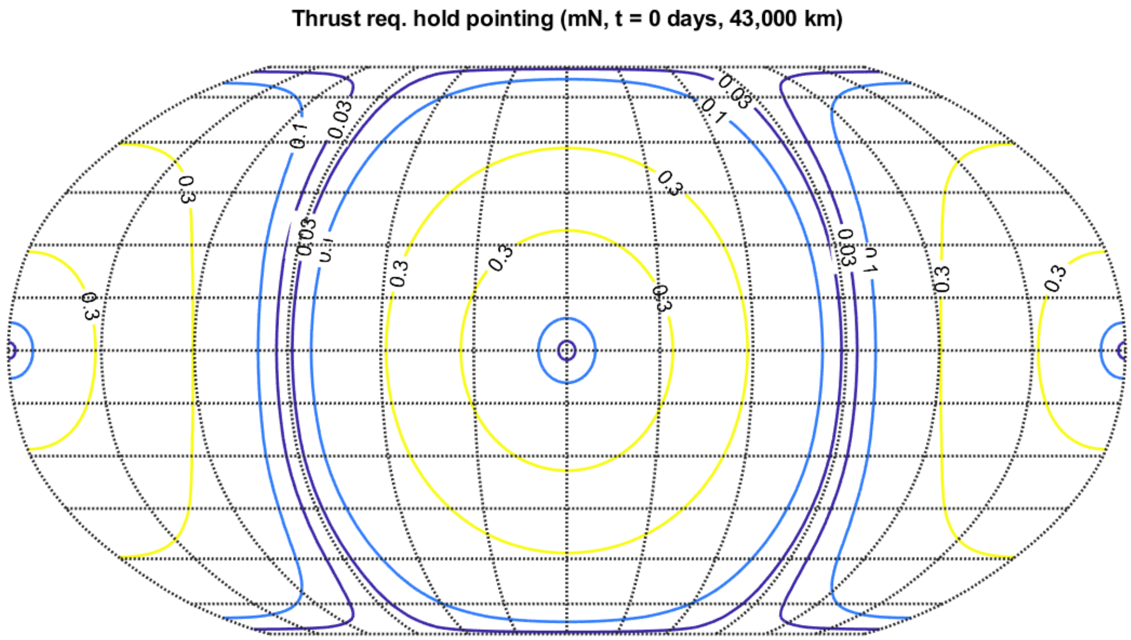


Figure 3-3: Thrust required to sustain the telescope-LGS line of sight in any direction, at time $t = 0$ (LUVOIR closest to Earth). The center of the diagram corresponds to looking directly away from the Sun; the left and right edges of the map are looking towards the Sun.

orbital period. The difference in acceleration between LUVOIR and the test particle is calculated at all points, and the component of that acceleration perpendicular to the line of sight between them is the thrust required (when multiplied by the spacecraft mass). Two "slices" through this space are shown here: Figure 3-3 shows the required thrust to observe in any particular direction at elapsed time $t = 0$ (when LUVOIR is closest to Earth), and Figure 3-4 shows the required thrust to observe in any direction in the equatorial plane, over the course of the six-month orbital period.

From these figures, we can see that it is generally least expensive (in terms of fuel cost) to support observations in alignment with or perpendicular to the Sun-Earth axis, and that the most expensive observations are those which take place when LUVOIR is closest to Earth, moving the fastest. The worst-case maximum thrust required to sustain the Telescope-LGS line of sight is 0.36 mN (29% of the BIT-3's maximum thrust), and the average thrust required, over the whole sky and orbital period, is 0.085 mN (85 μN, 7% of the BIT-3's maximum thrust).

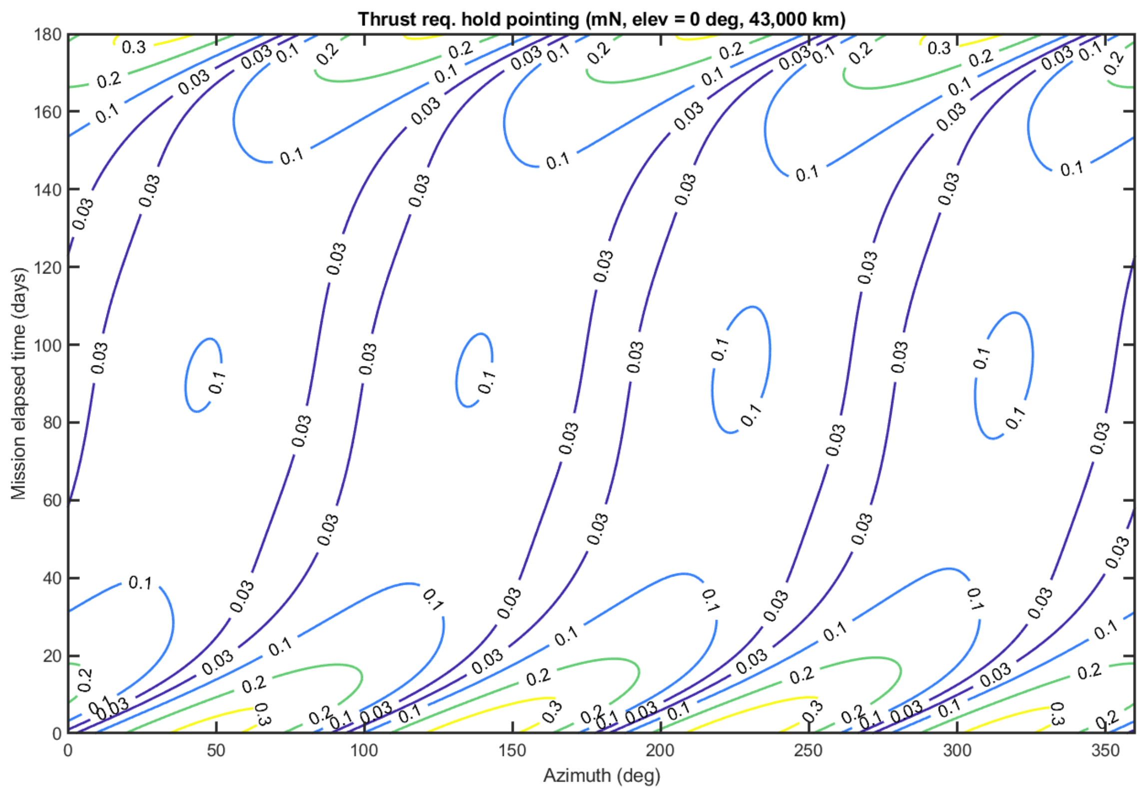


Figure 3-4: Thrust required to sustain the telescope-LGS line of sight at any azimuth within the ecliptic plane, over the course of the six-month halo orbit period. Inertial frame of reference, zero degrees azimuth corresponds to the Sun-Earth(-LUVOIR) vector at $t = 0$.

Because the thruster is mounted perpendicular to the laser, the difference in acceleration parallel to the line of sight cannot be corrected with the reference LGS design. As long as the LGS spacecraft only drifts slowly away from the telescope, the quality of the observation will not be significantly affected.[1] A simulation of a 24-hour observation (longer than any of LUVOIR's nominal observations) is performed to quantify the drift; in that time, the LGS spacecraft will drift only 100 km away from the telescope, or approximately 0.2% of the nominal range to the telescope (a reduction in brightness of 0.017 dB, or 0.004 magnitudes). Therefore, we can neglect this drift when developing the first-order constellation planner and mission scheduling in Chapter 4.

3.3 Thruster and sensor uncertainty requirements

Given the maximum required thrust to sustain the formation calculated in Section 3.2, 0.36 mN, we can place upper bounds on the uncertainty requirements for the LGS spacecraft's thrusters and sensors.

The first requirement is to determine how frequently the telescope should update the LGS on its relative position. Because the main disturbance is aligned with the thruster axis (the x-axis, in the frame of reference illustrated in Figure 3-1), it is expected that the fastest drift will be in that axis. It is also the easiest to adjust; for example, if the spacecraft has drifted towards the top of its target range (*i.e.* $x > 0$), then it can reduce thrust below the steady-state value, settle towards the line of sight, and then increase thrust above the steady-state value to cancel out its relative velocity. As long as the difference in acceleration required is less than the disturbance acceleration, this maneuver has no net cost of fuel compared to steady-state formation holding. However, if the spacecraft is required to accelerate faster than the disturbance acceleration, then it will need to turn and use its engine instead of reducing thrust and letting "gravity" do the work, which will out-pace the nom-

[1] Increasing the range will make the laser's wavefronts flatter, which means they will serve as a better reference for the telescope, but the laser will also become dimmer.

inal fuel budget. Therefore, from a fuel budget perspective, it is ideal to make the update period as long as possible, so that the spacecraft can take advantage of local acceleration rather than have to expend extra fuel to "sprint" into position.

For some maximum background acceleration of $a_{max} = 2.5 \times 10^{-5}$ m/s^2 (given the above maximum thrust and the spacecraft's mass of 14.4 kg), the maximum distance d_{zc} that can be transited for no additional cost in update interval t_{upd} is given in Equation 3.1. This curve is plotted in Figure 3-5 and compared to the target zone radius values calculated in Section 2.2. We can see that if the LGS spacecraft is to stay within the deepest part of the coronagraph IWA during an observation that requires maximum thrust, its update interval should be no longer than 5 minutes. For the average observation, that requirement relaxes to 10 minutes, or even to 20 minutes if the looser $1\lambda/D$ constraint (assuming LUVOIR installs additional notch filters for the laser wavelength, see Section 2.2.1) is feasible. In all cases, this is much longer than the 144 ms single-way light delay between the two spacecraft, so the telescope's updates will not become outdated in transmission.

$$d_{zc} = \frac{1}{4} a_{max} t_{upd}^2 \tag{3.1}$$

Another key metric is the limit on steady-state thrust error. This thrust error should be no greater than the value such that the spacecraft will drift half of the radius of the target zone within the update period, so that the drift can be corrected and halted within the next update period. This turns out to be exactly 1/4th of the steady-state thrust, independent of the size of the target zone or the update period. That magnitude of error is not the kind that results from electrical system noise; on-orbit electric propulsion system demonstrations show oscillations in thrust on the order of a few percent at most. [33] A thrust error of 25% would be indicative of some kind of malfunction of the spacecraft itself. If testing shows that this level of thrust control is ever a challenge to achieve, for example during an observation towards the poles or otherwise requiring very little thrust, then smaller thrusters could be installed to act as a fine-control RCS, such as Accion's Tiled Ionic Liquid Electrospray (TILE)

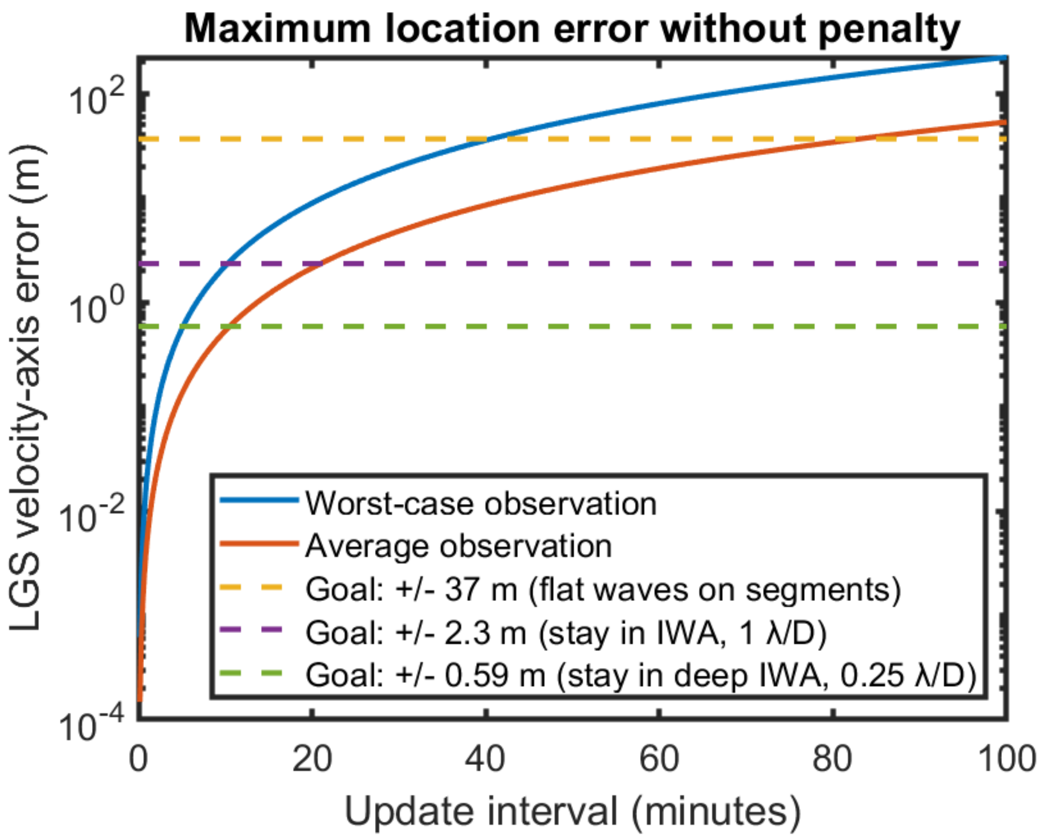


Figure 3-5: Maximum distance that can be transited for no additional cost over steady-state formation flight, versus the update interval.

modules.

The 25% maximum thrust error leads to the requirement on alignment of the thrust vector: the arcsine of 0.25, or 14 degrees (0.253 radians). Again, this value is independent of the size of the target zone, and is also easily achieved by any commercially available star-tracker. The BCT XACT has been verified on-orbit as controlling a spacecraft's attitude to less than 20 arcseconds RMS. [39]

From this, we can conclude that the LGS spacecraft ADCS needs are not challenging to meet. The strictest requirements will fall on the telescope's ability to track the LGS with milliarcsecond precision, although it will have several minutes available to filter and process each update, and the LGS will also be actively cooperating (and very bright).

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

LGS Constellation Design

Now that it has been established that the LGS spacecraft design developed in Chapter 2 is suitable for supporting individual observations with LUVOIR or similar space telescopes, it is possible to determine just how many of them are required to support the quantity and pace of observations desired by LUVOIR mission planners: 1,539 observations of 259 stars, carried out within five years. [57]

4.1 Constellation sizing

A *constellation sizing tool* was developed to calculate the number of LGS spacecraft with the mass, fuel capacity, thrust, specific impulse, and range to the telescope required to support an observation mission of given parameters (number of stars and observations, time required for each observation, and total mission duration). First, the Fibonacci spiral (a mathematical shape that produces a nearly-uniform distribution of points over a sphere [61]) is used to compute an expected angular separation between adjacent stars, as shown in Figure 2-8b. The desired number of observations is then divided into a number of domains. The telescope's desired pace of observations, the number of observations divided by the total mission duration, is multiplied by the number of domains to calculate the time available for each individual LGS spacecraft to transit from one star to the next (it is assumed that all LGS spacecraft stagger their maneuvers). That available maneuver time is used to set the throttle level of the transit maneuver, to maximize fuel savings by minimizing thrust used (as in Equation 2.4). Finally, the integration time from the LUVOIR target schedule [57] multiplied by the maximum acceleration from the formation-flight calculations (25 μm/s^2, see Section 3.2) is added to obtain the total delta-V cost per observation.

That cost is divided into the delta-V capacity of one LGS spacecraft to determine how many observations each LGS can support, and that number is divided into the number of observations within the domain to determine how many LGS spacecraft are required to support the full observation campaign. The sizing tool iterates over a range of domain sizes to find the value where the number of required LGS spacecraft is minimized. That value is the one such that each domain is small enough to be completely serviced by one LGS spacecraft (with no margin, although a margin requirement can be levied by reducing the delta-V capacity and/or maximum thrust of the LGS spacecraft).

The constellation sizer is run on all propulsion system options outlined in Table 2.2, both in the case where the baselined mass is used (11.5 kg plus the mass of each propulsion system) and a second case where the total LGS spacecraft mass is assumed to be 24 kg, the maximum value permitted for a 12U CubeSat. The resulting chart is shown in Figure 4-1, and it shows that the Busek BIT-3 results in the fewest LGS spacecraft required (19). This is in alignment with Table 2.3, which identified the BIT-3 as the thruster with the greatest delta-V capacity and the greatest maneuver capacity of the options considered.

4.1.1 Deployment and disposal

The constellation planning and scheduling tools do not presently consider the time and fuel cost of deploying LGS spacecraft, nor the manner in which they are deployed. If the number of LGS spacecraft required is small enough, such as in the single-digits (if LUVOIR can be modified to allow the LGS spacecraft to fly much closer than 43,000 km), they could be deployed from LUVOIR itself, but otherwise, it might be preferable to have the LGS spacecraft launched separately to minimize operations

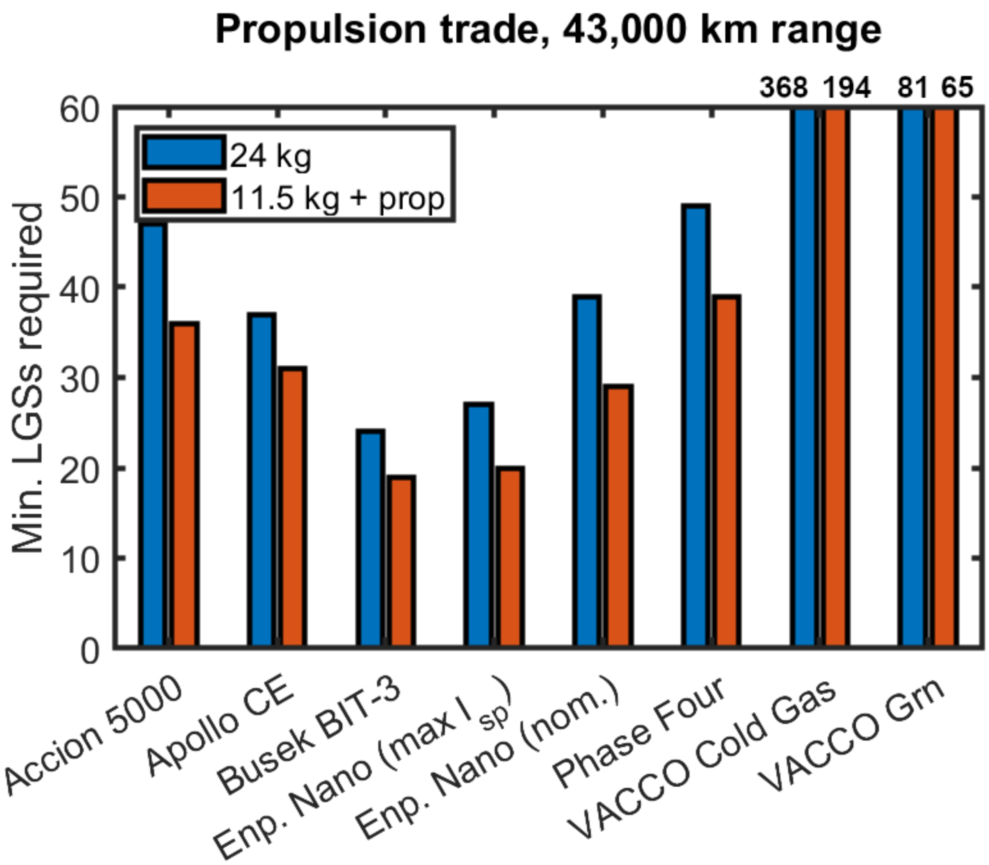


Figure 4-1: Number of LGS spacecraft estimated by the constellation sizing tool as necessary to support the LUVOIR campaign (259 stars, 1,539 observations) [57] for a selection of propulsion system options and 2 LGS mass values.

in close proximity to LUVOIR. This could be done by storing them in a secondary carrier or depot spacecraft that is launched with LUVOIR to the same L2 halo orbit, but which leads or trails LUVOIR by 43,000 km (approximately two days).

If the LGS spacecraft are deployed from LUVOIR itself, then they will drift away because of the natural instability of L2. This will allow them to reach the desired range without having to activate their thrusters in proximity to LUVOIR. If they are deployed with a separation velocity of 1.44 m/s [48], then they will drift 43,000 km away after 90 days, and the delta-V cost to arrest its drift will be 18 m/s (less than 1% of the total delta-V budget), achieved by 2 days of thrusting. After their propellant is depleted, they will drift away from L2 into a heliocentric orbit without active propulsion.

The benefit of the depot is that it can be used as a staging ground for launching new LGS spacecraft partway through the mission, if there are unexpected failures, or if the schedule calls for the deployment of new LGS spacecraft partway through the mission (see the "Number of simultaneously active LGS spacecraft" trade). Perhaps a mission to service LUVOIR's instruments (which is expected to be possible [28]) could deliver new LGS spacecraft to the depot after departing LUVOIR.

If the depot can be made even more advanced, like the "locker" system proposed by Ezinne Uzo-Okoro *et al.* [65], then LGS spacecraft can visit the depot for refueling[1] instead of being discarded as space junk. Even if there aren't enough LGS spacecraft required for LUVOIR's initial mission to make the depot worthwhile, it might be useful as an asset which can remain on-station to support mission extensions or to support other telescopes launched to L2.

4.1.2 Constellation size sensitivity

After computing the nominal constellation size, sensitivity studies were carried out over several of the variables to assess their relative importance to the result for the

[1]Because both the Busek BIT-3 and Enpulsion IFM Nano use propellants which are solid at room temperature, they wouldn't be refueled in the conventional sense, but would instead have their entire propulsion systems replaced.

LGS concept.

Range

One of the key parameters governing the LGS architecture is the range between the LGS spacecraft and the telescope it supports. The value used in this work, 43,184 km, is calculated so that the laser's wavefront curvature across the telescope's mirror is no more than a quarter-wave (see Equation 2.1). At 43,184 kg, the laser's light is "close enough to infinity" to focus like the target star's light and not require the telescope to have additional flattening or compensation optics. However, as shown in Figure 4-2, the closer the LGS is allowed to fly to the telescope, the less thrust it needs to maintain its formation and the less fuel it needs to transit between observations, and so half as many LGS's are required if they are allowed to fly at a range of 10,000 km instead of 43,000 km. An open trade going forward is to compare the complexity and cost of a defocus-compensation stage within the telescope's WFSC system versus the savings in the number of LGS spacecraft required to support the mission.

Mass

Another trade study was conducted on the effect of changing the LGS spacecraft's total mass, if the current estimate (14.4 kg) turns out to be incorrect – perhaps additional margin for fuel or radiation shielding is required, or additional lasers at multiple wavelengths. Fortunately, the effect turns out to be not quite as strong as the effect of range, as shown in Figure 4-3. Even if the LGS spacecraft's mass must rise to the maximum permitted value, 24 kg, only five additional spacecraft (from 19 to 24 LGS's) will be required to complete the mission.

Fuel

Another trade study was conducted on the effect of changing the LGS spacecraft's fuel mass, leaving the rest of the spacecraft's parts the same mass (which is slightly optimistic, as the propulsion system's mass would probably also have to increase slightly to accommodate the increased propellant capacity). The effect of doubling

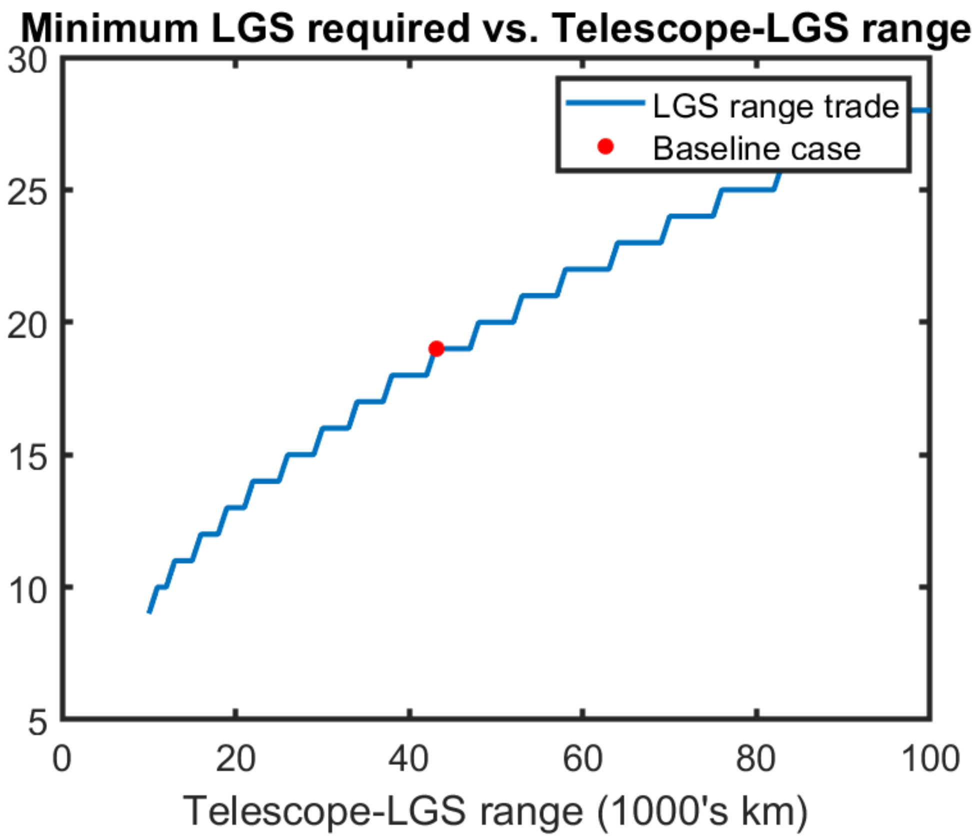


Figure 4-2: Number of LGS spacecraft required to support LUVOIR's observation campaign vs. the range to the telescope.

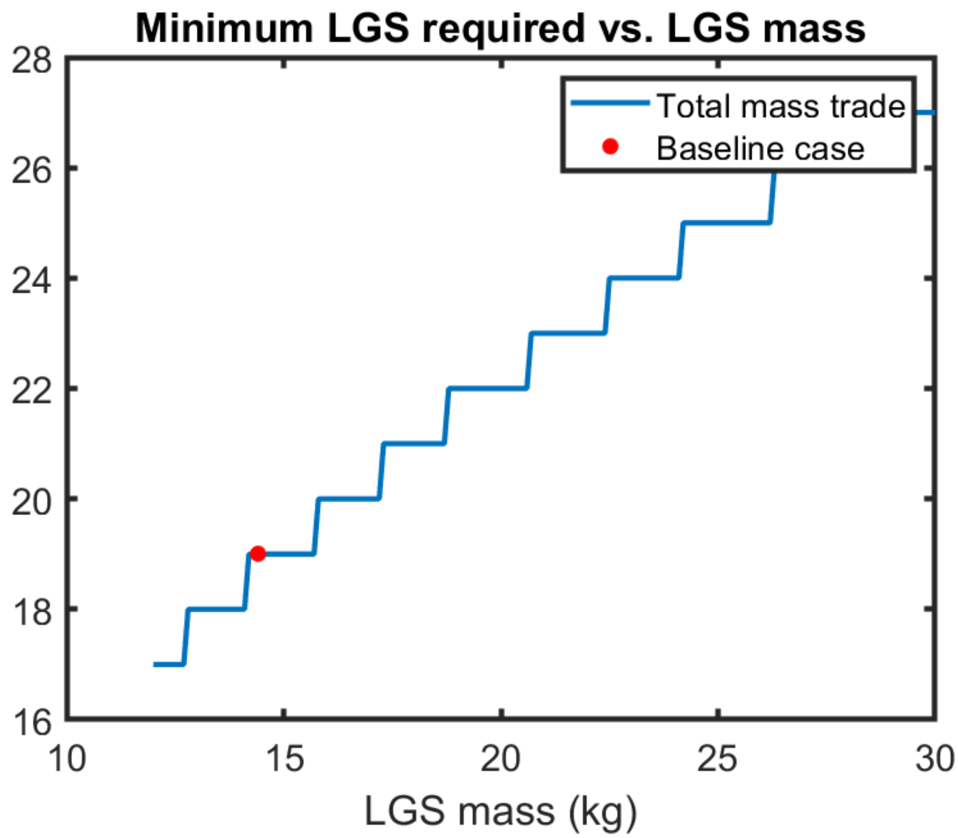


Figure 4-3: Number of LGS spacecraft required to support LUVOIR's observation campaign vs. the total mass of each LGS spacecraft..

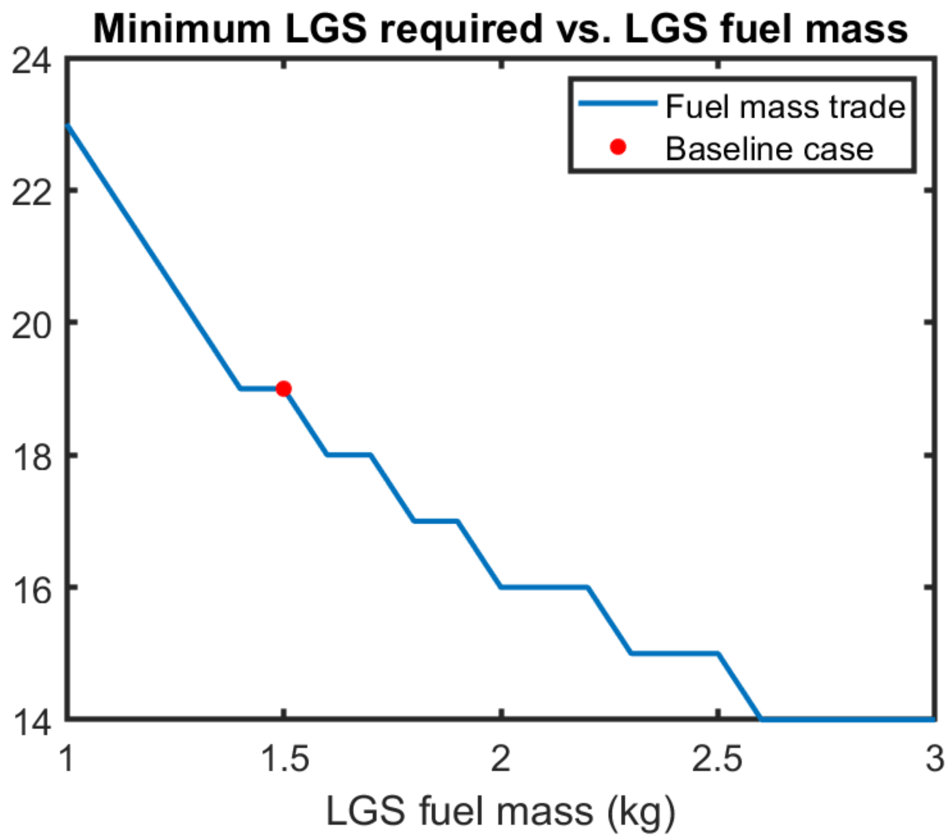


Figure 4-4: Number of LGS spacecraft required to support LUVOIR's observation campaign vs. the propulsion system's fuel mass.

the BIT-3 thruster's fuel mass from 1.5 kg to 3 kg is shown in Figure 4-4, reducing the number of spacecraft required from 19 to 14. Such a change would require some additional non-recurring engineering (NRE) to be paid for changing the BIT-3 from its current configuration, which would have to be weighed against the savings of not having to build so many LGS spacecraft (and especially their expensive propulsion systems and testing campaigns).

Number of simultaneously active LGS spacecraft

The final trade was to study the effect of the maximum number of LGS spacecraft operating simultaneously on the total number of LGS spacecraft required (Figure 4-5) and the total time required to complete the mission (Figure 4-6). From Figure 4-5, we can see that the number of LGS spacecraft required to complete the mission increases as fewer of them are allowed to be active at once, which forces them to operate at

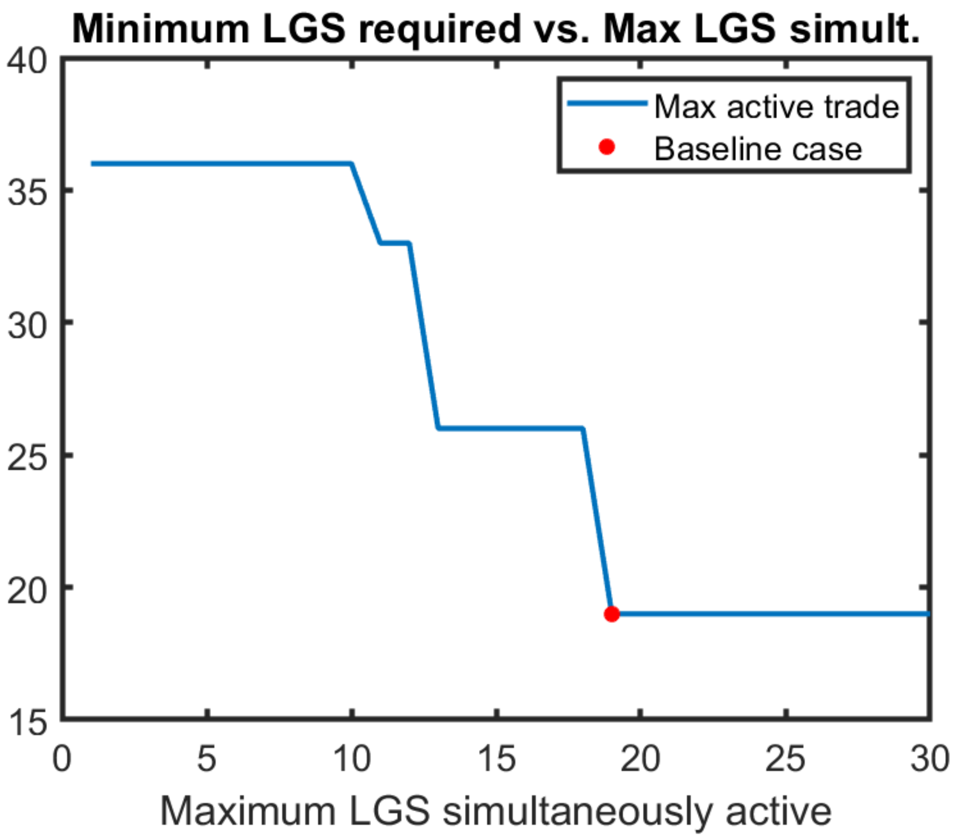


Figure 4-5: Number of LGS spacecraft required to support LUVOIR's observation campaign vs. the number of simultaneously-active LGS spacecraft.

higher thrust and greater fuel consumption to (attempt to) satisfy LUVOIR's observation schedule, but that for 10 domains or fewer, no more than 36 LGS spacecraft are required, because they are moving at maximum thrust and no further trade of fuel for time can be made. Instead, as shown in Figure 4-6, the total time required to execute all 1,539 observations starts to rise – requiring over 30 years (!), not shown on the chart due to space constraints, if all of the observations can only be supported by a single LGS spacecraft at a time.

From this trade study, we see that the constellation can satisfy the five-year time constraint all the way down to seven simultaneous guide stars, which offers an interesting trade between total manufacturing and testing cost against the operational challenges of managing the constellation in flight – if, for example, coordinating all nineteen LGS spacecraft plus LUVOIR at once is too challenging, the number of domains can be reduced accordingly with minimal schedule impact.

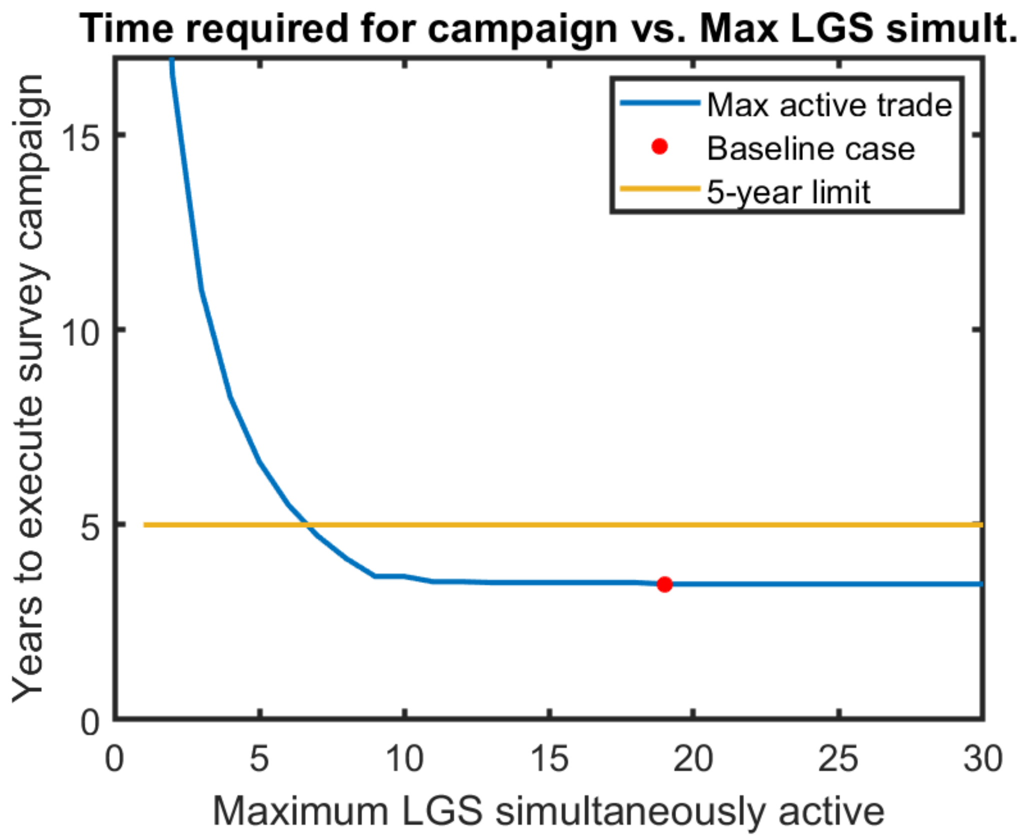


Figure 4-6: Time required to support LUVOIR's observation campaign vs. the number of simultaneously-active LGS spacecraft.

4.1.3 Mission enhancement

The total integration time of all of the desired exoplanet observations with LGS is 585 days, or just under a third of the five-year mission duration, so there is room in the schedule for LUVOIR to make observations for other purposes (*e.g.* imaging bodies within our Solar System, active galactic nuclei, or other targets of interest). One final constellation calculation was run to see how many LGS spacecraft would be required to support twice as many observations of twice as many stars (so 518 stars and 3,078 observations) in the same five-year mission duration. It was found at 43 LGS spacecraft would be required.

4.2 Validating the constellation sizer

The previous analyses were all done under the assumption that the targets of observation were evenly distributed over the sky, in the pattern of a Fibonacci spiral. However, actual stars are clustered, which should allow fuel savings, and hopefully will require fewer LGS spacecraft than were predicted in Section 4.1. To validate the constellation sizing tool, the list of actual stars and observations were processed into schedules for a constellation of LGS spacecraft, by three approaches: scientific priority, geographic segmentation, and a hybrid model.

4.2.1 Scheduling by scientific priority

The list of observations for LUVOIR [57] currently exists in order of scientific priority based on exoplanet yield, organized with the goal of finding and characterizing as many exoplanets as quickly as possible. It does not yet incorporate logistical details, such as Sun keep-out zones, but for the purpose of stress-testing the LGS concept, the list was ingested and processed strictly.

The scheduler works by evenly distributing the observations over the course of the five-year duration, and assigning them in sequence to a series of LGS spacecraft. The first LGS receives the first observation, deducts the fuel cost of the observation (per

Section 3.2), and then scans the list of future observations for the first one that is far enough in the future that it can complete its transit maneuver to be in front of that star before the observation is scheduled to begin. After the LGS either exhausts its fuel supply or reaches the end of the schedule, another LGS is instantiated at the first un-assigned observation, and the process continues until all observations have been assigned. A global throttle parameter is used to govern the thrust with which each LGS spacecraft will perform its transmit maneuvers. At lower values, spacecraft will require more time to complete their transit maneuver, but will be able perform more of them within their delta-V budget. The value of the throttle parameter which results in the fewest LGS spacecraft is 0.23 (*i.e.* transiting between targets at 23% of maximum thrust), requiring 34 LGS spacecraft.

This is substantially more than the 19 baseline predicted in Section 4.1, but examining the schedule and the delta-V expended by each LGS shows that it's not that the LGS spacecraft are performing worse than predicted; rather, the rigidity of the schedule is preventing them from operating at their full potential. Figure 4-7 shows the number of observations supported by each satellite, and it shows that the first 20 LGS satellites were well-utilized, supporting over 50 observations each, but that after that, the scheduler was left with smaller and smaller gaps that each needed a dedicated LGS to fill.[2] The last LGS added only supports a single observation. It may be desirable for a real mission to have some LGS spacecraft that are not fully utilized, to serve as spares or to support urgent unscheduled observations, but to study the lower bound on the *required* number of LGS spacecraft, a second scheduling approach is discussed in Section 4.2.2.

4.2.2 Scheduling by geographic segmentation

In an effort to determine the absolute minimum number of LGS spacecraft required to support LUVOIR's mission targets, a Traveling Salesman Problem solver was applied

[2] More broadly, this is a class of problem known to computer science as the "bin packing problem". The "greedy" scheduler developed here could probably be improved by converting the list of observations and LGS parameters into an optimization problem and turning it over to *e.g.* MATLAB's `intlinprog` solver.

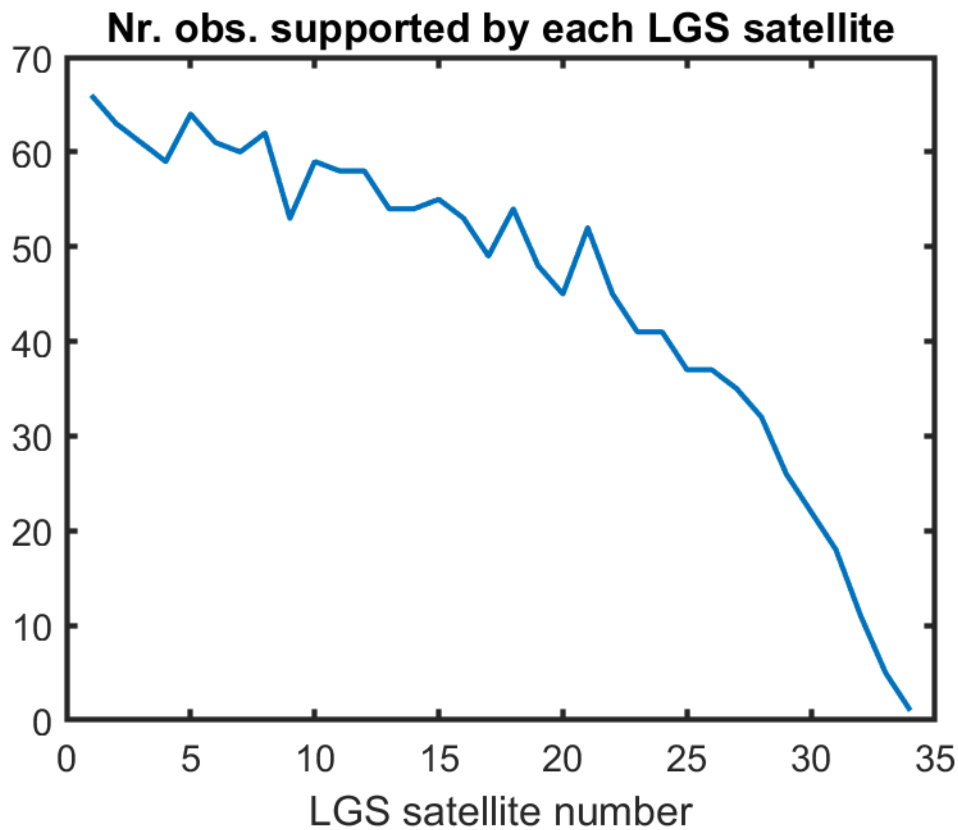


Figure 4-7: Number of observations supported by each LGS spacecraft in the scientific-priority schedule. [57]

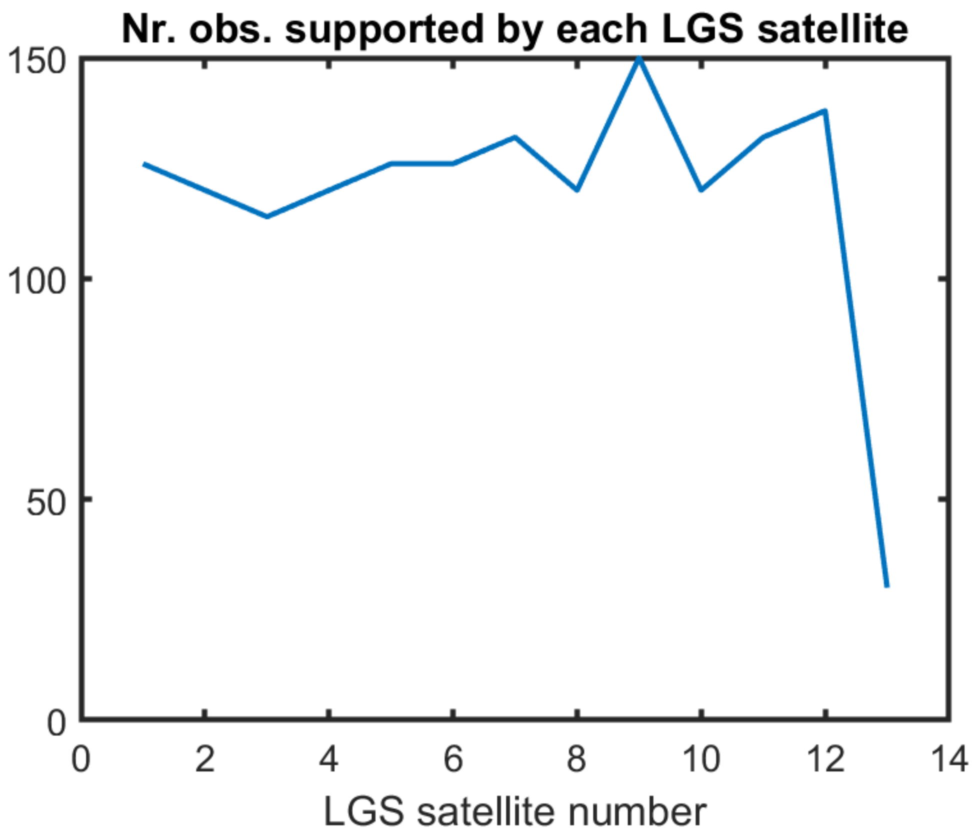


Figure 4-8: Number of observations supported by each LGS spacecraft in the geographic-segmentation schedule.

to the 259 target stars to produce a minimum-cost single track that connects all of them, shown in Figure 2-8a. That path was then split into segments such that one LGS spacecraft could transit from star to star, supporting observations along the way, using one-sixth of its delta-V capacity in ten months. Each would then be able to traverse back and forth along their segments three times (out and back) to support six observations per star in five years. The same global throttle parameter was used to find the optimum utilization point, this time at 19%, and with only 13 LGS spacecraft required – fewer than the uniform-distribution case, as expected. Figure 4-8 shows the number of observations supported by each satellite, and the segments themselves are depicted in Figure 4-9.[3]

Like the list of observations ordered by scientific priority, this schedule does not

[3]The equivalent figure for the scientific-priority scheduler does not look nearly so nice, so it is not included in this dissertation.

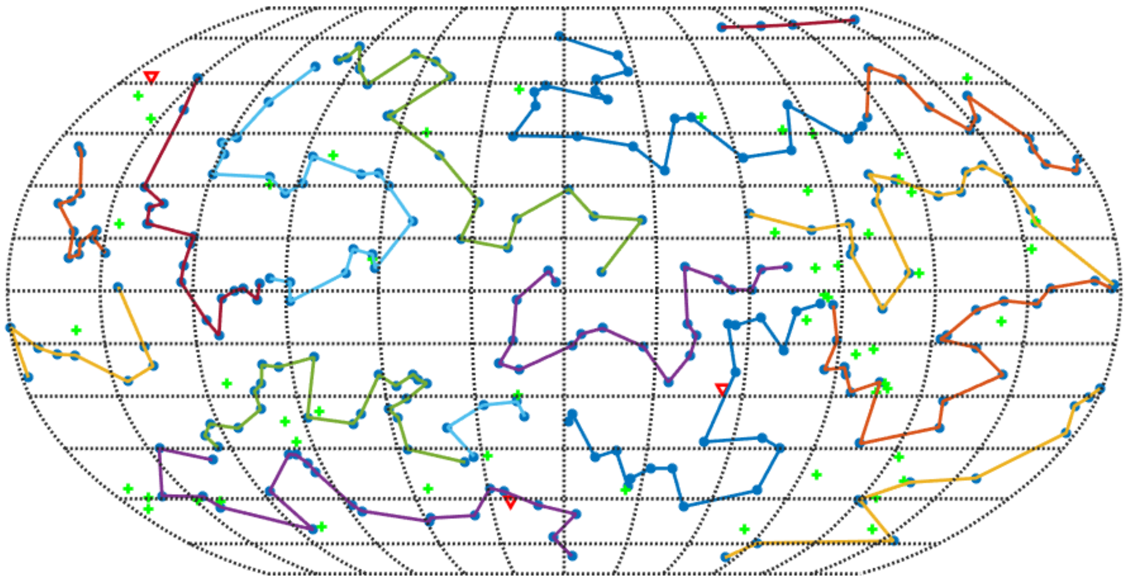


Figure 4-9: Segments traversed by each LGS spacecraft in the geographic-segmentation schedule. There are 13 spacecraft, although some wrap around the the edges of the map and so appear to be two separate tracks.

include details like Sun keep-out zones, and it will also be desirable for some stars to be observed at differing paces, rather than uniformly every ten months, depending on the orbital period of their habitable zones. A full, operational schedule will likely require a number of LGS satellites somewhere in between the two approaches.

4.2.3 "Compromise" scheduling

As an effort to find a compromise between the rigid priority and segmentation schedules, a hybrid scheduler was created that would assign LGS spacecraft to observations that are close to their first assignment. This was intended to obtain some of the best of both worlds: the first few high-priority observations would get dedicated LGS spacecraft, so that they could be observed in rapid succession, and then each LGS spacecraft would be able to stay in its segment of the sky to minimize fuel spent and reduce the total number of LGS spacecraft required. Unfortunately, the scheduler doesn't enforce that the domains should tile over the sky, which results in gaps that need filling, to an even worse extent than the scientific-priority schedule, with 43 LGS spacecraft required to support the entire mission. However, as Figures 4-10 and 4-11

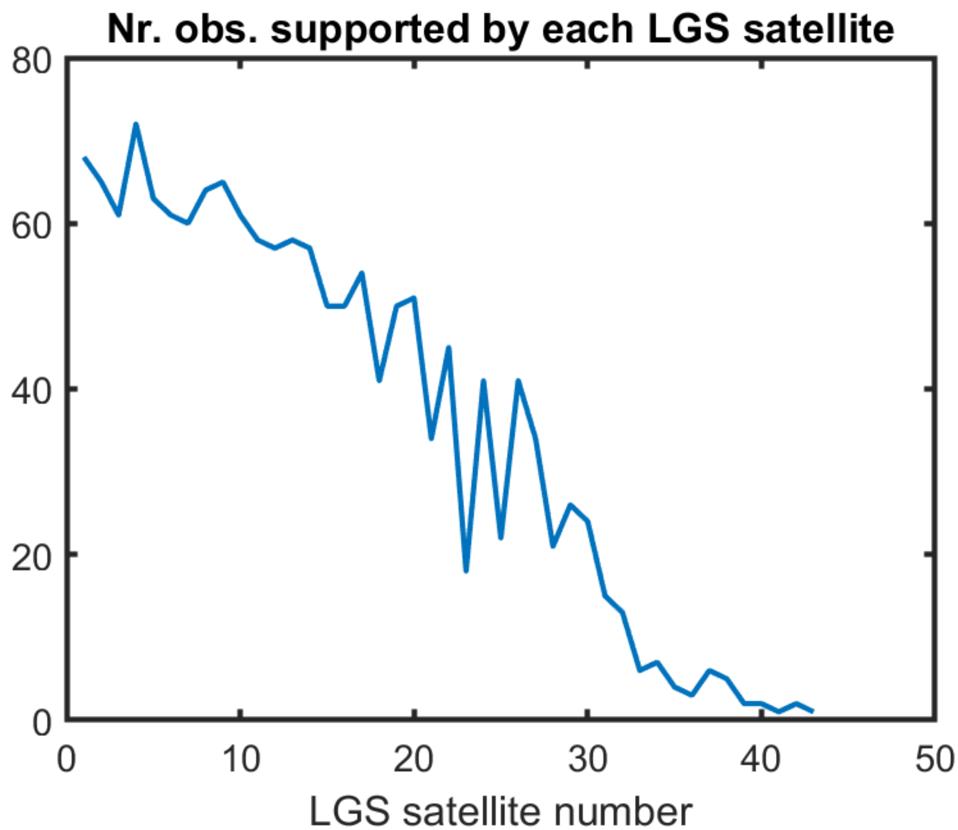


Figure 4-10: Number of observations supported by each LGS spacecraft in a hybrid schedule.

show, while there is a "long tail" of under-utilized LGS spacecraft, the peak utilization was slightly improved.

The next algorithm to be trialled will divide the sky first into segments, centered on a coarse Fibonacci grid, and then attempt to rotate the segmentation so that as many high-priority stars as possible end up in different segments from each other, so that they can all be the first star to be visited by their own LGS spacecraft.

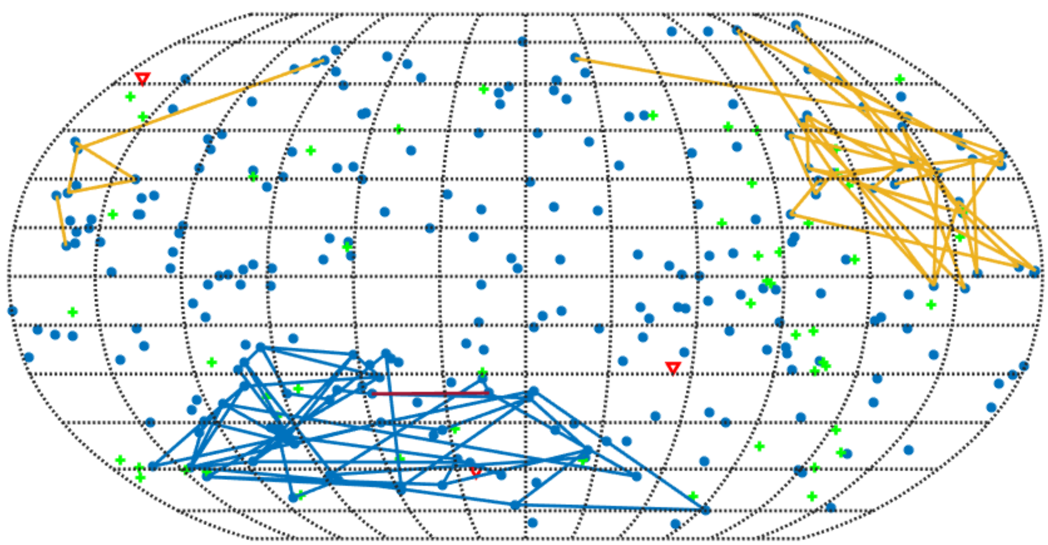


Figure 4-11: Segments traversed by a few LGS spacecraft in a hybrid schedule. The single maroon segment nestled amongst the blue track corresponds to a poorly-scheduled, under-utilized LGS spacecraft.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

LGS Pathfinder Missions

In this chapter, we consider options for a space LGS pathfinder mission. LUVOIR is not anticipated to launch before the 2030s [28], but it would be a beneficial risk-reduction exercise to demonstrate closing the AO loop using a laser on a satellite. There are already many telescopes on the ground with adaptive optics systems [18], and deploying lasers to Earth orbit is also technology readiness level (TRL) 9[1] [71], so it will be straightforward to conduct a test with these two technologies together to raise the TRL for the orbiting LGS concept as a whole.

5.1 Technology demonstration goals

In approximate order of difficulty, the needs of the L2 LGS architecture are as follows:

1. Demonstrating that the laser's light is correctly rejected ahead of the science detector (coronagraph, out-of-band filter, or in-band notch filter).
2. Demonstrating the wavelength stability, power stability and/or modulation, and lifetime of the laser in the space environment.
3. Closing the AO loop at a range of thousands or tens of thousands of kilometers.
4. Characterizing and controlling other potential LGS disturbances (glinting, thermal emission, thruster plume).

[1] Actual system "flight proven" through successful mission operations. [54]

5. Demonstrating WFSC on a space-based telescope.
6. Demonstrating telescope-LGS-target alignment and observation for up to a minute, without using LGS propulsion.
7. Demonstrating telescope-LGS-target alignment and observation for up to an hour, with or without LGS propulsion.
8. Demonstrating telescope-LGS-target alignment and observation for multiple hours, with LGS propulsion.

The first two items can be largely validated in laboratory testing, while the others will require the use of satellites, but most of the 'teething' challenges of technical integration only require for the laser and telescope to be separated by tens of thousands of kilometers, and so could be validated by a laser in geostationary orbit – even a laser communication satellite, not necessarily a dedicated LGS spacecraft – and a telescope on the ground. As we will show, this formation will only align with astronomical targets for a few seconds, but even that may be useful for some scientific results, especially relative to the modest expenditures required. When that demonstration has been successfully completed, a second demonstration mission could be performed by deploying a WFSC testbed telescope into geostationary orbit to validate the use of WFSC in space, as well as offer multi-minute astronomical observation windows. After both elements have been successfully demonstrated on orbit, then the hundreds of millions of dollars needed for a mission to higher orbits (with observations of multiple hours) can be justifiably spent.

5.2 Orbit selection

Greenaway and Clark's 1994 satellite LGS concept had the satellite deployed to a very highly elliptical orbit (HEO), so that it would linger for many hours at apogee and so that its motion would match that of the Earth's surface. [26] These orbits offer the greatest possible utility for telescope observations, but as we will see, they also impose serious costs to LGS development, deployment, and operation.

The lowest-energy orbit which matches sidereal motion at apogee is an elongated ellipse with an apogee altitude of 148,000 km, a perigee altitude of 400 km, and a period of 2.6 days. This orbit is illustrated to-scale with other orbits within Earth's sphere of influence (GTO, GEO, and the Moon) in Figure 5-1 as "Sidereal orbit 1". At apogee, the orbit's velocity is the same as the velocity of Earth's surface at the Equator, 465 m/s, while at perigee its velocity would be 10.6 km/s (compare to escape velocity from LEO, 10.8 km/s). This orbit would cross the altitudes of almost every satellite orbiting the Earth, from the International Space Station and Starlink, to the various GNSS constellations, to the GEO belt and graveyard, and would also pass through the van Allen belts twice in each orbit. The thermal and radiation design of a satellite in this orbit would be significantly more challenging than for a satellite which remained in LEO or in GEO, and there would also be operational challenges due to the number of satellites and constellations that are at risk of conjunction. This would require a more intense validation and testing campaign would be required to satisfy all stakeholders that the LGS spacecraft would be operational and not a hazard to navigation.

The fastest way to be deployed into this orbit would be to get a dedicated launch, but the launch cost of the Transiting Exoplanet Survey Satellite, which was deployed to a similar orbit, was $87 million. [3] To save that cost, the LGS could receive a rideshare to geostationary transfer orbit (GTO) and then raise its apogee from there. This would "only" cost 550 m/s of delta-V, but to have margin, the spacecraft should have twice that, or 1,100 m/s of delta-V capability. Even that figure is well within the capabilities of the LGS spacecraft design presented here (using the Busek BIT-3, see Section 2.4.6), but because the orbit transfer is essentially a large apogee-raising maneuver, the LGS spacecraft's electric propulsion system would not be very effective, because the thruster can only be active for a tiny fraction of the orbit and raise the apogee a little bit at a time.

To more precisely quantify this challenge, let us suppose that the spacecraft can activate its electric propulsion system on the section of its orbit around perigee which is the size of the diameter of the Earth. At 10.6 km/s, it will cross that distance

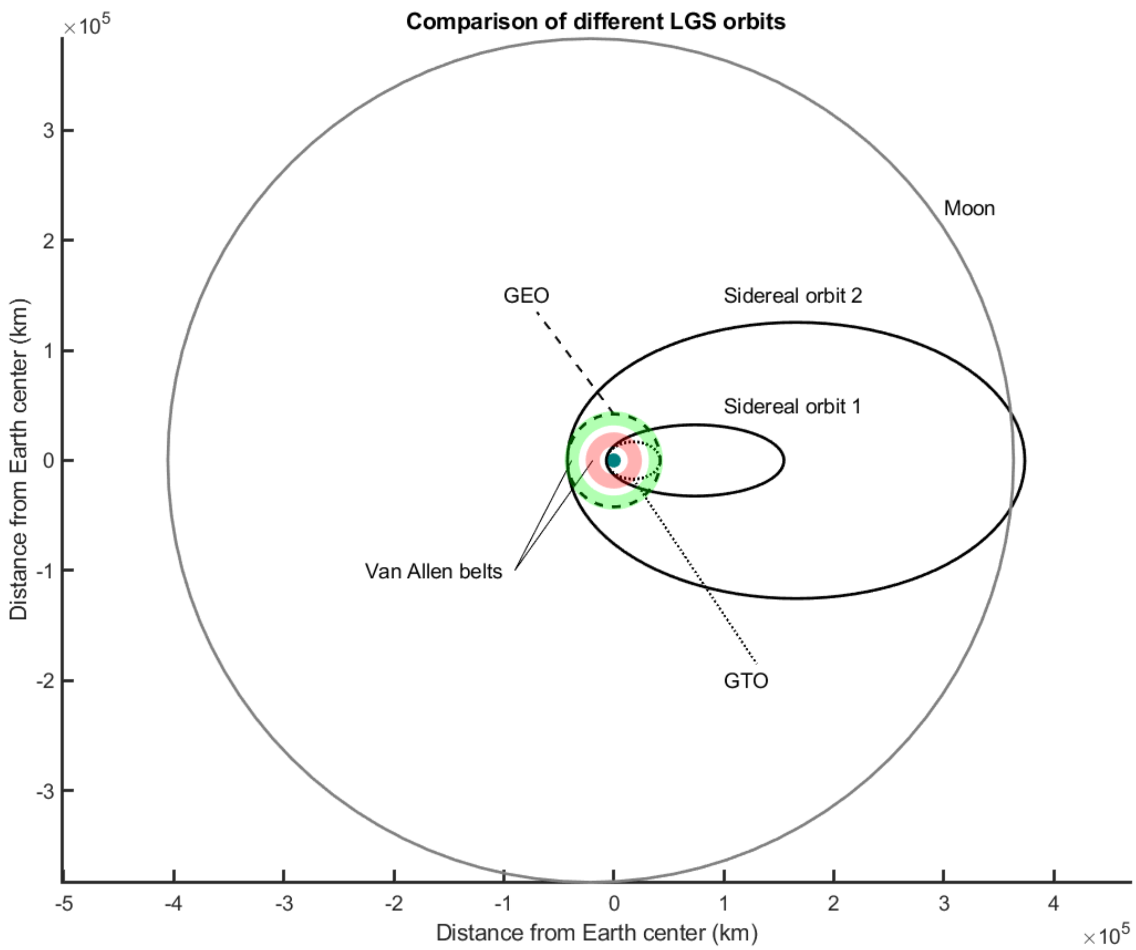


Figure 5-1: A to-scale comparison of the Earth (blue circle), GTO (dotted ellipse), GEO (dashed circle), the van Allen belts (shaded rings), the Moon's orbit (grey circle), and two "sidereal orbits" (black ellipses) whose apogee velocities are the same as the rotation speed of Earth's surface at the Equator.

in 20 minutes, but a maneuver of that duration that will only contribute 0.1 m/s of delta-V before it needs to wait for another orbit to thrust again. This means that over five thousand maneuvers (and over five thousand orbits, assuming that the thruster is always operational and never misses an opportunity) are required to completely raise the apogee to 148,000 km, but because the spacecraft must wait between 10 hours (GTO) and 2.6 days (as it approaches sidereal orbit 1) between maneuvers, that would mean a total mission duration of almost fifteen years[2] (!) before it was finally in position to be operational. During each of those orbits, the LGS spacecraft will pass twice through the van Allen radiation belts. As an optimistic estimate, we can use SPENVIS to calculate the total ionizing dose (TID) accumulated by the spacecraft after fifteen years in sidereal orbit 1 – this will be optimistic because, with the long orbital period, there will be fewer predicted passes through the van Allen belts (approximately 2,000 orbits total, rather than over 5,000) and those passes will be happening at higher speed and therefore lower duration per pass. Even under this optimistic projection, and with the typical protection of 1-mm-thick aluminum, the TID absorbed will stand at over 4 megarads, while the 5-year TID at L2 is less than 40 krad. Most commercial off-the-shelf (COTS) parts are tolerant of 5 krad, and some can tolerate up to 30 krad, but expensive radiation-hardened parts are required to tolerate dosages much higher than that. [55] If the shielding is increased to 7 mm of thickness (total mass: 6 kg out of the 24-kg mass budget), then the (optimistic) TID for the sidereal orbit 1 is reduced to 24 krad (and the 5-year TID at L2 is less than 5 krad). This is within the reach of COTS parts, but again, this is optimistic for the reasons mentioned above, and there is the additional difficulty that the spacecraft's thruster and laser must be uncovered and exposed to space to operate, and so the shield would require a mechanical articulation. Each of the five thousand (or more!) orbits would require four movements of the cover (two crossings of the Van Allen belts per orbit, closing the shield prior to each crossing and opening the shield after),

[2]To reduce this duration to, for example, six months, would require a thrust of 37 mN, or thirty times that provided by the Busek BIT-3. From Table 2.2, we can see that the only engine that could come close to supporting that is the Apollo Constellation Engine, which requires far more volume and power than the LGS design in Figure 2-6 can support, but could perhaps be supported with a 16U CubeSat.

so twenty thousand movements just to get to its operational orbit, and then however much longer the actuator will last for the actual mission.

If the LGS spacecraft had a chemical propulsion system, it could perform this maneuver with a single burn and reach its operational orbit immediately, but considering the VACCO Green Monopropellant System from Table 2.2, with its specific impulse of 170 seconds, the spacecraft would have to be 28% fuel by mass to achieve a delta-V capability of 550 m/s, with no margin for error. For a spacecraft of the same non-propulsion-system mass as the baseline LGS design (11.5 kg) plus the dry mass of the VACCO Green MiPS (3 kg), that means that over 5.6 kg of monopropellant are required, which would fit most densely into a sphere of 22 cm in diameter, not counting the tank walls, pressurant, pipes, valves, and other surrounding assemblies required. This barely fits within a 12U CubeSat structure on its own, and would crowd out the majority of the other subsystems of the LGS spacecraft. If we demand the same 100% margin, or 1100 m/s of delta-V capacity, the spacecraft would have to be nearly 50% fuel by mass, which would violate the 12U mass limit (29 kg total). A larger bus could be used, perhaps in the 180 kg ESPA-class form factor, but that would be even more expensive, and also substantially less attractive to the primary payload of the desired ride-share.

We could ease some of the operational challenges by getting a ride-share all the way to GEO, and then raising the apogee from there to an altitude of 367,000 km. The resulting orbit is drawn in Figure 5-1 as "Sidereal orbit 2". This orbit does not cross the GEO belt, or the inner van Allen belt, or any GNSS satellite orbits, but it does cross the Moon's orbit, and it costs 780 m/s to reach from GEO and has a period of nearly eleven days, so the deployment challenges of mission duration and the number of thruster activations are even greater. To reduce the deployment and operational difficulties, but still achieve the goals of validating the LGS concept, we propose that the LGS spacecraft should remain in or near GEO for the first pathfinder mission.

5.3 Pathfinder Mission "Zero": GEO Commsat

There are already satellites in GEO with laser communications payload, and so one of them could serve as a "zeroth pathfinder mission" by illuminating a telescope on the ground. Considering Keck, the MMT Observatory, and Gemini South as typical large telescopes with AO systems and a wide geographic dispersion, the declination that each telescope can see as a function of the GEO satellite's longitude is plotted in Figure 5-2. We can see in this figure that none of the telescopes have a line of sight to any of the GEO laser communication satellites currently in flight (which are all concentrated over Europe). This will change in the next few years, with NASA's Laser Communication Relay Demonstration payload launching on STP-3 in 2021 over the United States [21] and the next European Data Relay System (EDRS) satellite launching over the Asia/Pacific region before 2025 [32].

Such a mission is not guaranteed to have any scientific targets of interest at the other end of the telescope-LGS line of sight, but it would at least validate that an adaptive optics system can lock on to a laser guide star at a range of tens of thousands of kilometers.

5.4 Pathfinder Mission One: LGS in Inclined GEO

The first true pathfinder mission would feature a LGS spacecraft built to the design shown in Chapter 2 (or some descendant of it). It would initially be deployed to geostationary orbit as a ride-share, but to gain access to scientific targets, as well as validate the propulsion system, it would then transfer to an inclined geosynchronous orbit.

The delta-V cost of changing the inclination of velocity V_{orb} by an angle i using low-thrust propulsion is laid out in Equation 5.1 [40]:

$$\Delta V = \frac{\pi}{2} V_{orb} i \tag{5.1}$$

The inclination-change maneuver may be more expensive than raising an orbit

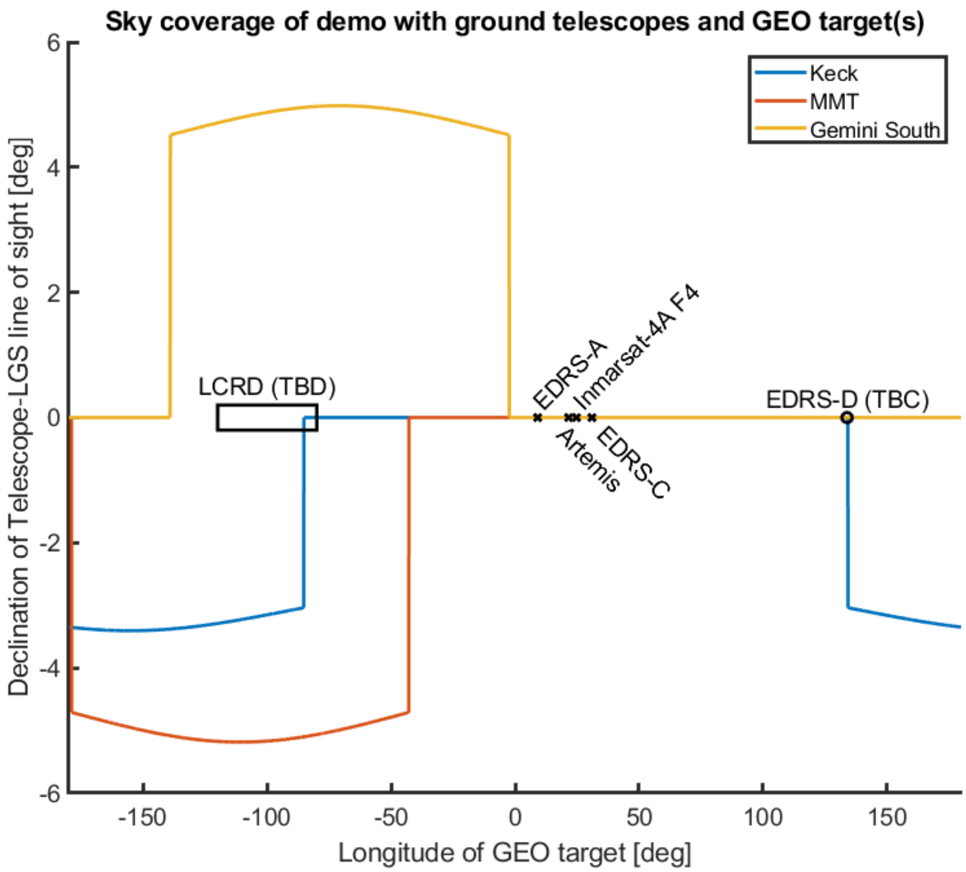


Figure 5-2: Declinations accessible by various ground-based telescopes looking through a laser guide star in GEO, with elevations greater than 10 degrees.

from GTO or GEO to the sidereal orbits, in terms of absolute delta-V costs (*e.g.* 1.26 km/s to achieve a 15 degree inclination change at GEO), but the thruster can remain active over the entire orbit, so the maneuver can be completed in fewer than six months. In this time, ground-based and/or space-based telescopes would observe the spacecraft to measure the impact of a spacecraft which is producing heat and firing a thruster on an astronomical observation, and test strategies for mitigating them. After the spacecraft has reached its mission orbit, its laser will be tested – both for the sake of validating the laser itself, and potentially for supporting scientific observations.

5.4.1 Observations from the ground through inclined GEO

The relative motion of the satellite LGS to the ground-based telescope is a slow north-and-south oscillation, and because of Earth's rotation, the telescope-LGS line of sight traces out a sinusoid curve over the sky every sidereal day. One such curve, the line of sight of Keck through a LGS spacecraft orbiting with inclination 15 degrees, RAAN 32.1 degrees, and true anomaly at the start of the sidereal day of 68.5 degrees, is plotted over the targets from Stark 2015 [57] in Figure 5-3. We can see that there are 4-8 stars that look readily accessible from this orbit. If we consider the set of all possible inclined geosynchronous orbits, the cost to observe any point in the sky (at the beginning of the sidereal day; the figure rotates around the sky depending on when exactly we want the satellite to be in a particular position) is shown in Figure 5-4. We can see that roughly a third of the sky falls within the 2000 m/s band, which is reachable by the LGS from deployment in GEO.

Those orbital parameters were initially chosen at random by the author (RAAN 30 degrees, true anomaly at epoch of 70 degrees) and then refined to achieve a very close pass to one of those targets.[3] The separation between the telescope-LGS line of sight and the telescope-target line of sight is plotted in Figure 5-5. When we compare

[3]The chosen target happened to be HIP 80824, or Wolf 1061, which is known to have at least three super-Earths in its planetary system, with two that roughly bracket its likely habitable zone. Another close pass in this orbit is 40 Eridani, notable in *Star Trek* as the home system of the Vulcans.

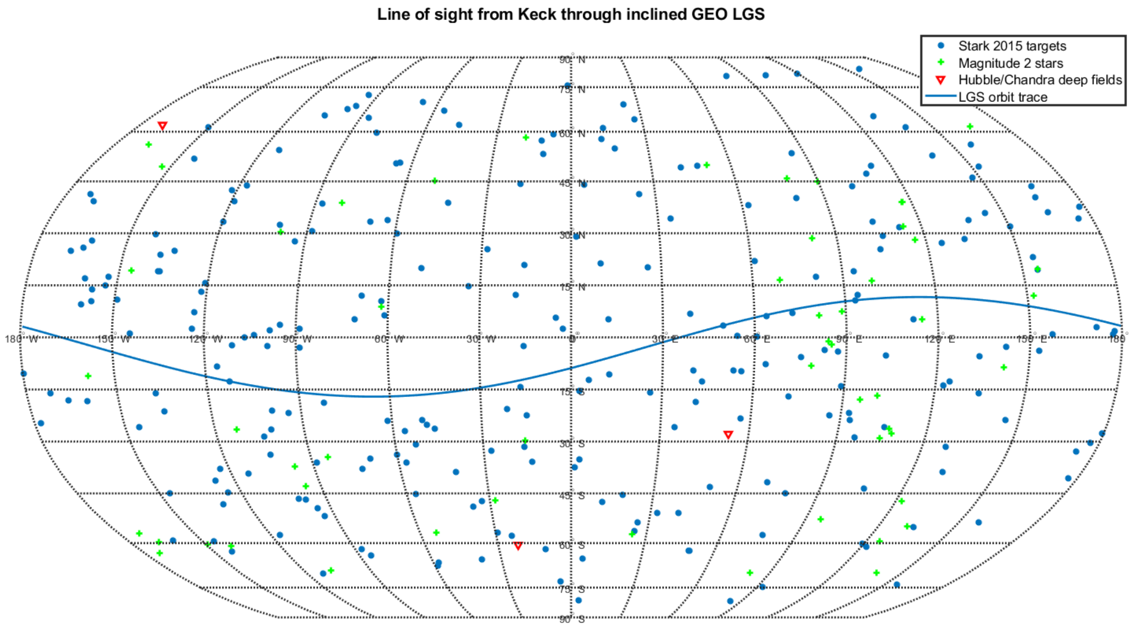


Figure 5-3: Line-of-sight from Keck through an orbiting LGS (inc. 15°, RAAN 32.1°, true anomaly at epoch 68.5°) plotted over the sky map of LUVOIR targets from Stark 2015 [57].

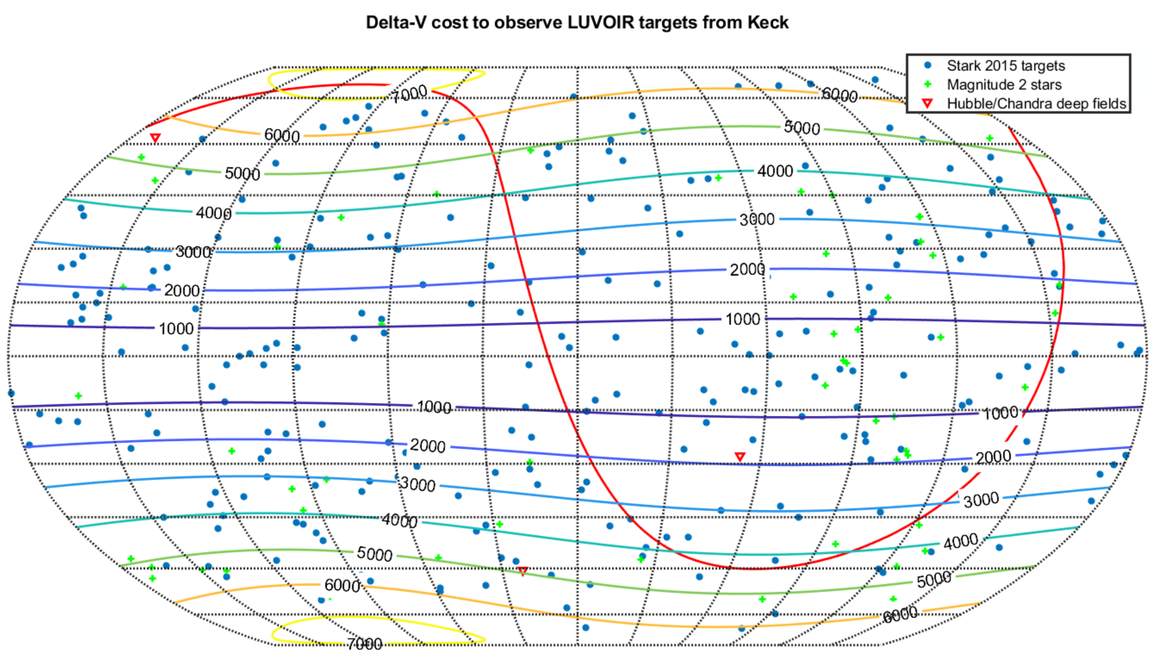


Figure 5-4: Sky map of LUVOIR targets from Stark 2015 [57], with delta-V cost (m/s) to incline a satellite in GEO to be on the line of sight in front of those stars (horizontal contours). Red curve: Keck horizon. All points North of this curve have elevation of at least ten degrees at this particular time (beginning of the sidereal day).

that distance to the distance limits of Keck's natural [37] and ground-based laser guide star systems [18], we see that the expected observation window is about eight seconds long, repeating every sidereal day. The LGS's thruster is not nearly powerful enough to meaningfully extend this time; the acceleration due to gravity is "only" 0.22 m/s^2 at GEO, but that would still require the spacecraft to carry a 3 N thruster (the equivalent of 2,500 Busek BIT-3 thrusters) to "hover" over the observatory.

The observation window *could* perhaps be improved with research into compensating for increased anisoplanatism with the higher photon counts achievable with the space-based LGS, or by having multiple LGS spacecraft with lasers at multiple wavelengths to get more information about the air column, but it is not going to support the multi-hour or even multi-minute observation that would be desirable for imaging exoplanets (the main criteria for Stark's targets). However, it may be useful for high-resolution, diffraction-limited imaging of bright targets, such as quasars, where the larger diameter of upcoming extremely large telescopes can be used to their full advantage. It will also prove that it is possible to coordinate and close the AO loop between a telescope and a laser guide star on a separate spacecraft, with relative separations and motions that are comparable in magnitude to that experienced at L2.

5.5 Pathfinder Mission Two: LGS in Super-GEO

To validate the operation of an AO system *in space* with an external LGS, the second LGS demo would be deployed to geostationary orbit, like the first, along with a WFSC testbed. That testbed could be something like a larger version of the Deformable Mirror Demonstration CubeSat ("DeMi"). [20] The LGS spacecraft would then use its propulsion system to raise its orbit to a higher altitude, into an orbit which has a 7/6 resonance with GEO[4] and whose velocity at periapsis matches that of GEO. The two orbits are shown in Figure 5-6. The yellow vector drawn on the image

[4]That is, the LGS satellite will orbit six times in the same time that the WFSC testbed, still in GEO, orbits seven times

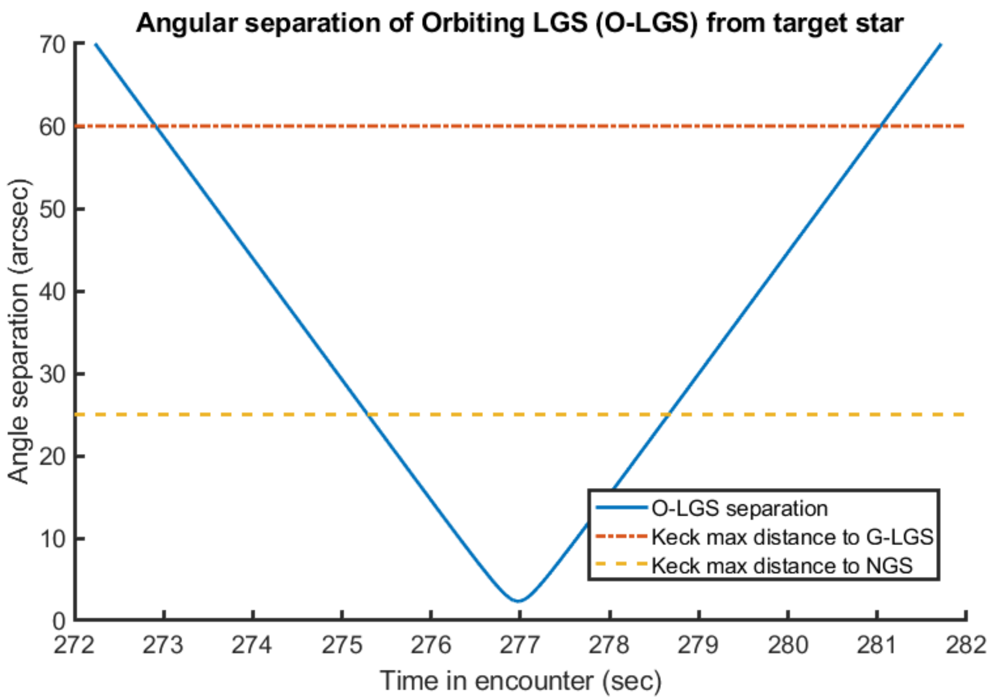


Figure 5-5: The separation between the telescope-LGS line of sight and the telescope-target line of sight during an observation pass, compared to distance limits of other Keck AO systems. The telescope would probably conduct this observation by locking on to the LGS spacecraft much earlier than this window, and then wait for the target star to pass through.

represents a twelve-minute interval in which the line-of-sight from the telescope to the LGS remained within a one-arcminute-radius circle on the sky, within the limit of Keck's ground-based laser guide star system. [18] This offers more time to properly lock on to a target star and validate AO system performance. Just over 150 m/s of delta-V is required to execute this maneuver, well within the capability of the LGS spacecraft, and the separation between the two satellites during the observation is approximately 7,000 km – not as much as will be necessary for LUVOIR, but as long as the testbed's telescope is less than 3.7 meters across, it will still meet the flat-wavefront requirement to work with the ZWFS. This particular orbit is not inclined, and so this precise configuration will only be able to observe stars near the equator, but that still includes some rather interesting targets, such as Ross 128 – a star only 11 light-years away from Earth, with at least one planet [11], and which attracted the attention of the SETI community when Arecibo detected radio signals from its vicinity, although these were later suspected to have actually come from a GEO communication satellite. [53]

5.6 Pathfinder Mission Three: LGS in HEO

After the LGS spacecraft's systems have been fully validated, either with a ground-based telescope or a space-based WFSC testbed, then funding can be pursued to reach the more ambitious orbits necessary to support longer observations. These orbits have not been investigated as thoroughly in this work, but as a preliminary investigation of the possibilities, we can consider an orbit similar to sidereal orbit 2 (the large ellipse in Figure 5-1) but with a periapsis at half the altitude to GEO.[5] Other orbital elements were chosen more-or-less at random by the author and simulated, and the resulting orbit is shown in Figure 5-7. The yellow vector drawn on the image represents a 108-minute interval in which the line-of-sight from the telescope to the LGS remained within a one-arcminute-radius circle on the sky, within the limit of Keck's ground-

[5]The sidereal orbits were designed to match Earth's motion *at the Equator*, and so to work with Keck or other observatories not at the Equator, the orbit should be slightly slower at apoapsis.

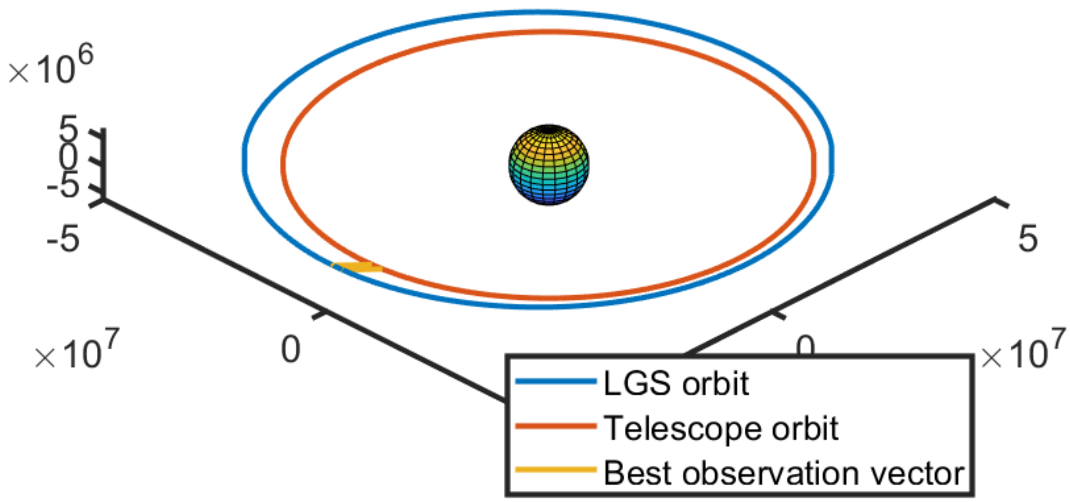


Figure 5-6: Pathfinder mission for demonstrating LGS with a space telescope.

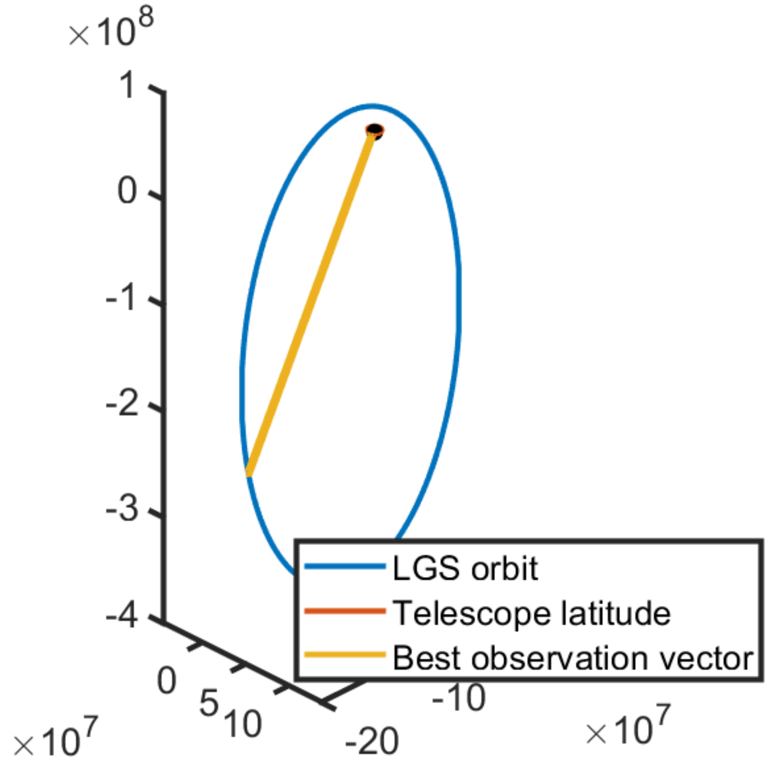


Figure 5-7: One potential orbit for long-duration ground-based observations supported by Earth-orbiting laser guide stars.

based laser guide star system. [18] Much work remains to be done on optimizing the selection of an orbit given a telescope and target, but similar observations (of slightly lesser quality) occur roughly once a night for nine days out of the ten-day orbital period at different points on the sky, as shown in Figure 5-8.

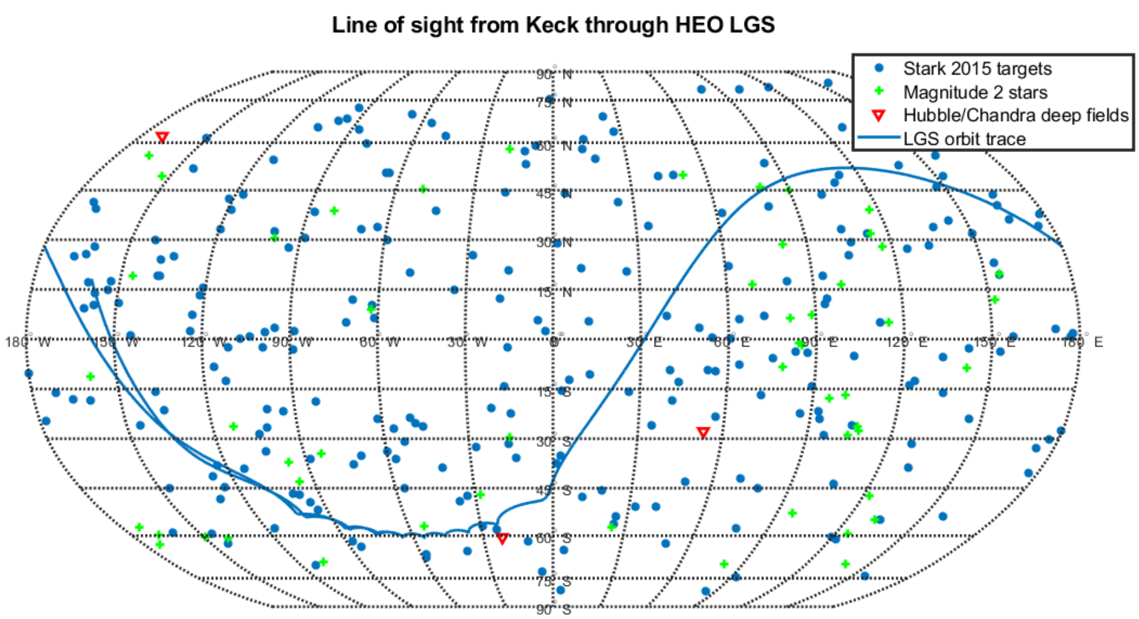


Figure 5-8: Trace of Keck-LGS line of sight, with the LGS in a highly elliptical orbit. Each ‘tooth’ in the southern part of the orbit track represents an observation opportunity.

Chapter 6

Conclusions

6.1 Contributions

In this work, we have developed tools for exploring the design space of laser guide stars, and for preliminary scheduling of observations to be supported. We have used these tools to develop a baseline laser guide star design that uses commercial off-the-shelf parts, and a design reference mission that services the number and pace of observations currently intended for the LUVOIR mission.

Even though the LGS concept presented here is intended to service a space telescope that will not be built for at least a decade, it does not use any parts or technologies which do not exist today; for that reason, we have also developed a path towards risk-reduction and eventual implementation of the laser guide star concept.

The first step is to deploy a LGS spacecraft to geostationary orbit, where it will incline its orbit to validate the propulsion system and align itself with targets of observation. It will shine its laser towards a ground-based or space-based telescope to validate the use of an external space-based laser guide star to inform an adaptive optics system. Because the telescope-LGS line of sight will only be in alignment with an astronomical target for a few seconds, this mission's scientific utility will come from supporting diffraction-limited observations of bright targets by large telescopes. It may also be useful for space situational awareness missions.

After the LGS spacecraft has been validated at GEO, support can be obtained to deploy one or more LGS spacecraft into highly-elliptical orbits, such that the spacecraft is nearly static against the background stars at apogee, as originally envisioned by Greenaway and Clark 1994 [26]. This will support observations with longer integration times, such as exoplanet imaging, deep-field observations, and other dimmer targets.

If desired, LGS spacecraft could then be deployed to L2 halo orbits before LUVOIR or another 'host' mission is launched, to validate the spacecraft's operation in the L2 environment or to service other telescopes already present.

6.2 Future Work

The tools and work presented here show that the LGS spacecraft has the capabilities required to service LUVOIR or other large space telescopes, but work remains to be done in developing detailed simulations of different mission phases.

Work has begun on simulating the control loop between the telescope and laser guide star during an observation. The model accommodates uncertainty on the part of the LGS spacecraft's thrusters, but as shown in Section 2.4.1, the more pressing constraint will actually be in the ability of the telescope to track the LGS spacecraft's relative position and velocity.

The mission scheduling can be further improved with more sophisticated scheduling; the current approaches use crude optimizations, but these can be improved by adapting them to use more appropriate solver tools. The scheduler should incorporate some notion of scientific priority (perhaps based on a small number of categories, if not a strict ranking) and different revisit rates based on habitable zone size, as well as clustering observations to save fuel.

The ultimate goal is to produce a day-in-the-life simulation where a LGS spacecraft is deployed to L2, services an observation, then transits, and then services a second observation. This will not only evaluate the propulsion and control systems, but also the power and thermal needs of the mission.

Eventually, the LGS concept may be evaluated for its utility in multi-telescope systems, such as a reference for interferometers or time distribution for radio astronomy.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A

Alternate LGS architectures

This appendix briefly discusses alternate architectures to the L2 LGS concept elaborated in the main body of this work.

A.1 Different LGS form factors

Using technologies available today, all required components of the LGS spacecraft can be fit within a 12U CubeSat form factor. However, during the course of the design process, the 16U form factor was also considered as a possibility in case greater battery volume or solar panel area were required. The delta-V capabilities of the propulsion system options (from Table 2.2) in the 16U form factor are presented in Table A.1.

In the decades between now and the launch of LUVOIR, it is possible that the technologies might be miniaturized such that all required components can fit within a 3U bus. For those propulsion systems which can fit within that form factor (those with a 1U cross-section, which would normally be doubled-up in the 12U or 16U LGS spacecraft), the same delta-V calculation was also performed for the 3U case.

If the calculations laid out in Chapter 4 are followed for the 16U case, then 24 LGS vehicles are required to support the LUVOIR baseline mission (1,539 observations of 259 stars in 5 years), rather than 19.

Prop system	**dV** (16U, m/s)	**dV** (3U, m/s)
Accion TILE 5000 x1 [1]	198	1227
Accion TILE 5000 x2 [1]	398	-
Apollo Constellation [5]	626	-
Busek BIT-3 [13]	1456	-
Enpulsion IFM Nano x1 [22]	283	1742
IFM Nano (max I_{sp}) x1 [22]	567	3484
Enpulsion IFM Nano x2 [22]	569	-
IFM Nano (max I_{sp}) x2 [22]	1139	-
Phase Four x1 [46]	186	1179
Phase Four x2 [46]	376	-
VACCO Green MiPS [67]	145	-
VACCO/JPL MarCO [66]	32	-

Table A.1: Table of propulsion system options considered for LGS spacecraft, as in Table 2.2, but with delta-V capacity calculated with total spacecraft mass of 24 kg (16U maximum mass) or 4 kg (3U maximum mass), as applicable. Compare to the delta-V capacity of the "stock" LGS spacecraft (12U with the Busek BIT-3), 2480 m/s.

A.2 Single laser to distributed reflectors

The LGS spacecraft could be simplified if, instead of all carrying their own lasers, they would simply reflect a laser transmitted by an external facility – perhaps LUVOIR itself, or from a large ground-based laser facility such as those proposed for interstellar communication or interstellar "chipsat" probe propulsion. [16, 36] Unfortunately, adding this reflection would mean that the power returned to the telescope is proportional to the inverse *fourth power* of the LGS-telescope range, rather than the inverse square.

As an optimistic estimate, let us suppose that LUVOIR can carry a laser ten times larger and more powerful than those carried on the LGS spacecraft (so 50 W coming through a 35 cm aperture). Let us further suppose that the LGS spacecraft has a full 20x30 cm face covered in a 100% reflective, perfectly-right-angled retroreflector. Therefore, the only angular divergence in the system will be Gaussian beam diffraction from the laser and the retroreflector. Even under these ideal conditions, the LGS spacecraft would only intercept 36 μW of power, and its return reflection to LUVOIR would only be magnitude 3, which would not be enough to support the WFSC needs identified in Section 2.3. If the laser were instead transmitted through LUVOIR's main telescope, then the return reflection would be magnitude -4, which would be bright enough (although still not quite as bright as the baseline LGS), but LUVOIR would then need to have additional optics for injecting a laser beam into its optical path without any leakage back to the science instruments, and would also have to handle additional phenomena like indirect scatter from the secondary mirror support structure.

LUVOIR could be spared those indignities if the laser were instead generated on Earth and beamed up to L2. This would require an even larger, more powerful laser, but conceivably, a megawatt-class laser, such as the Airborne Laser [64], could be installed in an extremely large telescope, such as the Thirty Meter Telescope.[1] Assuming no atmospheric distortion (!), the laser return from Earth, to an LGS, to LUVOIR could be as bright as magnitude -9, but with typical atmospheric turbulence (r_0 = 10 cm) that return drops to magnitude 3. Perhaps the ground-based laser facility could use adaptive optics aided by a dedicated Earth-pointing laser guide star at L2 (!!) to correct for that distortion.

[1] Both projects were/are multi-billion-dollar programs in their own right, so it's doubtful that any total savings would be realized. Constructing such a facility would also require both military and scientific agencies to collaborate in ways that don't serve either of their goals very efficiently. Finally, if it *were* built, it would be detectable by extraterrestrial intelligence, [16] so it would face demands that would reduce its utilization for "real astronomy". Still, in the words of Giorgio Tsoukalos, "Is such a thing even possible? Yes, it is!" [29]

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix B

LGS Design Code

To run this code, download it all into a single folder, along with the star and observation .csv files from Appendix C. Then open and run `LGSmain.m` with MATLAB (this code makes use of the Mapping Toolbox, the Optimization Toolbox, and the Phased Array System Toolbox). If run as-is, this will generate all plots used in this dissertation, although some were resized or otherwise edited for aesthetic purposes before being published.

Running MATLAB R2020a with a six-core Ryzen 5 2600 processor, this code runs in approximately 200 seconds on its first run, largely dominated by the traveling-salesman-problem solver; the TSP result is saved for later use, and subsequent runs will complete end-to-end in 80 seconds. A four-core Intel Core i7-4700MQ processor ran the code in 240 seconds for the first run, and 85 the next; this shows that MATLAB's solver code is well-parallelized and takes advantage of additional cores, but the author's code is not – several of the scripts loop sequentially over variables, which could be parallelized for a reduction in computing time.

Further revisions to this code may leverage Joseph Kirk's Traveling TSP Genetic Algorithm code [30], which seems to respond better to some cases which the author has preliminarily investigated.

All files are also available at the author's GitHub repository: `https://github.com/jimclark/LaserGuideStar`

B.1 LGSmain

The main program for setting up the LGS parameters (mass, LUVOIR size, propulsion system options, etc.) and invoking the other functions.

```
%% LGS Main Function

%% Setting up variables

AU = 1.496e11;
Re = 6371000;
g0 = 9.8066;
daysec = 60*60*24;
yrsec = daysec*365.25;
solarConst = 1366;
c = 299792458;
h = 6.626e-34;
ly = c*yrsec;
parsec = AU/(deg2rad(1/3600));

% Updated mission based on Chris Stark's figures.
num_stars = 259;
total_obs = 1539;
total_mission_time = 5*yrsec; % 5 years

% Telescope parameters
scope_d = 9.2; % meters
scope_seg_d = 1.15;
obs_lam = 500e-9; % visible observation

% Laser parameters
lambda = 980e-9; % staying out of the visible band
pwr_laser = 5;
D_laser = 0.035; % 3.5 cm aperture

% LGS parameters

sc_mass = 24; % Maximum mass of 12U CubeSat
sc_mass_opt = 11.5; % Mass of selected components, without propulsion system

% Propulsion system options
%prop_names = {'Accion TILE 5000 x2','Apollo Constellation Engine','Busek BIT-3','
    Enpulsion IFM Nano (typ. setting) x2','Enpulsion IFM Nano (max I_{sp}) x2','Phase
     Four x2','VACCO Green Monopropellant','VACCO Cold Gas'};
prop_names = {'Accion 5000','Apollo CE','Busek BIT-3','Enp. Nano (nom.)','Enp. Nano (
```

```
    max I_{sp})','Phase Four','VACCO Grn','VACCO Cold Gas'};
sc_fuel = [0.64, 1.0, 1.5, 0.46, 0.46, 1, 2, 1.03];
sc_prop_dry = [2.2, 4.5, 1.4, 1.44, 1.44, 3, 3, 2.46];
sc_isp = [1500, 1500, 2300, 3000, 6000, 900, 170, 75]; % Can flex Enpulsion up to
    6000 sec by reducing thrust.
sc_max_thrust = 1e-3*[3, 33, 1.24, 0.7, 0.5, 2, 0.4, 0.1];

%% Derived parameters

range_LGS = scope_d^2/(2*lambda); % quarter-wave curvature across the telescope
    mirror -- 43,000 km

iwa_box_rad = range_LGS*(0.25*obs_lam/scope_d); % Radius of "target box" trying to
    stay waaaay inside coronagraph, 0.25 lambda/D
iwa_box_rad_relax = range_LGS*(obs_lam/scope_d); % Radius of "target box" trying to
    stay kinda inside coronagraph, 1 lambda/D
seg_box_rad = range_LGS*(lambda/scope_seg_d); % Radius of "target box" trying to keep
     wavefronts flat (i.e. no more than one wavelength error) on each mirror segment

div_laser = 2*(lambda/(pi*(D_laser/6))); % 107 urad (980 nm), full-width Gaussian
    divergence
% Gaussian beam waist is 1/3rd the actual diameter of the main optic
% Todo: break out gaussian beam divergence into its own calculation

[Prx,Photrx,appMag,bw] = linkbudgetG(pwr_laser,D_laser,range_LGS,lambda,scope_d);
[Prx2,Photrx2,appMag2,bw2] = linkbudgetG(pwr_laser,D_laser,scope_d^2/(2*532e-9),532e
    -9,scope_d);

idxs = [1 3]; % Only looking at the first and third "stars" in the Fibonacci spiral
phi = acos(1-2.*(idxs-0.5)./num_stars);
theta = pi*(1+sqrt(5))*(idxs-0.5);
x = cos(theta).*sin(phi);
y = sin(theta).*sin(phi);
z = cos(phi);

sep_std = sqrt((x(1)-x(2)).^2+(y(1)-y(2)).^2+(z(1)-z(2)).^2); % standard unit-sphere
    separation between adjacent stars (scale by range)

%% Preliminary propulsion-system selection

single_maneuver_time = 2.*sqrt(range_LGS*sep_std*sc_mass./sc_max_thrust);
smt_opt = 2.*sqrt(range_LGS*sep_std*(sc_mass_opt+sc_prop_dry+sc_fuel)./sc_max_thrust)
    ;
smt_opt_days = smt_opt/(60*60*24);
smt_opt_dv = 2.*sqrt(range_LGS*sep_std*sc_max_thrust./(sc_mass_opt+sc_prop_dry+
```

```
    sc_fuel));
single_maneuver_days = single_maneuver_time./(60*60*24);
single_maneuver_dv = 2.*sqrt(range_LGS*sep_std.*sc_max_thrust./sc_mass);
dv_caps = g0.*sc_isp.*log(sc_mass./(sc_mass-sc_fuel));
dv_caps_opt = g0.*sc_isp.*log((sc_mass_opt+sc_prop_dry+sc_fuel)./(sc_mass_opt+
    sc_prop_dry));
number_maneuvers = floor(dv_caps./single_maneuver_dv);
num_man_opt = dv_caps_opt./smt_opt_dv;
slowest_ep_smt_opt_days = max(smt_opt_days.*(sc_isp>1000));
num_man_opt_et = num_man_opt.*(slowest_ep_smt_opt_days./smt_opt_days).*(smt_opt_days
    <=slowest_ep_smt_opt_days); % Scale by the maneuver time of the lowest-thrust
    electric prop system (i.e. the longest maneuver time with a prop system isp >
    1000 sec).

[max_mans_opt,idx_max_mans_opt] = max(num_man_opt_et);

fprintf("Selected propulsion system: %s\n",prop_names{idx_max_mans_opt});

sc_prop_dry_nom = sc_prop_dry(idx_max_mans_opt);
sc_fuel_nom = sc_fuel(idx_max_mans_opt);
sc_isp_nom = sc_isp(idx_max_mans_opt);
sc_max_thrust_nom = sc_max_thrust(idx_max_mans_opt);

sc_mass_opt_tot = sc_mass_opt+sc_prop_dry_nom+sc_fuel_nom; % total opt mass
dvcap = g0*sc_isp_nom*log(sc_mass_opt_tot/(sc_mass_opt_tot-sc_fuel_nom));

%% Run other scripts

OrbitCalcs2dome3 % to get the maximum background acceleration
OrbitCalcs2dome2
OrbitCalcs2dome
NoiseCalcsPropSens
NoiseCalcs
StarkSkymap

%%
DRM_prop_options
DRM_sensitivity

%%

if isfile('stark_skymap_tsp.mat')
    load('stark_skymap_tsp.mat')
else
    StarkSkymap_TSP % This can take some time.
```

```
end

if isfile('stark_skymap_tsp_ham.mat')
    load('stark_skymap_tsp_ham.mat')
else
    ham_StarkSkymap
end

%%

StarkSchedule
StarkScheduleAltB
StarkScheduleAltD

%%
SkyCalcs
SkyCalcsOffGEO
SkyCalcsOffGEO2

%%
HEO_LGS
SkyCalcsHEO
PowerCalcs
OrbitCalcs3CL3
LGSretro
```

B.2 linkbudgetG

A helper function for calculating Gaussian beam link budgets (power received, photons received per second, apparent magnitude, and beamwidth).

```
%% Gaussian beam link budget
% Assumption: Gaussian beam waist is 1/3 of the Tx aperture radius (Dtx/6)
% Provide all inputs in SI units (W, m, m, m, m).
% BWtx is FWHM beamwidth
% Includes 3 dB pointing loss (i.e. +/- BWtx/3.4)
function [Prx,Photrx,appMag,BWtx] = linkbudgetG(Ptx,Dtx,range,lambda,Drx)

c = 299792458;
h = 6.626e-34;

w0 = Dtx/6;
zR = pi.*(w0.^2)./lambda;

wZ = w0.*sqrt(1+(range./zR).^2);

BWtx = (2*atan(wZ./range))/1.7; % will converge to (2*lambda/(pi*w0))/1.7
fluxrec = 2*Ptx./(pi.*wZ.^2);

Arx = 0.25.*pi.*Drx.^2;
Prx = (10^-0.3).*Arx.*fluxrec; % 3 dB pointing loss accounted here.

Ephot = h.*c./lambda;
Photrx = Prx./Ephot;

Fx0 = zeros(size(lambda));
BWwideband = zeros(size(lambda));

for i = 1:numel(lambda)

    if (lambda(i) < 398e-9 && lambda(i) > 332e-9)     % U
        Fx0(i) = 1810;
        BWwideband(i) = 66/365;
    elseif (lambda(i) < 492e-9)                       % B
        Fx0(i) = 4260;
        BWwideband(i) = 94/445;
    elseif (lambda(i) < 595e-9)                        % V
        Fx0(i) = 3640;
        BWwideband(i) = 88/551;
    elseif (lambda(i) < 727e-9)                       % R
        Fx0(i) = 3080;
```

```
        BWwideband(i) = 138/658;
    elseif (lambda(i) < 880.5e-9)                   % I
        Fx0(i) = 2550;
        BWwideband(i) = 149/806;
    elseif (lambda(i) < 1080e-9)                    % Y, e.g. Starshot
        Fx0(i) = 2075; % ESTIMATE!!!!
        BWwideband(i) = 120/1020;
    elseif (lambda(i) < 1326.5e-9)                  % J, e.g. ABL
        Fx0(i) = 1600;
        BWwideband(i) = 213/1220;
    elseif (lambda(i) < 1783.5e-9)                  % H, e.g. NODE/FLARE
        Fx0(i) = 1080;
        BWwideband(i) = 307/1630;
    elseif (lambda(i) < 2385e-9)                    % K
        Fx0(i) = 670;
        BWwideband(i) = 390/2190;
    end

end

FJy = Photrx.*h.*1e26./(BWwideband.*Arx);

appMag = -2.5.*log10(FJy./Fx0);

end
```

B.3 L2 orbit calculations

B.3.1 OrbitCalcs2dome3

Calculates the thrust required to hold the Telescope-LGS formation to observe any direction over the course of a full six-month halo orbit period.

```
%% Run a full cube of az, el, and time, average over time, divide dV cap by it,
    divide by 2.

AU = 1.496e11;
Re = 6371000;
Tnd = 365.25*24*60*60/(2*pi);
muSE = 3.036e-6;
aMoon = 384400e3;
aGeo = 42164000;
dt = (3600/Tnd);  % 1 hour steps.

%%


yScopeInit = [1.00717285919175; 0; 0; 0; 0.0163636; 0];



tspan = 0:(3600/Tnd):3.2; % ~6 mos, time for scope to stay on-target
%tspan = 0:0.001:1; %~2 mos, time for LGS to stay close without active station-
    keeping
%tspan = linspace(0,24*60*60/Tnd,10000); % 1 day

[tScope,yScopeMat] = ode45(@cr3bpse,tspan,yScopeInit);

xpScope = yScopeMat(:,1);
ypScope = yScopeMat(:,2);
zpScope = yScopeMat(:,3);
xvScope = yScopeMat(:,4);
yvScope = yScopeMat(:,5);
zvScope = yScopeMat(:,6);



xpiScope = xpScope.*cos(tScope) - ypScope.*sin(tScope); % inertial reference frame
ypiScope = ypScope.*cos(tScope) + xpScope.*sin(tScope);
xviScope = xvScope.*cos(tScope) - yvScope.*sin(tScope);
yviScope = yvScope.*cos(tScope) + xvScope.*sin(tScope);
```

```
% idxStart = 1; % Start at beginning.
% idxStart = 300; % Trying to get to max turn...
% idxStart = find(ypScope==max(ypScope)); % First corner.
% idxStart = find(xpScope==max(xpScope)); % D loop.




% disp(tStart*365.25/(2*pi))

azvec = 0:0.5:360;
elvec = 0:0.5:90;
tidxvec = 1:24:numel(tScope);  % Index of time points, skipping 1 day at a time.

[elevs,azimuths,tidx] = ndgrid(elvec,azvec,tidxvec);

[elevs2d, azimuths2d] = ndgrid(elvec,azvec);



avgaccs = zeros(size(azimuths2d));

accs = zeros(size(azimuths));
thrusts = zeros(size(azimuths));
times = zeros(size(azimuths));

desrange = range_LGS/AU;

%%

total = numel(azimuths);

for i = 1:total


    if mod(i,10000) == 0
        clc
        fprintf('%.1f%%\n',100*i/total)
    end


    idxStart = tidx(i);
    tStart = tScope(idxStart);
    times(i) = tStart;
```

```
goalAzI = azimuths(i);
goalAzRi = goalAzI-rad2deg(tStart);
goalEl = elevs(i);

goalXinit = xpScope(idxStart)+desrange*cosd(goalEl)*cosd(goalAzRi);
goalYinit = ypScope(idxStart)+desrange*cosd(goalEl)*sind(goalAzRi);
goalZinit = zpScope(idxStart)+desrange*sind(goalEl);

velXinit = xvScope(idxStart) + goalYinit - ypScope(idxStart);
velYinit = yvScope(idxStart) - goalXinit + xpScope(idxStart);

goalXi = goalXinit*cos(tStart) - goalYinit*sin(tStart); % inertial reference
frame
goalYi = goalYinit*cos(tStart) + goalXinit*sin(tStart);

radInit = sqrt(goalXi^2 + goalYi^2 + goalZinit^2);

yScope = [xpScope(idxStart); ypScope(idxStart); zpScope(idxStart); ...
    xvScope(idxStart); yvScope(idxStart); zvScope(idxStart)]; %

yLGS = [goalXinit; goalYinit; goalZinit; ...
    velXinit; velYinit; zvScope(idxStart)]; %

dydtLGS = cr3bpse(tStart,yLGS);
dydtScope = cr3bpse(tStart,yScope);

xvScoper = dydtScope(1);
yvScoper = dydtScope(2);
zvScoper = dydtScope(3);
xaScoper = dydtScope(4);
yaScoper = dydtScope(5);
zaScoper = dydtScope(6);

xaScopeic = xaScoper - xpScope(idxStart) - 2*yvScoper; % Coaligned inertial
reference frame
yaScopeic = yaScoper - ypScope(idxStart) + 2*xvScoper;

xaScopei = xaScopeic*cos(tStart) - yaScopeic*sin(tStart); % rotate
yaScopei = yaScopeic*cos(tStart) + xaScopeic*sin(tStart);

xvLGSr = dydtLGS(1);
yvLGSr = dydtLGS(2);
zvLGSr = dydtLGS(3);
xaLGSr = dydtLGS(4);
```

```
    yaLGSr = dydtLGS(5);
    zaLGSr = dydtLGS(6);

    xaLGSic = xaLGSr - goalXinit - 2*yvLGSr; % Coaligned inertial reference frame
    yaLGSic = yaLGSr - goalYinit + 2*xvLGSr;

    xaLGSi = xaLGSic*cos(tStart) - yaLGSic*sin(tStart); % rotate
    yaLGSi = yaLGSic*cos(tStart) + xaLGSic*sin(tStart);

    dposi = [desrange*cosd(goalEl)*cosd(goalAzI) ; desrange*cosd(goalEl)*sind(goalAzI
    ) ; desrange*sind(goalEl)];
    dacci = [xaLGSi-xaScopei ; yaLGSi-yaScopei ; zaLGSr-zaScoper ];

    dposin = norm(dposi);
    daccin = norm(dacci);

    dotp = dot(dposi,dacci);

    angle = acosd(dotp/(dposin*daccin));

    accreq = daccin*sind(angle);

    accreqSI = accreq*(AU/Tnd^2);
    TreqSI = accreqSI*sc_mass_opt_tot;

    accs(i) = accreqSI;
    thrusts(i) = TreqSI;

end


%%
for i = 1:numel(elvec)
    for j = 1:numel(azvec)
        avgaccs(i,j) = mean(accs(i,j,:));
    end
end

max_bg_acc = max(max(max(accs)));
min_bg_acc = min(min(min(accs)));
max_bg_thrust = max_bg_acc*sc_mass_opt_tot;
min_bg_thrust = min_bg_acc*sc_mass_opt_tot;

avg_acc = mean(mean(avgaccs));
```

```
% Technically this should go to a later script, after prop selection

timeObs = 0.5*(dvcap./avgaccs);

%%

% Contours = [100 150 200 300];
Contours = [300 600 1200 2400 4800];
figureMap = figure;
surf(azimuths2d,elevs2d, log(timeObs/(24*60*60)),'EdgeColor','none');
colorbar('YTick',log(Contours),'YTickLabel',Contours);
colormap(jet);
caxis(log([Contours(1) Contours(length(Contours))]));
colorbar('FontSize',12,'YTick',log(Contours),'YTickLabel',Contours);
title(sprintf('Max. obs. time (days, 12U RF Ion, %.2g,000 km range)',range_LGS/1e6))
xlabel('Ecliptic longitude (deg)')
ylabel('Ecliptic latitude (deg)')
view([0 90]);
xlim([0 360]);
ylim([0 90]);

% set(findall(gcf,'-property','FontSize'),'FontSize',14);

% save('cost-of-observation-10k.mat','accs','avgaccs','elevs','azimuths','tidx')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figureMap,sprintf('Max-watching-time-%.2gk.png',range_LGS/1e6));

%%

xshad = [1-muSE, 1.014, 1.014, 1-muSE];
yshad = [Re/AU, 1.0829e-04, -1.0829e-04, -Re/AU];

tOrb = 0:360;
xMoon = (1-muSE) + (aMoon/AU)*cosd(tOrb);
yMoon = (aMoon/AU)*sind(tOrb);

xGeo = (1-muSE) + (aGeo/AU)*cosd(tOrb);
yGeo = (aGeo/AU)*sind(tOrb);

figureOVR = figure;
hold on;
plot(xpScope,ypScope,'linewidth',2)
scatter(1-muSE,0,'b*','linewidth',2)
patch(xshad,yshad,[0.4 0.4 0.6],'linewidth',2);
plot(xMoon,yMoon,'linewidth',2,'color',[0.5 0.5 0.5]);
```

```
quiver(1.003,-0.005,0.0037,0.0045,0)
text(1.0027,-0.005,'t=0','FontSize',14,'HorizontalAlignment','right')
% annotation('textarrow',[1.00717285919175 1.003],[0 -0.005],'String','t = 0 ')
% plot(xGeo,yGeo,'k--','linewidth',2);
% scatter(-muSE,0,'y*')
daspect([1 1 1]);
xlim([0.995 1.015])
xlabel('AU')
ylabel('AU')
hold off;
title('Telescope orbit at L2 (rotating frame)')
legend('L2 halo orbit','Earth','Earth''s penumbra','Moon''s orbit','Location','
    northwest')
set(gca,'linewidth',2,'FontSize',14)
saveas(figureOVR,'Scope-position-rotating.png')

%%

figureOVR_all = figure;
hold on;
plot(xpScope,ypScope,'linewidth',2)
scatter(1-muSE,0,'b*','linewidth',2)
scatter(-muSE,0,'r*','linewidth',2)
% scatter([1-muSE-(muSE/3)^(1/3),1-muSE+(muSE/3)^(1/3),-1-muSE-(7*muSE/12),0.5-muSE
    ,0.5-muSE],[0,0,0,sqrt(3)/2,-sqrt(3)/2],'g*','linewidth',2)
scatter([1-muSE-(muSE/3)^(1/3),1-muSE+(muSE/3)^(1/3)],[0,0],'g*','linewidth',2)
% patch(xshad,yshad,[0.4 0.4 0.6],'linewidth',2);
plot(xMoon,yMoon,'linewidth',2,'color',[0.5 0.5 0.5]);
% plot(xGeo,yGeo,'k--','linewidth',2);
% scatter(-muSE,0,'y*')
daspect([1 1 1]);
ylim([-0.1 0.1])
xlim([-0.1 1.1])
hold off;
title('Telescope orbit at L2 (AU, rotating frame)')
legend('L2 Halo orbit','Earth','Sun','L1/L2 Lagrange points','Moon''s orbit','
    Location','northwest')
set(gca,'linewidth',2,'FontSize',14)
saveas(figureOVR_all,'Scope-position-rotating-all-lagrange.png')
```

B.3.2 OrbitCalcs2dome2

Calculates a slice of observations and produces a contour map of thrust vs. time and azimuth (constant elevation).

```
%% Take a slice at constant elevation, sweep over azimuth and time of year.

AU = 1.496e11;
Re = 6371000;
Tnd = 365.25*24*60*60/(2*pi);
muSE = 3.036e-6;

dt = (3600/Tnd);  % 1 hour steps.

yScopeInit = [1.00717285919175; 0; 0; 0; 0.0163636; 0];


tspan = 0:(3600/Tnd):3.2; % ~6 mos, time for scope to stay on-target
%tspan = 0:0.001:1; %~2 mos, time for LGS to stay close without active station-
    keeping
%tspan = linspace(0,24*60*60/Tnd,10000); % 1 day

[tScope,yScopeMat] = ode45(@cr3bpse,tspan,yScopeInit);

xpScope = yScopeMat(:,1);
ypScope = yScopeMat(:,2);
zpScope = yScopeMat(:,3);
xvScope = yScopeMat(:,4);
yvScope = yScopeMat(:,5);
zvScope = yScopeMat(:,6);

xpiScope = xpScope.*cos(tScope) - ypScope.*sin(tScope); % inertial reference frame
ypiScope = ypScope.*cos(tScope) + xpScope.*sin(tScope);
xviScope = xvScope.*cos(tScope) - yvScope.*sin(tScope);
yviScope = yvScope.*cos(tScope) + xvScope.*sin(tScope);

% idxStart = 1; % Start at beginning.
% idxStart = 300; % Trying to get to max turn...
% idxStart = find(ypScope==max(ypScope)); % First corner.
% idxStart = find(xpScope==max(xpScope)); % D loop.

% disp(tStart*365.25/(2*pi))

azvec = 0:0.5:360;
tidxvec = 1:24:(1+24*180);  % Index of time points
```

```
% azvec = 92:0.001:96; % Super high resolution!
% tidxvec = 1:25;  % Once per hour for a day

[tidx,azimuths] = ndgrid(tidxvec,azvec);

accs = zeros(size(azimuths));
thrusts = zeros(size(azimuths));
times = zeros(size(azimuths));

for i = 1:numel(azimuths)

idxStart = tidx(i);
tStart = tScope(idxStart);
times(i) = tStart;

goalAzI = azimuths(i);
goalAzRi = goalAzI-rad2deg(tStart);
goalEl = 0;

desrange = range_LGS/AU;

goalXinit = xpScope(idxStart)+desrange*cosd(goalEl)*cosd(goalAzRi);
goalYinit = ypScope(idxStart)+desrange*cosd(goalEl)*sind(goalAzRi);
goalZinit = zpScope(idxStart)+desrange*sind(goalEl);

velXinit = xvScope(idxStart) + goalYinit - ypScope(idxStart);
velYinit = yvScope(idxStart) - goalXinit + xpScope(idxStart);

goalXi = goalXinit*cos(tStart) - goalYinit*sin(tStart); % inertial reference frame
goalYi = goalYinit*cos(tStart) + goalXinit*sin(tStart);

radInit = sqrt(goalXi^2 + goalYi^2 + goalZinit^2);

yScope = [xpScope(idxStart); ypScope(idxStart); zpScope(idxStart); ...
    xvScope(idxStart); yvScope(idxStart); zvScope(idxStart)]; %

yLGS = [goalXinit; goalYinit; goalZinit; ...
    velXinit; velYinit; zvScope(idxStart)]; %

dydtLGS = cr3bpse(tStart,yLGS);
dydtScope = cr3bpse(tStart,yScope);

xvScoper = dydtScope(1);
yvScoper = dydtScope(2);
```

```
zvScoper = dydtScope(3);
xaScoper = dydtScope(4);
yaScoper = dydtScope(5);
zaScoper = dydtScope(6);

xaScopeic = xaScoper - xpScope(idxStart) - 2*yvScoper; % Coaligned inertial reference
     frame
yaScopeic = yaScoper - ypScope(idxStart) + 2*xvScoper;

xaScopei = xaScopeic*cos(tStart) - yaScopeic*sin(tStart); % rotate
yaScopei = yaScopeic*cos(tStart) + xaScopeic*sin(tStart);

xvLGSr = dydtLGS(1);
yvLGSr = dydtLGS(2);
zvLGSr = dydtLGS(3);
xaLGSr = dydtLGS(4);
yaLGSr = dydtLGS(5);
zaLGSr = dydtLGS(6);

xaLGSic = xaLGSr - goalXinit - 2*yvLGSr; % Coaligned inertial reference frame
yaLGSic = yaLGSr - goalYinit + 2*xvLGSr;

xaLGSi = xaLGSic*cos(tStart) - yaLGSic*sin(tStart); % rotate
yaLGSi = yaLGSic*cos(tStart) + xaLGSic*sin(tStart);

dposi = [desrange*cosd(goalEl)*cosd(goalAzI) ; desrange*cosd(goalEl)*sind(goalAzI) ;
    desrange*sind(goalEl)];
dacci = [xaLGSi-xaScopei ; yaLGSi-yaScopei ; zaLGSr-zaScoper ];

dposin = norm(dposi);
daccin = norm(dacci);

dotp = dot(dposi,dacci);

angle = acosd(dotp/(dposin*daccin));

accreq = daccin*sind(angle);

accreqSI = accreq*(AU/Tnd^2);
TreqSI = accreqSI*sc_mass_opt_tot;

accs(i) = accreqSI;
thrusts(i) = TreqSI;

end
```

```
figureMap = figure;
[Cont,handle] = contour(azimuths,times*(Tnd/(24*60*60)),thrusts*1000,[0.03 0.1 0.2
    0.3 0.5 0.7 1 3],'linewidth',2); % 0, 15, 30, 45, 60, 75 deg
% [C,h] = contour(azimuths,times*(Tnd/(24*60*60)),thrusts*1000); % 89 deg
clabel(Cont,handle,'FontSize',14);
title(sprintf('Thrust req. hold pointing (mN, elev = %d deg, %.2g,000 km)',goalEl,
    range_LGS/1e6))
xlabel('Azimuth (deg)')
ylabel('Mission elapsed time (days)')
set(gca, 'fontsize', 14,'linewidth',2)

saveas(figureMap,sprintf('Cost-of-watching-thrust-time-e%d-fixed.png',goalEl));
```

B.3.3 OrbitCalcs2dome

Calculates a slice of observations and produces a contour map of thrust vs. elevation and azimuth (at one time of observation).

```
% At a given time, show thrust needed vs. az-el

close all
Tnd = 365.25*24*60*60/(2*pi);
muSE = 3.036e-6;

dt = (3600/Tnd);  % 1 hour steps.

yScopeInit = [1.00717285919175; 0; 0; 0; 0.0163636; 0];

tspan = 0:(3600/Tnd):3.2; % ~6 mos, time for scope to stay on-target
%tspan = 0:0.001:1; %~2 mos, time for LGS to stay close without active station-
    keeping
%tspan = linspace(0,24*60*60/Tnd,10000); % 1 day

[tScope,yScopeMat] = ode45(@cr3bpse,tspan,yScopeInit);

xpScope = yScopeMat(:,1);
ypScope = yScopeMat(:,2);
zpScope = yScopeMat(:,3);
xvScope = yScopeMat(:,4);
yvScope = yScopeMat(:,5);
zvScope = yScopeMat(:,6);

xpiScope = xpScope.*cos(tScope) - ypScope.*sin(tScope); % inertial reference frame
```

```
ypiScope = ypScope.*cos(tScope) + xpScope.*sin(tScope);
xviScope = xvScope.*cos(tScope) - yvScope.*sin(tScope);
yviScope = yvScope.*cos(tScope) + xvScope.*sin(tScope);

idxStart = 1; % Start at beginning.
% idxStart = 300; % Trying to get to max turn...
% idxStart = find(ypScope==max(ypScope)); % First corner, 45 days.
% idxStart = find(xpScope==max(xpScope)); % outside of D loop, 93 days.

tStart = tScope(idxStart);
tDays = tStart*365.25/(2*pi);

azvec = 0:0.5:360;
elvec = 0:0.5:90;

[elevs,azimuths] = ndgrid(elvec,azvec);

accs = zeros(size(azimuths));
thrusts = zeros(size(azimuths));

for i = 1:numel(azimuths)

goalAzI = azimuths(i);
goalAzRi = goalAzI-rad2deg(tStart);
goalEl = elevs(i);

desrange = range_LGS/AU;

goalXinit = xpScope(idxStart)+desrange*cosd(goalEl)*cosd(goalAzRi);
goalYinit = ypScope(idxStart)+desrange*cosd(goalEl)*sind(goalAzRi);
goalZinit = zpScope(idxStart)+desrange*sind(goalEl);

velXinit = xvScope(idxStart) + goalYinit - ypScope(idxStart);
velYinit = yvScope(idxStart) - goalXinit + xpScope(idxStart);

goalXi = goalXinit*cos(tStart) - goalYinit*sin(tStart); % inertial reference frame
goalYi = goalYinit*cos(tStart) + goalXinit*sin(tStart);

radInit = sqrt(goalXi^2 + goalYi^2 + goalZinit^2);

yScope = [xpScope(idxStart); ypScope(idxStart); zpScope(idxStart); ...
    xvScope(idxStart); yvScope(idxStart); zvScope(idxStart)]; %

yLGS = [goalXinit; goalYinit; goalZinit; ...
    velXinit; velYinit; zvScope(idxStart)]; %
```

```
dydtLGS = cr3bpse(tStart,yLGS);
dydtScope = cr3bpse(tStart,yScope);

xvScoper = dydtScope(1);
yvScoper = dydtScope(2);
zvScoper = dydtScope(3);
xaScoper = dydtScope(4);
yaScoper = dydtScope(5);
zaScoper = dydtScope(6);

xaScopeic = xaScoper - xpScope(idxStart) - 2*yvScoper; % Coaligned inertial reference
     frame
yaScopeic = yaScoper - ypScope(idxStart) + 2*xvScoper;

xaScopei = xaScopeic*cos(tStart) - yaScopeic*sin(tStart); % rotate
yaScopei = yaScopeic*cos(tStart) + xaScopeic*sin(tStart);

xvLGSr = dydtLGS(1);
yvLGSr = dydtLGS(2);
zvLGSr = dydtLGS(3);
xaLGSr = dydtLGS(4);
yaLGSr = dydtLGS(5);
zaLGSr = dydtLGS(6);

xaLGSic = xaLGSr - goalXinit - 2*yvLGSr; % Coaligned inertial reference frame
yaLGSic = yaLGSr - goalYinit + 2*xvLGSr;

xaLGSi = xaLGSic*cos(tStart) - yaLGSic*sin(tStart); % rotate
yaLGSi = yaLGSic*cos(tStart) + xaLGSic*sin(tStart);

dposi = [desrange*cosd(goalEl)*cosd(goalAzI) ; desrange*cosd(goalEl)*sind(goalAzI) ;
    desrange*sind(goalEl)];
dacci = [xaLGSi-xaScopei ; yaLGSi-yaScopei ; zaLGSr-zaScoper ];

dacciInline = dposi*dot(dposi,dacci)/dot(dposi,dposi);
dacciCross = dacci-dacciInline;

dposin = norm(dposi);
daccin = norm(dacci);

dotp = dot(dposi,dacci);

angle = acosd(dotp/(dposin*daccin));
```

```
accreq = daccin*sind(angle);
% accreq = norm(dacciCross);

accreqSI = accreq*(AU/Tnd^2);
TreqSI = accreqSI*sc_mass_opt_tot;

accs(i) = accreqSI;
thrusts(i) = TreqSI;

end


figureMap = figure;
colormap cool
axesm('MapProjection','robinson','Grid','on','GLineWidth',2)
[Cont,handle] = contourm(elevs,azimuths,thrusts*1000,[0.03 0.1 0.3 0.5 0.7 1 3],'
    linewidth',2); % 0 days, 100k km
[Cont2,handle2] = contourm(-elevs,azimuths,thrusts*1000,[0.03 0.1 0.3 0.5 0.7 1 3],'
    linewidth',2); % 0 days, 100k km
% [C,h] = contour(azimuths,elevs,thrusts*1000,[0.01 0.05 0.09 0.13]); % 0 days, 10k
    km sep
% [C,h] = contour(azimuths,elevs,thrusts*1000,[0.01 0.05 0.09 0.13]); % 45/90 days,
    10k km sep
% [C,h] = contour(azimuths,elevs,thrusts*1000);
% [C,h] = contour(azimuths,elevs,thrusts*1000,[0.03 0.1 0.2 0.3 0.5 0.7 1 3]); % 90
    days, D loop, 100k km sep
% clabel(C,h,'FontSize',14);
htext = clabelm(Cont,handle);
set(htext,'fontsize',14);
title(sprintf('Thrust req. hold pointing (mN, t = %d days, %.2g,000 km)',round(tDays)
    ,range_LGS/1e6))
colorbar
% xlabel('Ecliptic longitude (deg)')
% ylabel('Ecliptic latitude (deg)')
set(gca, 'fontsize', 14,'linewidth',2)

saveas(figureMap,sprintf('Cost-of-watching-thrust-t0-fixed-%.2gk.png',range_LGS/1e6))
    ;
% colormap default

% figureSphere = figure;
% xthr = (2.7e-4-thrusts).*cosd(elevs).*cosd(azimuths);  % Exactly 100,000 km away,
    in the proper direction
% ythr = (2.7e-4-thrusts).*cosd(elevs).*sind(azimuths);
% zthr = (2.7e-4-thrusts).*sind(elevs);
```

```
% plot3(xthr,ythr,zthr)
% daspect([1 1 1])
%
% set(findall(gcf,'-property','FontSize'),'FontSize',14);
```

B.3.4 OrbitCalcs3

Calculates the time and delta-V cost of deploying a LGS from LUVOIR.

```
Tnd = 365.25*24*60*60/(2*pi);

yScopeInit = [1.00717285919175; 0; 0; 0; 0.0163636; 0];

tspan = 0:(3600/Tnd):3.2; % ~6 mos, time for scope to stay on-target

[tScope,yScopeMat] = ode45(@cr3bpse,tspan,yScopeInit);

xpScope = yScopeMat(:,1);
ypScope = yScopeMat(:,2);
zpScope = yScopeMat(:,3);
xvScope = yScopeMat(:,4);
yvScope = yScopeMat(:,5);
zvScope = yScopeMat(:,6);

yLGSInit = [xpScope(1); ypScope(1); zpScope(1); ...
    xvScope(1)+1.44*(Tnd/AU); yvScope(1); zvScope(1)]; %adding pushoff velocity

[tLGS,yLGSMat] = ode45(@cr3bpse,tspan,yLGSInit);

xpLGS = yLGSMat(:,1);
ypLGS = yLGSMat(:,2);
zpLGS = yLGSMat(:,3);
xvLGS = yLGSMat(:,4);
yvLGS = yLGSMat(:,5);
zvLGS = yLGSMat(:,6);

dist = sqrt((xpLGS-xpScope).^2+(zpLGS-zpScope).^2+(ypLGS-ypScope).^2);
dvs = sqrt((xvLGS-xvScope).^2+(zvLGS-zvScope).^2+(yvLGS-yvScope).^2);

drift_thresh = find(dist*AU>range_LGS);

drift_time = Tnd*tspan(drift_thresh(1))/daysec;
drift_vel = dvs(drift_thresh(1))*AU/Tnd;
```

```
decel_time = (drift_vel*sc_mass_opt_tot/sc_max_thrust_nom)/daysec;
```

B.3.5 OrbitCalcs3CL3

Simulates the formation flight of a laser guide star with a telescope, including closed-loop control, for evaluating drift and noise sensitivity.

```
Tnd = 365.25*24*60*60/(2*pi);
muSE = 3.036e-6;
x1 = -muSE;
x2 = 1-muSE;

dt = (3600/Tnd);  % 1 hour steps.

yScopeInit = [1.00717285919175; 0; 0; 0; 0.0163636; 0];

tspan = 0:(3600/Tnd):3.2; % ~6 mos, time for scope to stay on-target
% tspan = 0:(60/Tnd):(1*24*3600/Tnd); % 1 day
%tspan = 0:0.001:1; %~2 mos, time for LGS to stay close without active station-
    keeping
%tspan = linspace(0,24*60*60/Tnd,10000); % 1 day

[tScope,yScopeMat] = ode45(@cr3bpse,tspan,yScopeInit);

xpScope = yScopeMat(:,1);
ypScope = yScopeMat(:,2);
zpScope = yScopeMat(:,3);
xvScope = yScopeMat(:,4);
yvScope = yScopeMat(:,5);
zvScope = yScopeMat(:,6);

xpiScope = xpScope.*cos(tScope) - ypScope.*sin(tScope); % inertial reference frame
ypiScope = ypScope.*cos(tScope) + xpScope.*sin(tScope);
xviScope = xvScope.*cos(tScope) - yvScope.*sin(tScope) - ypiScope;
yviScope = yvScope.*cos(tScope) + xvScope.*sin(tScope) + xpiScope;

xaScope = xpScope + 2.*yvScope -(1-muSE).*(xpScope-x1)./((xpScope-x1).^2 + ypScope.^2
     + zpScope.^2).^(3./2) -muSE.*(xpScope-x2)./((xpScope-x2).^2 + ypScope.^2 +
    zpScope.^2).^(3./2);
yaScope = ypScope - 2.*xvScope -(1-muSE).*ypScope./((xpScope-x1).^2 + ypScope.^2 +
    zpScope.^2).^(3./2) -muSE.*ypScope./((xpScope-x2).^2 + ypScope.^2 + zpScope.^2)
    .^(3./2);
zaScope = -(1-muSE).*zpScope./((xpScope-x1).^2 + ypScope.^2 + zpScope.^2).^(3./2) -
    muSE.*zpScope./((xpScope-x2).^2 + ypScope.^2 + zpScope.^2).^(3./2);
```

```
xaicScope = xaScope - xpScope - 2.*yvScope;
yaicScope = yaScope - ypScope + 2.*xvScope;

xaiScope = xaicScope.*cos(tScope) - yaicScope.*sin(tScope);
yaiScope = yaicScope.*cos(tScope) + xaicScope.*sin(tScope);

zaiScope = zaScope;

% plot(xpScope,ypScope,xpiScope,ypiScope)
%plot(xvScope,yvScope,xviScope,yviScope)
%plot(xaScope,yaScope,xaiScope,yaiScope)
% daspect([1 1 1])
% figurePVA = figure;
% plot(tScope,xpiScope,tScope,xviScope,tScope,xaiScope)
% xlabel('Time (radians)')
% ylabel('$x, \dot{x}, \ddot{x}$ (AU, AU/rad, AU/rad$^2$)','interpreter','latex')
% title('X-position, velocity, acceleration of L2 halo orbit (inertial frame)')
% saveas(figurePVA,'PVA.png')

% clc

idxStart = 1; % Start at beginning.
% idxStart = 300; % Trying to get to max turn...
% idxStart = find(ypScope==max(ypScope)); % First corner.
% idxStart = find(xpScope==max(xpScope)); % D loop.

tStart = tScope(idxStart);

goalAzI = 0;
goalAzRi = goalAzI-rad2deg(tStart);
goalEl = 0;

goalRange = range_LGS/AU;

goalXinit = xpScope(idxStart)+goalRange*cosd(goalEl)*cosd(goalAzRi);  % Exactly
    100,000 km away, in the proper direction
goalYinit = ypScope(idxStart)+goalRange*cosd(goalEl)*sind(goalAzRi);
goalZinit = zpScope(idxStart)+goalRange*sind(goalEl);
%goalYinit = goalYinit + 100/AU; % Adding 100 m of position error

velXinit = xvScope(idxStart) + goalYinit - ypScope(idxStart);
velYinit = yvScope(idxStart) - goalXinit + xpScope(idxStart);
% velYinit = velYinit + 0.0001*(Tnd/AU); % Adding 0.1 mm/s of velocity error
```

```
yLGSInit = [xpScope(idxStart); ypScope(idxStart); zpScope(idxStart); ...
    xvScope(idxStart); yvScope(idxStart); zvScope(idxStart); ...
    goalXinit; goalYinit; goalZinit; ...
    velXinit; velYinit; zvScope(idxStart); ...
    goalAzI; goalEl; goalRange; ...
    0; 0; 0]; %

% tspan2 = tStart:(60/Tnd):(tStart+(3600*24*15/Tnd)); % 15 days, one-minute intervals
    .
tspan2 = tStart:(60/Tnd):(tStart+(3600*24/Tnd)); % one day, one-minute intervals.
% tspan2 = tStart:(60/Tnd):(tStart+(3600*3*24/Tnd)); % three days, one-minute
    intervals.

[tLGS,yLGSMat] = ode45(@cr3bpsepropCLazel3,tspan2,yLGSInit);

rpts = zeros(size(tspan2,2),3);
rptsnorm = zeros(size(tspan2,2),1);

for i = 1:size(tspan2,2)
    rpt = cr3bpsepropCLazel3rpt(tspan2(i),yLGSMat(i,:));
    rpts(i,:) = rpt;
    rptsnorm(i) = norm(rpt);
end

rptsI = rpts;
rptsI(:,1) = rpts(:,1).*cos(tLGS) - rpts(:,2).*sin(tLGS);
rptsI(:,2) = rpts(:,2).*cos(tLGS) + rpts(:,1).*sin(tLGS);

xpScope2 = yLGSMat(:,1);
ypScope2 = yLGSMat(:,2);
zpScope2 = yLGSMat(:,3);
xvScope2 = yLGSMat(:,4);
yvScope2 = yLGSMat(:,5);
zvScope2 = yLGSMat(:,6);

xpiScope2 = xpScope2.*cos(tLGS) - ypScope2.*sin(tLGS); % inertial reference frame
ypiScope2 = ypScope2.*cos(tLGS) + xpScope2.*sin(tLGS);
xviScope2 = xvScope2.*cos(tLGS) - yvScope2.*sin(tLGS);
yviScope2 = yvScope2.*cos(tLGS) + xvScope2.*sin(tLGS);

xpLGS = yLGSMat(:,7);
ypLGS = yLGSMat(:,8);
zpLGS = yLGSMat(:,9);
xvLGS = yLGSMat(:,10);
yvLGS = yLGSMat(:,11);
```

```
zvLGS = yLGSMat(:,12);

xpiLGS = xpLGS.*cos(tLGS) - ypLGS.*sin(tLGS); % inertial reference frame
ypiLGS = ypLGS.*cos(tLGS) + xpLGS.*sin(tLGS);
xviLGS = xvLGS.*cos(tLGS) - yvLGS.*sin(tLGS); % inertial reference frame
yviLGS = yvLGS.*cos(tLGS) + xvLGS.*sin(tLGS);

dxp = xpLGS-xpScope2;
dyp = ypLGS-ypScope2;
dzp = zpLGS - zpScope2;

range = sqrt(dxp.^2 + dyp.^2 + dzp.^2);

dxpi = xpiLGS - xpiScope2;
dypi = ypiLGS - ypiScope2;

angleAz = atan2(dypi,dxpi);
angleAzR = atan2(dyp,dxp);
dangAz = angleAz(2:end) - angleAz(1:end-1);
angleEl = atan2(dzp,sqrt(dxpi.^2+dypi.^2));

dyvi = yviLGS - yviScope2;

%%

close all

xshad = [1-muSE, 1.014, 1.014, 1-muSE];
yshad = [Re/AU, 1.0829e-04, -1.0829e-04, -Re/AU];
figureOVR = figure;
hold on;
patch(xshad,yshad,[0.4 0.4 0.6]);
plot(xpScope,ypScope,xpLGS,ypLGS)
scatter(1-muSE,0,'b*')
xlim([0.995 1.015])
daspect([1 1 1]);
hold off;
title('Telescope and LGS at L2 (AU, rotating frame)')
legend('Earth''s penumbra','Scope','Earth','LGS','Location','northwest')
saveas(figureOVR,'LGS-Scope-position-rotating.png');

figureOVI = figure;
hold on;
plot(((xpiLGS-xpiScope2)*AU/1000e3),(ypiLGS-ypiScope2)*AU, 'linewidth', 2)
% scatter(0,0,'b*', 'linewidth', 2)
```

```
plot([1.1*range_LGS/1e6,range_LGS/1e6,range_LGS/1e6,1.1*range_LGS/1e6],[1.1*
    iwa_box_rad,iwa_box_rad,-iwa_box_rad,-1.1*iwa_box_rad], 'linewidth', 2)
plot([range_LGS/1e6,1.1*range_LGS/1e6],[0, 0],':','linewidth',2)
% scatter(1-muSE,0,'b*')
% daspect([1 1 1]);
% hold off;
xlim([4.31e1,4.35e1])
ylim([-1,1])
title('LGS relative position to Scope (inertial frame)')
xlabel('Distance from telescope (1000''s of km)')
ylabel('Distance from line of sight (m)')
% legend('LGS','Scope','Goal','Location','northwest')
legend('LGS','Requirement (deep IWA, 0.25 lam/D)','Goal')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figureOVI,'LGS-Scope-position-inertial.png');

figureAng = figure;
plot(tspan2.*Tnd./(24*60*60),angleAz*1e9,tspan2.*Tnd./(24*60*60),(iwa_box_rad/
    range_LGS)*1e9*ones(size(tspan2)))
% daspect([1 1 1]);
title('Scope-LGS vector azimuth (ecliptic plane, inertial frame)')
xlabel('Mission time [days]')
ylabel('Scope-LGS angle, nrad')
% saveas(figureAng,'LGS-Scope-Ecl-Ang-10k-CL-old.png');
saveas(figureAng,'LGS-Scope-Ecl-Ang-10k-CL-new.png');

figureTHR = figure;
plot(tspan2*Tnd/(3600*24),rptsnorm*(AU/Tnd^2)*24*1000)
title('Thrust over time (24 kg) for drift compensation')
xlabel('Mission time (days)')
ylabel('Total thrust (mN)')

figureTHR2I = figure;
hold on
plot(rptsI(:,1)*(AU/Tnd^2)*24*1000,rptsI(:,2)*(AU/Tnd^2)*24*1000)
quiver(0,0,0.5*cos(goalAzI),0.5*sin(goalAzI))
hold off
xlim([-1 1])
ylim([-1 1])
daspect([1 1 1])
title('Thrust vector (inertial space, 24 kg) for drift compensation')
legend('Thrust vector','Goal line of sight')
xlabel('X-thrust (mN)')
ylabel('Y-thrust (mN)')
```

B.3.6 `cr3bpse`

A differential-equation representation of the circular restricted 3-body problem, for use with `ode45`. Used by the various OrbitCalcs scripts.

```
function dydt = cr3bpse(t,y)
% non-dimentional circular restricted 3-body problem for Sun-Earth.
% y(1) = xp, y(2) = yp, y(3) = zp,
% y(4) = xv, y(5) = yv, y(6) = zv
    xp = y(1);
    yp = y(2);
    zp = y(3);
    xv = y(4);
    yv = y(5);
    zv = y(6);

    muSE = 3.036e-6;
    x1 = -muSE;
    x2 = 1-muSE;

    xa = xp + 2*yv -(1-muSE)*(xp-x1)/((xp-x1)^2 + yp^2 + zp^2)^(3/2) -muSE*(xp-x2)/((
    xp-x2)^2 + yp^2 + zp^2)^(3/2);
    ya = yp - 2*xv -(1-muSE)*yp/((xp-x1)^2 + yp^2 + zp^2)^(3/2) -muSE*yp/((xp-x2)^2 +
     yp^2 + zp^2)^(3/2);
    za = -(1-muSE)*zp/((xp-x1)^2 + yp^2 + zp^2)^(3/2) -muSE*zp/((xp-x2)^2 + yp^2 + zp
    ^2)^(3/2);

    dydt = [xv; yv; zv; xa; ya; za];
end
```

B.3.7 `cr3bpsepropCLazel3`

An advanced version of `cr3bpse` which includes a rudimentary controller, for investigating formation flight.

```
function [dydt] = cr3bpsepropCLazel3(t,y)
% non-dimensional circular restricted 3-body problem for Sun-Earth.
% y(1) = xp, y(2) = yp, y(3) = zp,  % all of Scope
% y(4) = xv, y(5) = yv, y(6) = zv
% y(7) = xp, y(8) = yp, y(9) = zp,  % all of Laser
% y(10) = xv, y(11) = yv, y(12) = zv
% y(13) = goalAz, y(14) = goalEl, y(15) = goalRange

muSE = 3.036e-6;
```

```
AU = 1.496e11;
Tnd = 365.25*24*60*60/(2*pi);
% tDays = t*Tnd/(24*60*60);
x1 = -muSE;
x2 = 1-muSE;

xpS = y(1);
ypS = y(2);
zpS = y(3);
xvS = y(4);
yvS = y(5);
zvS = y(6);
xpL = y(7);
ypL = y(8);
zpL = y(9);
xvL = y(10);
yvL = y(11);
zvL = y(12);

goalAz = y(13);
goalEl = y(14);
goalRange = y(15);

dintXI = y(16);
dintYI = y(17);
dintZI = y(18);

xpiS = xpS.*cos(t) - ypS.*sin(t); % inertial reference frame
ypiS = ypS.*cos(t) + xpS.*sin(t);

xpiL = xpL.*cos(t) - ypL.*sin(t); % inertial reference frame
ypiL = ypL.*cos(t) + xpL.*sin(t);

goalX = xpiS+goalRange*cosd(goalEl)*cosd(goalAz);
goalY = ypiS+goalRange*cosd(goalEl)*sind(goalAz);
goalZ = zpS+goalRange*sind(goalEl);

% dxpi = xpiL-xpiS;
% dypi = ypiL-ypiS;
% dzpi = zpL - zpS;

% To convert velocities, first convert to instantaneously aligned inertial
% reference frame, then remove rotation.

xviS = xvS.*cos(t) - yvS.*sin(t) - ypiS;
```

```
yviS = yvS.*cos(t) + xvS.*sin(t) + xpiS;

xviL = xvL.*cos(t) - yvL.*sin(t) - ypiL;
yviL = yvL.*cos(t) + xvL.*sin(t) + xpiL;

% dxpr = xpL - xpS;
% dypr = ypL - ypS;

dxvi = xviL-xviS;
dyvi = yviL-yviS;
dzvi = zvL - zvS;

% range = sqrt(dxpi.^2 + dypi.^2 + dzpi.^2);
% rangeH = sqrt(dxpi.^2 + dypi.^2);

% angleAz = atan2(dypi,dxpi);
% angleAzR = atan2(dypr,dxpr);
% dangleAz = (dxpi*dyvi - dypi*dxvi)/rangeH^2; % Probably not stable around poles.
% angleEl = atan2(dzpi,rangeH);
% dangleEl = (rangeH*dzvi - dzpi*(dxpi*dxvi + dypi*dyvi)/rangeH)/(range^2); %
    definitely not stable around poles!

% errAz = angleAz-goalAz;
% errEl = angleEl-goalEl;

% corrAz = angleAzR - (pi/2)*sign(errAz);

xaS = xpS + 2*yvS -(1-muSE)*(xpS-x1)/((xpS-x1)^2 + ypS^2 + zpS^2)^(3/2) -muSE*(xpS-x2
    )/((xpS-x2)^2 + ypS^2 + zpS^2)^(3/2);
yaS = ypS - 2*xvS -(1-muSE)*ypS/((xpS-x1)^2 + ypS^2 + zpS^2)^(3/2) -muSE*ypS/((xpS-x2
    )^2 + ypS^2 + zpS^2)^(3/2);
zaS = -(1-muSE)*zpS/((xpS-x1)^2 + ypS^2 + zpS^2)^(3/2) -muSE*zpS/((xpS-x2)^2 + ypS^2
    + zpS^2)^(3/2);

xaL = xpL + 2*yvL -(1-muSE)*(xpL-x1)/((xpL-x1)^2 + ypL^2 + zpL^2)^(3/2) -muSE*(xpL-x2
    )/((xpL-x2)^2 + ypL^2 + zpL^2)^(3/2);
yaL = ypL - 2*xvL -(1-muSE)*ypL/((xpL-x1)^2 + ypL^2 + zpL^2)^(3/2) -muSE*ypL/((xpL-x2
    )^2 + ypL^2 + zpL^2)^(3/2);
zaL = -(1-muSE)*zpL/((xpL-x1)^2 + ypL^2 + zpL^2)^(3/2) -muSE*zpL/((xpL-x2)^2 + ypL^2
    + zpL^2)^(3/2);

% xaD = xaL - xaS;
% yaD = yaL - yaS;
zaD = zaL - zaS;
```

```
xaicS = xaS - xpS - 2*yvS;  % First, convert to an inertial reference frame that is
    instantaneously co-aligned with the rotating reference frame.
yaicS = yaS - ypS + 2*xvS;

xaicL = xaL - xpL - 2*yvL;
yaicL = yaL - ypL + 2*xvL;

xaiS = xaicS.*cos(t) - yaicS.*sin(t); % Then rotate to the actual angle of inertial
    space.
yaiS = yaicS.*cos(t) + xaicS.*sin(t);

xaiL = xaicL.*cos(t) - yaicL.*sin(t);
yaiL = yaicL.*cos(t) + xaicL.*sin(t);

xaiD = xaiL - xaiS;
yaiD = yaiL - yaiS;

losvec = [cosd(goalEl)*cosd(goalAz); cosd(goalEl)*sind(goalAz); sind(goalEl)];

dposIx = xpiL-goalX;
dposIy = ypiL-goalY;
dposIz = zpL-goalZ;

daccI = [xaiD ; yaiD ; zaD]; % Insert sensor noise here
dvelI = [dxvi ; dyvi ; dzvi];
dposI = [dposIx; dposIy; dposIz];
dintI = [dintXI ; dintYI ; dintZI];

% dvelIinline = losvec*dot(dvelI,losvec);
% dvelIperp = dvelI-dvelIinline;

dposIinline = losvec*dot(dposI,losvec);
dposIperp = dposI-dposIinline;

if mod(t,3600*24/Tnd) == 0 % Every 24 hours
    format long
    disp(dposIperp*AU);
    format short
end

kacc = 1;% + 0.01*tanh(norm(dposIperp)*AU/4); Do not change from 1!
kvel = 10;% + 9*tanh(norm(dposIperp)*AU/4); %1000; % Velocity control scaling -- ~1
    seems to work best at low displacements, ~10 at moderate
kpos = 1000;% + 1000*tanh(norm(dposIperp)*AU/400); %100000; % Position control
    scaling, needs ~1000 to matter
```

```
kint = 100 + 1000*tanh(norm(dposIperp)*AU/4);%1000; % Integral of position control
    scaling.

daccIinit = -kacc*daccI - kvel*dvelI - kpos*dposI -kint*dintI;

daccIinline = losvec*(dot(daccIinit,losvec));
daccIperp = daccIinit-daccIinline;

daccIcommand = daccIperp;
% daccIcommand = daccIinit;

% Insert thruster noise here

% daccIcommand = daccIcommand.*random('Normal',1,0.01,size(daccIcommand));

xThrI = daccIcommand(1);
yThrI = daccIcommand(2);
zThr = daccIcommand(3);

xThrR = xThrI.*cos(t) + yThrI.*sin(t);
yThrR = yThrI.*cos(t) - xThrI.*sin(t);

xaL = xaL + xThrR;
yaL = yaL + yThrR;
zaL = zaL + zThr;

dydt = [xvS; yvS; zvS; xaS; yaS; zaS; xvL; yvL; zvL; xaL; yaL; zaL; 0; 0; 0; dposIx;
    dposIy; dposIz];
% rpt = [xThrR;yThrR;zThr];

end
```

B.3.8 cr3bpsepropCLazel3rpt

A variant of cr3bpsepropCLazel3 which includes additional return variables; it is not to be used with ode45, but can be used on the trajectory coming out of ode45 to see what the controller in cr3bpsepropCLazel3 was doing.

```
function [rpt] = cr3bpsepropCLazel3rpt(t,y)
% Extract thrust metrics
% non-dimensional circular restricted 3-body problem for Sun-Earth.
% y(1) = xp, y(2) = yp, y(3) = zp,  % all of Scope
% y(4) = xv, y(5) = yv, y(6) = zv
```

```
% y(7) = xp, y(8) = yp, y(9) = zp,  % all of Laser
% y(10) = xv, y(11) = yv, y(12) = zv
% y(13) = goalAz, y(14) = goalEl, y(15) = goalRange

muSE = 3.036e-6;
x1 = -muSE;
x2 = 1-muSE;

xpL = y(7);
ypL = y(8);
zpL = y(9);
xvL = y(10);
yvL = y(11);

% Accelerations under CR3BP only

xaL = xpL + 2*yvL -(1-muSE)*(xpL-x1)/((xpL-x1)^2 + ypL^2 + zpL^2)^(3/2) -muSE*(xpL-x2
    )/((xpL-x2)^2 + ypL^2 + zpL^2)^(3/2);
yaL = ypL - 2*xvL -(1-muSE)*ypL/((xpL-x1)^2 + ypL^2 + zpL^2)^(3/2) -muSE*ypL/((xpL-x2
    )^2 + ypL^2 + zpL^2)^(3/2);
zaL = -(1-muSE)*zpL/((xpL-x1)^2 + ypL^2 + zpL^2)^(3/2) -muSE*zpL/((xpL-x2)^2 + ypL^2
    + zpL^2)^(3/2);

dydt = cr3bpsepropCLazel3(t,y);

xaLt = dydt(10);
yaLt = dydt(11);
zaLt = dydt(12);

rpt = [xaLt-xaL;yaLt-yaL;zaLt-zaL];

end
```

B.4 LGS performance

B.4.1 NoiseCalcsPropSens

Calculates the sensitivity of the formation to thruster noise and calculates the required check-in frequency from the telescope during an observation.

Watch out for the λ characters, they might come through strangely if you copy-paste this code from this document instead of getting it from the repository.

```
%% Sensor and thruster noise calcs

close all;

max_upd_int = 6000;

upd_int_vec = 10:10:max_upd_int; % update interval

max_zerocost_pos_err = 0.25*max_bg_acc*upd_int_vec.^2; % run OrbitCalcs2dome3 first
typ_zerocost_pos_err = 0.25*avg_acc*upd_int_vec.^2;

figMaxPosErr = figure;
semilogy(upd_int_vec/60,max_zerocost_pos_err,'-',upd_int_vec/60,typ_zerocost_pos_err,
    '-',[0 max_upd_int/60],[seg_box_rad seg_box_rad],'--',[0 max_upd_int/60],[
    iwa_box_rad_relax iwa_box_rad_relax],'--',[0 max_upd_int/60],[iwa_box_rad
    iwa_box_rad],'--','linewidth', 2)
title('Maximum location error without penalty')
ylabel('LGS velocity-axis error (m)')
xlabel('Update interval (minutes)')
legend('Worst-case observation','Average observation',sprintf('Goal: +/- %.2g m (flat
     waves on segments)',seg_box_rad),sprintf('Goal: +/- %.2g m (stay in IWA, 1 λ/D)'
    ,iwa_box_rad_relax),sprintf('Goal: +/- %.2g m (stay in deep IWA, 0.25 λ/D)',
    iwa_box_rad),'Location','southeast')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figMaxPosErr,'max_pos_err_nocost.png')

seg_upd_time = 2*sqrt(seg_box_rad/max_bg_acc);
iwa_upd_time = 2*sqrt(iwa_box_rad/max_bg_acc);
iwa_upd_time_relax = 2*sqrt(iwa_box_rad_relax/max_bg_acc);

seg_upd_time_typ = 2*sqrt(seg_box_rad/avg_acc);
iwa_upd_time_typ = 2*sqrt(iwa_box_rad/avg_acc);
iwa_upd_time_typ_relax = 2*sqrt(iwa_box_rad_relax/avg_acc);

max_thr_err = 0.25*max_bg_thrust;
```

```
max_ang_err = asind(0.25);
```

B.4.2 NoiseCalcs

Calculates the possible negative effects of the LGS spacecraft on the telescope's observation (thermal emission, sun glinting, etc.).

```
%% Photon noise calcs

LGS_size = 0.3; % 30 cm height
LGS_ang_size = LGS_size/range_LGS; % 7 nrad = 1.5 mas
planet_ang_size = 2*Re/(10*parsec); % 40 picorad = 8 uas
worst_case_planet_ang_size = 2*11*Re/(4.3*ly); % Jupiter around Alpha Cen would be 3
    nrad (0.6 mas) across, still smaller than LGS.

area = 5*0.01*0.01; % 1 cm2 of area
flux = 1368; % W/m2
div = deg2rad(0.5); % divergence from solar size, flat reflector
scopearea = 0.25*pi*scope_d^2;
spotsize = range_LGS*div;
spotarea = pi*0.25*spotsize^2;
spherearea = 4*pi*0.25*range_LGS^2;
fluxrec = flux*area/spotarea;

recangle = scope_d/range_LGS;
recSA = 2*pi*(1-cos(recangle/2));

Imax = flux/pi;
fluxrec2 = Imax*recSA*area/scopearea;

fluxrecSilly = flux*area/spherearea; % What if we just radiate isotropically

areaFull = 0.2*0.3; % What if we put a full 6U reflection down LUVOIR?
fluxrecFull = flux*areaFull/spotarea;



flux0jy = 3640; % zero magnitude in V band, per http://www.astro.umd.edu/~ssm/ASTR620
    /mags.html, janskys
flux0pps = 1.51e7*flux0jy*0.16;
Ephot = h*c/(500e-9);
flux0wm2 = flux0pps*Ephot;
mag = -2.5*log10(fluxrec/flux0wm2);
mag2 = -2.5*log10(fluxrec2/flux0wm2);
magSilly = -2.5*log10(fluxrecSilly/flux0wm2);
```

```
magFull = -2.5*log10(fluxrecFull/flux0wm2);

% IR emissions, sum up to 2500 nm (LUVOIR sensitivity)

lambdas = (100:2500)*1e-9;
Blams = PlancksLaw(lambdas,273+30);
pow_ir = 0.03*(sum(Blams)*1e-9)*4*pi*(4*.2*.3); % W/m^2/sr/nm * nm * sr * m^2 = W
% Summing over all lambda, but really it's just 1700 nm+ that matters (i.e.
% K band).  The factor of 0.03 is the IR emissivity of aluminum.
pow_ir_rec = pow_ir*(0.25*pi*scope_d^2)/(4*pi*range_LGS^2); % how much IR received

Fx0 = 670;
BWwideband = 390/2190;
Arx = 0.25*pi*scope_d^2;

Ephot = h.*c./2385e-9;
Photrx = pow_ir_rec./Ephot;

FJy = Photrx.*h.*1e26./(BWwideband.*Arx);

appMag = -2.5.*log10(FJy./Fx0);
```

B.4.3 PlancksLaw

An implementation of Planck's Law.

```
function [Blam] = PlancksLaw(lambda,T)

h = 6.626e-34;
c = 299792458;
kB = 1.38065e-23;

Blam = (2.*h.*c^2./lambda.^5)./(exp(h.*c./(lambda.*kB.*T))-1);

end
```

B.4.4 PowerCalcs

Calculations of power and ADCS subsystem performance and requirements during formation flight.

```
% Power needs
% 4x 6U panels and "2x2U" panels from GOMSpace
maxPwrGen = 1.15*(4*16+2*5); % 85 W

maxPwrThrust = 0.3*75 + 0.7*30; % Power cycle of the thruster
maxPwrLaser = 30;
maxPwrDraw = maxPwrLaser+ maxPwrThrust;

typPwrGen = 0.7*maxPwrGen;

pwrDef = maxPwrDraw-typPwrGen;

pwr_cap = 77*2; % Whr

pwrDur = pwr_cap/pwrDef;

pwrAng = acosd(maxPwrDraw/maxPwrGen);

% ADCS
% Sun torque

torque_solar = (solarConst/c)*(4*.2*.3+.2*.2)*cosd(45)*1.5*(0.15)*cosd(45); % Solar
    torque from the "flower" panels.

max_total_ang_mom = torque_solar*max(obs_dur)*daysec;
torque_thruster = speed_factor*sc_max_thrust_nom*sind(10)*0.15; % max torque from the
     thruster
desat_factor = torque_thruster/torque_solar;
```

B.4.5 LGSretro

Calculations of the brightness of a "laser guide star" spacecraft that reflects a laser generated externally (LUVOIR itself, or a ground-based facility).

```
% Generate the laser somewhere else, the LGS reflects it to the telescope

% Let's give LUVOIR a laser that is 10X bigger and more powerful than LGS
pwr_laser_scope = 10*pwr_laser;
D_laser_scope = 10*D_laser;

LGS_face_area = 0.2*0.3; % 20x30 cm face for the retro
eff_retro_d = sqrt(4*LGS_face_area/pi);

```

```
% LUVOIR -> LGS
[PrxTx,PhotrxTx,appMagTx,bwTx] = linkbudgetG(pwr_laser_scope,D_laser_scope,range_LGS,
    lambda,eff_retro_d);

% LGS back to LUVOIR
[PrxRx,PhotrxRx,appMagRx,bwRx] = linkbudgetG(PrxTx,eff_retro_d,range_LGS,lambda,
    scope_d);

% Case 2: using LUVOIR's main telescope itself as the transmitter!

% LUVOIR -> LGS
[PrxTx2,PhotrxTx2,appMagTx2,bwTx2] = linkbudgetG(pwr_laser_scope,scope_d,range_LGS,
    lambda,eff_retro_d);

% LGS back to LUVOIR
[PrxRx2,PhotrxRx2,appMagRx2,bwRx2] = linkbudgetG(PrxTx2,eff_retro_d,range_LGS,lambda,
    scope_d);

% Giant ABL-TMT facility (Is such a thing even possible? Yes it is.)

pwr_laser_gnd = 1e6;
D_laser_gnd = 30;
lambda_abl = 1315e-9;
range_L2 = ((muSE/3)^(1/3))*AU;

% ABL-TMT -> LGS, assuming no atmosphere (!!!)
[PrxTx3,PhotrxTx3,appMagTx3,bwTx3] = linkbudgetG(pwr_laser_gnd,D_laser_gnd,range_L2,
    lambda_abl,eff_retro_d);

% LGS back to LUVOIR
[PrxRx3,PhotrxRx3,appMagRx3,bwRx3] = linkbudgetG(PrxTx3,eff_retro_d,range_LGS,
    lambda_abl,scope_d);

% Now let's take atmospheric effects into account
D_laser_eff_gnd = 0.1; % Fried parameter ~10 cm eff D

% ABL-TMT -> LGS
[PrxTx4,PhotrxTx4,appMagTx4,bwTx4] = linkbudgetG(pwr_laser_gnd,D_laser_eff_gnd,
    range_L2,lambda_abl,eff_retro_d);

% LGS back to LUVOIR
[PrxRx4,PhotrxRx4,appMagRx4,bwRx4] = linkbudgetG(PrxTx4,eff_retro_d,range_LGS,
    lambda_abl,scope_d);
```

B.5 Design Reference Mission (DRM)

B.5.1 StarkSkymap

Ingests lists of stars and observations for later use, and plots them on a map.

```
close all

fileID = fopen('simbad-trim.csv','r');
C = textscan(fileID,['HIP' '%d' ';' '%f' '%f' '%f' '%f' '%f' '%f']);

starids = zeros(size(C{1}));
starlats = zeros(size(C{1}));
starlons = zeros(size(C{1}));

for i = 1:size(C{1},1)

    starids(i) = C{1}(i);

    rahr = C{2}(i);
    ramn = C{3}(i);
    rasc = C{4}(i);

    rasc = 15*rasc;

    carry = floor(rasc/60);
    rasc = rasc - 60*carry;

    ramn = 15*ramn + carry;

    carry = floor(ramn/60);
    ramn = ramn - 60*carry;

    radg = 15*rahr + carry;

    starlons(i) = dms2degrees([radg ramn rasc]);

    dcdg = C{5}(i);
    dcmn = C{6}(i);
    dcsc = C{7}(i);
    starlats(i) = dms2degrees([dcdg dcmn dcsc]);
end

fileID = fopen('bright_stars_simbad_trim.csv','r');
C = textscan(fileID,['%f' '%f' '%f' '%f' '%f' '%f']);
```

```
brightlats = zeros(size(C{1}));
brightlons = zeros(size(C{1}));

for i = 1:size(C{1},1)

    rahr = C{1}(i);
    ramn = C{2}(i);
    rasc = C{3}(i);

    rasc = 15*rasc;

    carry = floor(rasc/60);
    rasc = rasc - 60*carry;

    ramn = 15*ramn + carry;

    carry = floor(ramn/60);
    ramn = ramn - 60*carry;

    radg = 15*rahr + carry;

    brightlons(i) = dms2degrees([radg ramn rasc]);

    dcdg = C{4}(i);
    dcmn = C{5}(i);
    dcsc = C{6}(i);
    brightlats(i) = dms2degrees([dcdg dcmn dcsc]);
end

% Hubble Deep Field (north), HDF South, HU(X)DF/Chandra South
deeplons = [189.2058,338.2343,53.1625];
deeplats = [62.2161,-60.5507,-27.7914];

[brightlatmat,starlatmat] = ndgrid(brightlats,starlats);
[brightlonmat,starlonmat] = ndgrid(brightlons,starlons);
[arclens,azs] = distance(brightlatmat,brightlonmat,starlatmat,starlonmat);
[closest_to_each_bright,idx_to_each_bright] = min(arclens,[],2);

fileID = fopen('LUVOIR-Architecture_A-NOMINAL_OCCRATES-observations-trim.csv','r');
% HIP,Visit #,Visit dt (years),Exp Time (days)
C = textscan(fileID,'%d %d %f %f','Delimiter',',');

obs_ids = C{1}; % which star is being visited
obs_cts = C{2}; % which visit number to this star
```

```
obs_dts = C{3}; % how many years since the first visit to the star
obs_dur = C{4}; % how many days in that observation

%%

% axesm ('globe','Grid', 'on');
% view(60,60)
% axis off
figureMap = figure;
axesm('MapProjection','robinson','Grid','on','GLineWidth',2)
% axesm('MapProjection','stereo','MapLatLimit',[-83 -90],'PLineLocation',1,'
    ParallelLabel','on','Grid','on','GLineWidth',2)
p1 = scatterm(starlats,starlons,'*', 'linewidth', 2,'DisplayName','Stark 2015 targets
    ');
p2 = scatterm(deeplats,deeplons,'rv', 'linewidth', 2);
p3 = scatterm(brightlats,brightlons,'g+', 'linewidth', 2);
legend([p1 p3 p2],{'Stark 2015 targets','Magnitude 2 stars','Hubble/Chandra deep
    fields'})
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figureMap,'SkyMap_hires.png')

% figureRA = figure;
% plot(starids,(starlons>-5.183)&(starlons<-4.711),starids,(starlons>-3.404)&(
    starlons<-3.019),...
%     starids,(starlons>4.515)&(starlons<4.982))


%%
figureMinSeps = figure;
plot(closest_to_each_bright)
title('Closest target star to each bright star')
xlabel('Bright star index value (arb.)')
ylabel('Angular distance to nearest target star (deg.)')
```

B.5.2 DRM_prop_options

Calculates the number of LGS spacecraft required to support a mission, evaluating the different propulsion system options considered in LGSmain.

```
close all

% Eventual todo: use this script to pick which propulsion system goes
% into the "nom" values.
```

```
exp_time = mean(obs_dur)*daysec; % Average time from Chris's schedule
exp_acc = max_bg_acc; % From OrbitCalcs2dome3 calculations

max_simult_lgs = 100; % Let's say we want as many as 30 of these things active at
    once...

lgs_req = zeros(size(sc_fuel));
domains_req = zeros(size(sc_fuel));
time_req = zeros(size(sc_fuel));

for i = 1:numel(sc_fuel)
    [lgs_req_temp,domains_req_temp,time_req_temp] = DRMfunc(num_stars,total_obs,
    range_LGS,sc_mass,sc_fuel(i),sc_isp(i),sc_max_thrust(i),total_mission_time,
    exp_time,exp_acc,max_simult_lgs);
    lgs_req(i) = lgs_req_temp;
    domains_req(i) = domains_req_temp;
    time_req(i) = time_req_temp;
end


lgs_req_opt = zeros(size(sc_fuel));
domains_req_opt = zeros(size(sc_fuel));
time_req_opt = zeros(size(sc_fuel));

for i = 1:numel(sc_fuel)
    [lgs_req_temp,domains_req_temp,time_req_temp] = DRMfunc(num_stars,total_obs,
    range_LGS,sc_mass_opt+sc_prop_dry(i)+sc_fuel(i),sc_fuel(i),sc_isp(i),
    sc_max_thrust(i),total_mission_time,exp_time,exp_acc,max_simult_lgs);
    lgs_req_opt(i) = lgs_req_temp;
    domains_req_opt(i) = domains_req_temp;
    time_req_opt(i) = time_req_temp;
end

prop_labels = categorical(prop_names);

figBar = figure;
bar(prop_labels,[lgs_req;lgs_req_opt]','linewidth',2)
legend('24 kg','11.5 kg + prop','Location','northwest')
title(sprintf('Propulsion trade, %.2g,000 km range',range_LGS/1e6))
ylabel('Min. LGSs required')
set(gca, 'fontsize', 14,'linewidth',2)

%%
saveas(figBar,sprintf('DRM_prop_options_%.2gk.png',range_LGS/1e6))
ylim([0 60])
```

```
saveas(figBar,sprintf('DRM_prop_options_%.2gk_zoom.png',range_LGS/1e6))
```

B.5.3 DRM_sensitivity

Conducts sensitivity analyses of the design reference mission with respect to various parameters (range to the telescope, total spacecraft mass, etc.).

```
close all


% Jim's first DRM
% num_stars = 350;
% total_obs = 450; % 350 + revisit top 100

% Jim's second DRM (actually the first computed using this code) based on hearsay
    (350 stars, 1000 observations)
% num_stars = 350;
% total_obs = num_pts*2.8; % 2 observations for all targets, plus 8 more (10 total)
    for top 10%


exp_time = mean(obs_dur)*daysec; % Average time from Chris's schedule
exp_acc = max_bg_acc; % From OrbitCalcs2dome3 calculations

max_simult_lgs = 60; % Let's say we want as many as 30 of these things active at once
    ...

[lgs_req_std,domains_req_std,time_req_std] = DRMfunc(num_stars,total_obs,range_LGS,
    sc_mass,sc_fuel_nom,sc_isp_nom,sc_max_thrust_nom,total_mission_time,exp_time,
    exp_acc,max_simult_lgs);

[lgs_req_opt,domains_req_opt,time_req_opt] = DRMfunc(num_stars,total_obs,range_LGS,
    sc_mass_opt_tot,sc_fuel_nom,sc_isp_nom,sc_max_thrust_nom,total_mission_time,
    exp_time,exp_acc,max_simult_lgs);


sc_masses = 12:0.1:30;
lgs_req_mass = zeros(size(sc_masses));
domains_req_mass = zeros(size(sc_masses));
time_req_mass = zeros(size(sc_masses));

for i = 1:numel(sc_masses)
    [lgs_req_temp,domains_req_temp,time_req_temp] = DRMfunc(num_stars,total_obs,
    range_LGS,sc_masses(i),sc_fuel_nom,sc_isp_nom,sc_max_thrust_nom,
```

```
    total_mission_time,exp_time,exp_acc,max_simult_lgs);
    lgs_req_mass(i) = lgs_req_temp;
    domains_req_mass(i) = domains_req_temp;
    time_req_mass(i) = time_req_temp;
end

sc_fuels = 1.0:0.1:3.0;

lgs_req_fuel = zeros(size(sc_fuels));
domains_req_fuel = zeros(size(sc_fuels));
time_req_fuel = zeros(size(sc_fuels));

for i = 1:numel(sc_fuels)
    [lgs_req_temp,domains_req_temp,time_req_temp] = DRMfunc(num_stars,total_obs,
    range_LGS,sc_mass_opt + sc_prop_dry_nom + sc_fuels(i),sc_fuels(i),sc_isp_nom,
    sc_max_thrust_nom,total_mission_time,exp_time,exp_acc,max_simult_lgs);
    lgs_req_fuel(i) = lgs_req_temp;
    domains_req_fuel(i) = domains_req_temp;
    time_req_fuel(i) = time_req_temp;
end

ranges_LGS = 10e6:1e6:100e6;

lgs_req_range = zeros(size(ranges_LGS));
domains_req_range = zeros(size(ranges_LGS));
time_req_range = zeros(size(ranges_LGS));

for i = 1:numel(ranges_LGS)
    [lgs_req_temp,domains_req_temp,time_req_temp] = DRMfunc(num_stars,total_obs,
    ranges_LGS(i),sc_mass_opt_tot,sc_fuel_nom,sc_isp_nom,sc_max_thrust_nom,
    total_mission_time,exp_time,exp_acc,max_simult_lgs);
    lgs_req_range(i) = lgs_req_temp;
    domains_req_range(i) = domains_req_temp;
    time_req_range(i) = time_req_temp;
end

max_simult_lgses = 1:30;

lgs_req_simult = zeros(size(max_simult_lgses));
domains_req_simult = zeros(size(max_simult_lgses));
time_req_simult = zeros(size(max_simult_lgses));

for i = 1:numel(max_simult_lgses)
    [lgs_req_temp,domains_req_temp,time_req_temp] = DRMfunc(num_stars,total_obs,
    range_LGS,sc_mass_opt_tot,sc_fuel_nom,sc_isp_nom,sc_max_thrust_nom,
```

```matlab
    total_mission_time,exp_time,exp_acc,max_simult_lgses(i));
    lgs_req_simult(i) = lgs_req_temp;
    domains_req_simult(i) = domains_req_temp;
    time_req_simult(i) = time_req_temp;
end

% Looking at increasing the number of stars, observations, and mission
% times...getting some weird results here.  Going from 19 needed for stock
% LUVOIR to 43 needed for observing twice as many stars (and twice as many
% observations) in the same time is reasonable, but it's saying we can get
% that done in 1.9 years when we needed 3.5 for stock LUVOIR is a little
% weird...something for future work.

nums_stars = [num_stars, num_stars, num_stars*2, num_stars*2];
totals_obs = [total_obs, total_obs*2, total_obs*2, total_obs*2];
total_mission_times = [total_mission_time, total_mission_time*2, total_mission_time,
    total_mission_time*2];

new_missions_titles = categorical({'1x stars, 1x obs, 1x dur (stock)','1x stars, 2x
    obs, 2x dur','2x stars, 2x obs, 1x dur','2x stars, 2x obs, 2x dur'});

lgs_req_nstars = zeros(size(nums_stars));
domains_req_nstars = zeros(size(nums_stars));
time_req_nstars = zeros(size(nums_stars));

for i = 1:numel(nums_stars)
    [lgs_req_temp,domains_req_temp,time_req_temp] = DRMfunc(nums_stars(i),totals_obs(
    i),range_LGS,sc_mass_opt_tot,sc_fuel_nom,sc_isp_nom,sc_max_thrust_nom,
    total_mission_time,exp_time,exp_acc,max_simult_lgs);
    lgs_req_nstars(i) = lgs_req_temp;
    domains_req_nstars(i) = domains_req_temp;
    time_req_nstars(i) = time_req_temp;
end

figuremass = figure;
plot(sc_masses,lgs_req_mass,sc_mass_opt_tot,lgs_req_opt,'r*','linewidth',2)
legend('Total mass trade','Baseline case')
title('Minimum LGS required vs. LGS mass')
xlabel('LGS mass (kg)')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figuremass,sprintf('DRM-sensit-mass-%.2gk.png',range_LGS/1e6))

figurefuel = figure;
plot(sc_fuels,lgs_req_fuel,sc_fuel_nom,lgs_req_opt,'r*','linewidth',2)
legend('Fuel mass trade','Baseline case')
```

```
title('Minimum LGS required vs. LGS fuel mass')
xlabel('LGS fuel mass (kg)')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figurefuel,sprintf('DRM-sensit-fuel-%.2gk.png',range_LGS/1e6))

figurerange = figure;
plot(ranges_LGS/1e6,lgs_req_range,range_LGS/1e6,lgs_req_opt,'r*','linewidth',2)
legend('LGS range trade','Baseline case')
title('Minimum LGS required vs. Telescope-LGS range')
xlabel('Telescope-LGS range (1000''s km)')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figurerange,sprintf('DRM-sensit-range-%.2gk.png',range_LGS/1e6))

figuresimult = figure;
plot(max_simult_lgses,lgs_req_simult,domains_req_opt,lgs_req_opt,'r*','linewidth',2)
legend('Max active trade','Baseline case')
title('Minimum LGS required vs. Max LGS simult.')
xlabel('Maximum LGS simultaneously active')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figuresimult,sprintf('DRM-sensit-domains-%.2gk.png',range_LGS/1e6))

figuresimulttime = figure;
plot(max_simult_lgses,time_req_simult/yrsec,domains_req_opt,time_req_opt/yrsec,'r*',
    max_simult_lgses,5*ones(size(max_simult_lgses)),'linewidth',2)
legend('Max active trade','Baseline case','5-year limit')
ylim([0 17])
title('Time required for campaign vs. Max LGS simult.')
xlabel('Maximum LGS simultaneously active')
ylabel('Years to execute survey campaign')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figuresimulttime,sprintf('DRM-sensit-domains-time-%.2gk.png',range_LGS/1e6))

figurenstars = figure;
bar(new_missions_titles,lgs_req_nstars)
```

B.5.4 DRMfunc

The main function for evaluating a design reference mission, invoked by `DRM_prop_options` and `DRM_sensitivity`.

```
close all


% Jim's first DRM
```

```
% num_stars = 350;
% total_obs = 450; % 350 + revisit top 100

% Jim's second DRM (actually the first computed using this code) based on hearsay
    (350 stars, 1000 observations)
% num_stars = 350;
% total_obs = num_pts*2.8; % 2 observations for all targets, plus 8 more (10 total)
    for top 10%


exp_time = mean(obs_dur)*daysec; % Average time from Chris's schedule
exp_acc = max_bg_acc; % From OrbitCalcs2dome3 calculations

max_simult_lgs = 60; % Let's say we want as many as 30 of these things active at once
    ...

[lgs_req_std,domains_req_std,time_req_std] = DRMfunc(num_stars,total_obs,range_LGS,
    sc_mass,sc_fuel_nom,sc_isp_nom,sc_max_thrust_nom,total_mission_time,exp_time,
    exp_acc,max_simult_lgs);

[lgs_req_opt,domains_req_opt,time_req_opt] = DRMfunc(num_stars,total_obs,range_LGS,
    sc_mass_opt_tot,sc_fuel_nom,sc_isp_nom,sc_max_thrust_nom,total_mission_time,
    exp_time,exp_acc,max_simult_lgs);


sc_masses = 12:0.1:30;
lgs_req_mass = zeros(size(sc_masses));
domains_req_mass = zeros(size(sc_masses));
time_req_mass = zeros(size(sc_masses));

for i = 1:numel(sc_masses)
    [lgs_req_temp,domains_req_temp,time_req_temp] = DRMfunc(num_stars,total_obs,
    range_LGS,sc_masses(i),sc_fuel_nom,sc_isp_nom,sc_max_thrust_nom,
    total_mission_time,exp_time,exp_acc,max_simult_lgs);
    lgs_req_mass(i) = lgs_req_temp;
    domains_req_mass(i) = domains_req_temp;
    time_req_mass(i) = time_req_temp;
end

sc_fuels = 1.0:0.1:3.0;

lgs_req_fuel = zeros(size(sc_fuels));
domains_req_fuel = zeros(size(sc_fuels));
time_req_fuel = zeros(size(sc_fuels));
```

```
for i = 1:numel(sc_fuels)
    [lgs_req_temp,domains_req_temp,time_req_temp] = DRMfunc(num_stars,total_obs,
    range_LGS,sc_mass_opt + sc_prop_dry_nom + sc_fuels(i),sc_fuels(i),sc_isp_nom,
    sc_max_thrust_nom,total_mission_time,exp_time,exp_acc,max_simult_lgs);
    lgs_req_fuel(i) = lgs_req_temp;
    domains_req_fuel(i) = domains_req_temp;
    time_req_fuel(i) = time_req_temp;
end

ranges_LGS = 10e6:1e6:100e6;

lgs_req_range = zeros(size(ranges_LGS));
domains_req_range = zeros(size(ranges_LGS));
time_req_range = zeros(size(ranges_LGS));

for i = 1:numel(ranges_LGS)
    [lgs_req_temp,domains_req_temp,time_req_temp] = DRMfunc(num_stars,total_obs,
    ranges_LGS(i),sc_mass_opt_tot,sc_fuel_nom,sc_isp_nom,sc_max_thrust_nom,
    total_mission_time,exp_time,exp_acc,max_simult_lgs);
    lgs_req_range(i) = lgs_req_temp;
    domains_req_range(i) = domains_req_temp;
    time_req_range(i) = time_req_temp;
end

max_simult_lgses = 1:30;

lgs_req_simult = zeros(size(max_simult_lgses));
domains_req_simult = zeros(size(max_simult_lgses));
time_req_simult = zeros(size(max_simult_lgses));

for i = 1:numel(max_simult_lgses)
    [lgs_req_temp,domains_req_temp,time_req_temp] = DRMfunc(num_stars,total_obs,
    range_LGS,sc_mass_opt_tot,sc_fuel_nom,sc_isp_nom,sc_max_thrust_nom,
    total_mission_time,exp_time,exp_acc,max_simult_lgses(i));
    lgs_req_simult(i) = lgs_req_temp;
    domains_req_simult(i) = domains_req_temp;
    time_req_simult(i) = time_req_temp;
end

% Looking at increasing the number of stars, observations, and mission
% times...getting some weird results here.  Going from 19 needed for stock
% LUVOIR to 43 needed for observing twice as many stars (and twice as many
% observations) in the same time is reasonable, but it's saying we can get
% that done in 1.9 years when we needed 3.5 for stock LUVOIR is a little
% weird...something for future work.
```

```
nums_stars = [num_stars, num_stars, num_stars*2, num_stars*2];
totals_obs = [total_obs, total_obs*2, total_obs*2, total_obs*2];
total_mission_times = [total_mission_time, total_mission_time*2, total_mission_time,
    total_mission_time*2];

new_missions_titles = categorical({'1x stars, 1x obs, 1x dur (stock)','1x stars, 2x
    obs, 2x dur','2x stars, 2x obs, 1x dur','2x stars, 2x obs, 2x dur'});

lgs_req_nstars = zeros(size(nums_stars));
domains_req_nstars = zeros(size(nums_stars));
time_req_nstars = zeros(size(nums_stars));

for i = 1:numel(nums_stars)
    [lgs_req_temp,domains_req_temp,time_req_temp] = DRMfunc(nums_stars(i),totals_obs(
    i),range_LGS,sc_mass_opt_tot,sc_fuel_nom,sc_isp_nom,sc_max_thrust_nom,
    total_mission_time,exp_time,exp_acc,max_simult_lgs);
    lgs_req_nstars(i) = lgs_req_temp;
    domains_req_nstars(i) = domains_req_temp;
    time_req_nstars(i) = time_req_temp;
end

figuremass = figure;
plot(sc_masses,lgs_req_mass,sc_mass_opt_tot,lgs_req_opt,'r*','linewidth',2)
legend('Total mass trade','Baseline case')
title('Minimum LGS required vs. LGS mass')
xlabel('LGS mass (kg)')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figuremass,sprintf('DRM-sensit-mass-%.2gk.png',range_LGS/1e6))

figurefuel = figure;
plot(sc_fuels,lgs_req_fuel,sc_fuel_nom,lgs_req_opt,'r*','linewidth',2)
legend('Fuel mass trade','Baseline case')
title('Minimum LGS required vs. LGS fuel mass')
xlabel('LGS fuel mass (kg)')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figurefuel,sprintf('DRM-sensit-fuel-%.2gk.png',range_LGS/1e6))

figurerange = figure;
plot(ranges_LGS/1e6,lgs_req_range,range_LGS/1e6,lgs_req_opt,'r*','linewidth',2)
legend('LGS range trade','Baseline case')
title('Minimum LGS required vs. Telescope-LGS range')
xlabel('Telescope-LGS range (1000''s km)')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figurerange,sprintf('DRM-sensit-range-%.2gk.png',range_LGS/1e6))
```

```
figuresimult = figure;
plot(max_simult_lgses,lgs_req_simult,domains_req_opt,lgs_req_opt,'r*','linewidth',2)
legend('Max active trade','Baseline case')
title('Minimum LGS required vs. Max LGS simult.')
xlabel('Maximum LGS simultaneously active')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figuresimult,sprintf('DRM-sensit-domains-%.2gk.png',range_LGS/1e6))

figuresimulttime = figure;
plot(max_simult_lgses,time_req_simult/yrsec,domains_req_opt,time_req_opt/yrsec,'r*',
    max_simult_lgses,5*ones(size(max_simult_lgses)),'linewidth',2)
legend('Max active trade','Baseline case','5-year limit')
ylim([0 17])
title('Time required for campaign vs. Max LGS simult.')
xlabel('Maximum LGS simultaneously active')
ylabel('Years to execute survey campaign')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figuresimulttime,sprintf('DRM-sensit-domains-time-%.2gk.png',range_LGS/1e6))

figurenstars = figure;
bar(new_missions_titles,lgs_req_nstars)
```

B.5.5 StarkSkymap_TSP

Uses MATLAB's `intlinprog` solver to solve the Traveling Salesman Problem for the list of star targets. This program takes longer to run than all the rest put together, so it saves its result for later reuse.

```
% Run StarkSkymap first to initialize starlats and starlons
close all;

nstars = numel(starlats);

sc_max_acc = sc_max_thrust_nom/sc_mass_opt_tot;

idxs = nchoosek(1:nstars,2);

[arclens az] = distance(starlats(idxs(:,1)),starlons(idxs(:,1)),...
    starlats(idxs(:,2)),starlons(idxs(:,2)));
arcrads = deg2rad(arclens);

costs = 2*sqrt(range_LGS.*sc_max_acc.*arcrads);
```

```
lendist = length(costs);

G = graph(idxs(:,1),idxs(:,2));

figureChart = figure;
hGraph = plot(G,'XData',starlons-360.*(starlons>180),'YData',starlats,'LineStyle','
    none','NodeLabel',{});

figureMap = figure;
axesm('MapProjection','robinson','Grid','on','GLineWidth',2)
p1 = scatterm(starlats,starlons,'*', 'linewidth', 2,'DisplayName','Stark 2015 targets
    ');

%%

Aeq = spalloc(nstars,length(idxs),nstars*(nstars-1)); % Allocate a sparse matrix
for ii = 1:nstars
    whichIdxs = (idxs == ii); % Find the trips that include stop ii
    whichIdxs = sparse(sum(whichIdxs,2)); % Include trips where ii is at either end
    Aeq(ii,:) = whichIdxs'; % Include in the constraint matrix
end
beq = 2*ones(nstars,1);

intcon = 1:lendist;
lb = zeros(lendist,1);
ub = ones(lendist,1);
%%
opts = optimoptions('intlinprog');
[x_tsp,costopt,exitflag,output] = intlinprog(costs,intcon,[],[],Aeq,beq,lb,ub,opts);

x_tsp = logical(round(x_tsp));
Gsol = graph(idxs(x_tsp,1),idxs(x_tsp,2));

%%
highlight(hGraph,Gsol,'LineStyle','-')

tourIdxs = conncomp(Gsol);
numtours = max(tourIdxs); % number of subtours
fprintf('# of subtours: %d\n',numtours);

%%

A = spalloc(0,lendist,0); % Allocate a sparse linear inequality constraint matrix
b = [];
```

```
while numtours > 1 % Repeat until there is just one subtour
    % Add the subtour constraints
    b = [b;zeros(numtours,1)]; % allocate b
    A = [A;spalloc(numtours,lendist,nstars)]; % A guess at how many nonzeros to
    allocate
    for ii = 1:numtours
        rowIdx = size(A,1) + 1; % Counter for indexing
        subTourIdx = find(tourIdxs == ii); % Extract the current subtour
%          The next lines find all of the variables associated with the
%          particular subtour, then add an inequality constraint to prohibit
%          that subtour and all subtours that use those stops.
        variations = nchoosek(1:length(subTourIdx),2);
        for jj = 1:length(variations)
            whichVar = (sum(idxs==subTourIdx(variations(jj,1)),2)) & ...
                       (sum(idxs==subTourIdx(variations(jj,2)),2));
            A(rowIdx,whichVar) = 1;
        end
        b(rowIdx) = length(subTourIdx) - 1; % One less trip than subtour stops
    end

    % Try to optimize again
    [x_tsp,costopt,exitflag,output] = intlinprog(costs,intcon,A,b,Aeq,beq,lb,ub,opts)
    ;
    x_tsp = logical(round(x_tsp));
    Gsol = graph(idxs(x_tsp,1),idxs(x_tsp,2));

    % Visualize result
    hGraph.LineStyle = 'none'; % Remove the previous highlighted path
    highlight(hGraph,Gsol,'LineStyle','-')
    drawnow

    % How many subtours this time?
    tourIdxs = conncomp(Gsol);
    numtours = max(tourIdxs); % number of subtours
    fprintf('# of subtours: %d\n',numtours)
end

%%

save('stark_skymap_tsp.mat','Gsol')

%%

dists = 1:numel(Gsol.Edges);
```

```
for i = 1:numel(Gsol.Edges)
    dists(i) = distance(starlats(Gsol.Edges{i,1}(1)),starlons(Gsol.Edges{i,1}(1)),...
        starlats(Gsol.Edges{i,1}(2)),starlons(Gsol.Edges{i,1}(2)));
end

dists = deg2rad(dists);

times = 2*sqrt(range_LGS.*dists./sc_max_acc);

days = times/86400;

dvs = times.*sc_max_acc;

%%


figureGlobe = figure;
axesm('globe','Grid','on','GLineWidth',2,'MeridianLabel','on','MLabelParallel','
    equator','ParallelLabel','on','PLabelMeridian','prime')
p1 = scatterm(starlats,starlons,'*', 'linewidth', 2,'DisplayName','Stark 2015 targets
    ');
p2 = scatterm(deeplats,deeplons,'rv', 'linewidth', 2);
p3 = scatterm(brightlats,brightlons,'g+', 'linewidth', 2);
% legend([p1 p3 p2],{'Stark 2015 targets','Magnitude 2 stars','Hubble/Chandra deep
    fields'})
set(gca, 'fontsize', 14,'linewidth',2)

for i = 1:numel(Gsol.Edges)
    plot3m([starlats(Gsol.Edges{i,1}(1)) starlats(Gsol.Edges{i,1}(2))],...
        [starlons(Gsol.Edges{i,1}(1)) starlons(Gsol.Edges{i,1}(2))],...
        [0.02 0.02],'k-','LineWidth',2);
end

base = zeros(180,360);
baseR = georefcells([-90 90],[0 360],size(base));
copperColor = [0.62 0.38 0.24];
geoshow(base,baseR,'FaceColor',copperColor)
camlight right
material([.8 .9 .4])
saveas(figureGlobe,'skyglobe.png')

%%
figureBar = figure;
bar(days)
```

B.5.6 `ham_StarkSkymap`

Uses Pramit Biswas's Hamiltonian code [8] to process the graph produced by `StarkSkymap_TSP` into a Hamiltonian route, *i.e.* a list of stars visited in order.

```
hamStark = hamiltonian(adjacency(Gsol),1,1);
```

B.5.7 `StarkSchedule`

A simple greedy scheduler, for assigning observations to LGS spacecraft. This code rigidly adheres to the schedule of observations in order of scientific priority (*i.e.* expected exo-Earth yield).

```
% Ingests the list of observations from Stark, verbatim, rigidly fits them
% into a schedule of evenly-spaced observations over four years, and
% schedules LGSs accordingly.  Levels out around 34 LGSs required, should
% only need 20 or so.

% start by running StarkSkymap

close all;

obs_asgn = zeros(size(obs_cts));

lgs_count = 1;
lgs_dv = dvcap;

lgs_dvs = dvcap;

speed_factor = 0.23; % reduce this to force the LGS to transit more slowly than max
    speed
% Empirically, 0.23 seems to have the best results.

sc_max_acc = (sc_max_thrust_nom/sc_mass_opt_tot)*speed_factor;

while sum(obs_asgn==0) > 0

    unasgn = find(obs_asgn==0);

%      disp(numel(unasgn))
%      disp(lgs_count)

    curr_idx = unasgn(1);
```

```
exp_dv = max_bg_acc*daysec*obs_dur(curr_idx);

if(exp_dv > lgs_dv)
    lgs_dvs(lgs_count) = lgs_dv;
    lgs_count = lgs_count + 1;
    lgs_dv = dvcap;
end

lgs_dv = lgs_dv - exp_dv;
obs_asgn(curr_idx) = lgs_count;

curr_starid = obs_ids(curr_idx);
curr_starlat = starlats(find(starids==curr_starid));
curr_starlon = starlons(find(starids==curr_starid));

next_obs = 2;

while next_obs <= numel(unasgn)
    test_idx = unasgn(next_obs);
    test_starid = obs_ids(test_idx);

    test_starlat = starlats(find(starids==test_starid));
    test_starlon = starlons(find(starids==test_starid));

    test_dist = deg2rad(distance(curr_starlat,curr_starlon,test_starlat,
test_starlon));

    transit_time_req = ((test_idx-curr_idx)*total_mission_time/total_obs)-daysec*
obs_dur(curr_idx);
    transit_time_actual = 2*sqrt(range_LGS*test_dist/sc_max_acc);
    transit_dv = transit_time_actual*sc_max_acc + max_bg_acc*daysec*obs_dur(
curr_idx);

    if (transit_time_actual < transit_time_req) && (transit_dv < lgs_dv)
        curr_idx = test_idx;
        lgs_dv = lgs_dv - transit_dv;
        obs_asgn(curr_idx) = lgs_count;

        curr_starid = obs_ids(curr_idx);
        curr_starlat = starlats(find(starids==curr_starid));
        curr_starlon = starlons(find(starids==curr_starid));
    end
    next_obs = next_obs + 1;
end
```

```
    if sum(obs_asgn==0) > 0
        lgs_dvs(lgs_count) = lgs_dv;
        lgs_count = lgs_count + 1;
        lgs_dv = dvcap;
    end

end

lgs_dvs(lgs_count) = lgs_dv;

num_sats = max(obs_asgn);

disp(num_sats)

C_obs = cell(num_sats,3);

obs_per_sat = zeros(size(1:num_sats));

for i = 1:num_sats
    obs_per_sat(i) = sum(obs_asgn==i);

    obs_made = obs_ids(find(obs_asgn == i));
    obs_idxs = zeros(size(obs_made));
    for j = 1:obs_per_sat(i)
        obs_idxs(j) = find(starids == obs_made(j));
    end
    obs_lats = starlats(obs_idxs);
    obs_lons = starlons(obs_idxs);
    C_obs{i,1} = obs_idxs;
    C_obs{i,2} = obs_lats;
    C_obs{i,3} = obs_lons;
end

figOPS = figure;
plot(obs_per_sat,'linewidth',2)
title('Nr. obs. supported by each LGS satellite')
xlabel('LGS satellite number')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figOPS,'StarkSchedule_rigid_obs_per_sat.png')

figCloud = figure;
plot(obs_asgn,'x','linewidth',2)
title('Observations supported by each LGS satellite')
```

```
xlabel('Observation number')
ylabel('LGS satellite number')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figCloud,'StarkSchedule_rigid_obs_cloud.png')

% figure;
% hold on
% plot(lgs_times/daysec)
% plot([0 num_sats],[total_mission_time/daysec total_mission_time/daysec])
% hold off
% title('Days of engagement by LGS spacecraft')
% legend('LGS operation time','Max mission duration')

figDV = figure;
hold on
plot(lgs_dvs,'linewidth',2)
plot([0 num_sats],[dvcap dvcap],'linewidth',2)
hold off
title('dV remaining in each LGS spacecraft')
legend('LGS dV remaining','Initial dV capacity')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figDV,'StarkSchedule_rigid_dv_remaining.png')


colors_for_plot = lines(num_sats);

figureMap = figure;
axesm('MapProjection','robinson','Grid','on','GLineWidth',2)
% axesm('MapProjection','stereo','MapLatLimit',[-83 -90],'PLineLocation',1,'
    ParallelLabel','on','Grid','on','GLineWidth',2)
p1 = scatterm(starlats,starlons,'*', 'linewidth', 2);
p2 = scatterm(deeplats,deeplons,'rv', 'linewidth', 2);
p3 = scatterm(brightlats,brightlons,'g+', 'linewidth', 2);
% legend([p1 p3 p2],{'Stark 2015 targets','Magnitude 2 stars','Hubble/Chandra deep
    fields'})
for i = [1 10 floor(num_sats/7)*7]
    plotm(C_obs{i,2},C_obs{i,3},'Color',colors_for_plot(i,:), 'linewidth', 2)

end

set(gca, 'fontsize', 14,'linewidth',2)
saveas(figureMap,'StarkSchedule_rigid_map.png')
```

B.5.8 StarkScheduleAltB

An attempted evolution of StarkSchedule that only assigns 'nearby' observations to LGS spacecraft, to keep their movements segmented. Unfortunately it doesn't yet ensure that the segments are butted against each other, so it actually performs worse (*i.e.* assigns more LGS spacecraft) than StarkSchedule.

```
% Ingests the list of observations from Stark, verbatim, and allows
% satellites to run a little over time as long as the total schedule is
% respected.

% start by running StarkSkymap

close all

obs_asgn = zeros(size(obs_cts));

lgs_count = 1;
lgs_dv = dvcap;
lgs_time = 0;

lgs_dvs = dvcap;
lgs_times = 0;

speed_factor = 0.18; % reduce this to force the LGS to transit more slowly than max
    speed
% Empirically, X seems to have the best results.

sc_max_acc = (sc_max_thrust_nom/sc_mass_opt_tot)*speed_factor;

while sum(obs_asgn==0) > 0

    unasgn = find(obs_asgn==0);

%      disp(numel(unasgn))
%      disp(lgs_count)

    curr_idx = unasgn(1);

    exp_time = daysec*obs_dur(curr_idx);
    exp_dv = avg_acc*exp_time;

    if (exp_dv > lgs_dv) || (lgs_time + exp_time > total_mission_time)
        lgs_dvs(lgs_count) = lgs_dv;
```

```
        lgs_times(lgs_count) = lgs_time;
        lgs_count = lgs_count + 1;
        lgs_dv = dvcap;
        lgs_time = 0;
    end

    lgs_dv = lgs_dv - exp_dv;
    lgs_time = lgs_time + exp_time;
    obs_asgn(curr_idx) = lgs_count;

    curr_starid = obs_ids(curr_idx);
    curr_starlat = starlats(find(starids==curr_starid));
    curr_starlon = starlons(find(starids==curr_starid));

    next_obs = 2;

    while next_obs <= numel(unasgn)
        test_idx = unasgn(next_obs);
        test_starid = obs_ids(test_idx);

        test_starlat = starlats(find(starids==test_starid));
        test_starlon = starlons(find(starids==test_starid));

        test_dist = deg2rad(distance(curr_starlat,curr_starlon,test_starlat,
    test_starlon));

        transit_time_req = 1.5*((test_idx-curr_idx)*total_mission_time/total_obs)-
    daysec*obs_dur(curr_idx);
        transit_time_actual = 2*sqrt(range_LGS*test_dist/sc_max_acc) + daysec*obs_dur
    (test_idx);
        transit_dv = transit_time_actual*sc_max_acc +  avg_acc*daysec*obs_dur(
    test_idx);

        my_visits = find(obs_asgn == lgs_count);

        first_idx = my_visits(1);
        first_starid = obs_ids(first_idx);

        first_starlat = starlats(find(starids==first_starid));
        first_starlon = starlons(find(starids==first_starid));

        test_dist_from_home = deg2rad(distance(first_starlat,first_starlon,
    test_starlat,test_starlon));

        if (transit_dv < lgs_dv) && (lgs_time + transit_time_actual <
```

```
    total_mission_time)  && (test_dist > 0) && (transit_time_actual <
    transit_time_req) && (test_dist_from_home < deg2rad(60))
            curr_idx = test_idx;
            lgs_dv = lgs_dv - transit_dv;
            lgs_time = lgs_time + transit_time_actual;
            obs_asgn(curr_idx) = lgs_count;

            exp_time = daysec*obs_dur(curr_idx);
            exp_dv = avg_acc*exp_time;
            lgs_dv = lgs_dv - exp_dv;
            lgs_time = lgs_time + exp_time;

            curr_starid = obs_ids(curr_idx);
            curr_starlat = starlats(find(starids==curr_starid));
            curr_starlon = starlons(find(starids==curr_starid));
        end
        next_obs = next_obs + 1;
    end


    if sum(obs_asgn==0) > 0

        lgs_dvs(lgs_count) = lgs_dv;
        lgs_times(lgs_count) = lgs_time;

        lgs_count = lgs_count + 1;
        lgs_dv = dvcap;
        lgs_time = 0;
    end

end

lgs_dvs(lgs_count) = lgs_dv;
lgs_times(lgs_count) = lgs_time;

num_sats = max(obs_asgn);

disp(num_sats)

C_obs = cell(num_sats,3);

obs_per_sat = zeros(size(1:num_sats));

for i = 1:num_sats
    obs_per_sat(i) = sum(obs_asgn==i);
```

```
    obs_made = obs_ids(find(obs_asgn == i));
    obs_idxs = zeros(size(obs_made));
    for j = 1:obs_per_sat(i)
        obs_idxs(j) = find(starids == obs_made(j));
    end
    obs_lats = starlats(obs_idxs);
    obs_lons = starlons(obs_idxs);
    C_obs{i,1} = obs_idxs;
    C_obs{i,2} = obs_lats;
    C_obs{i,3} = obs_lons;
end

figOPS = figure;
plot(obs_per_sat,'linewidth',2)
title('Nr. obs. supported by each LGS satellite')
xlabel('LGS satellite number')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figOPS,'StarkSchedule_flex_obs_per_sat.png')

figCloud = figure;
plot(obs_asgn,'x','linewidth',2)
title('Observations supported by each LGS satellite')
xlabel('Observation number')
ylabel('LGS satellite number')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figCloud,'StarkSchedule_flex_obs_cloud.png')

figTime = figure;
hold on
plot(lgs_times/daysec,'linewidth',2)
plot([0 num_sats],[total_mission_time/daysec total_mission_time/daysec],'linewidth'
    ,2)
hold off
title('Days of engagement by LGS spacecraft')
legend('LGS operation time','Max mission duration')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figTime,'StarkSchedule_flex_time_remaining.png')

figDV = figure;
hold on
plot(lgs_dvs,'linewidth',2)
plot([0 num_sats],[dvcap dvcap],'linewidth',2)
hold off
title('dV remaining in each LGS spacecraft')
```

```
legend('LGS dV remaining','Initial dV capacity')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figDV,'StarkSchedule_flex_dv_remaining.png')



colors_for_plot = lines(num_sats);

figureMap = figure;
axesm('MapProjection','robinson','Grid','on','GLineWidth',2)
% axesm('MapProjection','stereo','MapLatLimit',[-83 -90],'PLineLocation',1,'
    ParallelLabel','on','Grid','on','GLineWidth',2)
p1 = scatterm(starlats,starlons,'*', 'linewidth', 2);
p2 = scatterm(deeplats,deeplons,'rv', 'linewidth', 2);
p3 = scatterm(brightlats,brightlons,'g+', 'linewidth', 2);
% legend([p1 p3 p2],{'Stark 2015 targets','Magnitude 2 stars','Hubble/Chandra deep
    fields'})
for i = [1 10 floor(num_sats/7)*7]
    plotm(C_obs{i,2},C_obs{i,3},'Color',colors_for_plot(i,:), 'linewidth', 2)

end

set(gca, 'fontsize', 14,'linewidth',2)
saveas(figureMap,'StarkSchedule_flex_map.png')
```

B.5.9 StarkScheduleAltD

A scheduler that ingests the TSP solution and segments it among LGS spacecraft.

```
% This one is based on the TSP single-salesman circuit -- each LGS will
% pick up stars until it can't anymore, and then hand off to the next.
% Trying to fit a full circuit into 1/6th the time and dV budget.

% Needs to run ham_StarkSkymap first.

close all

lgs_count = 1;
lgs_dv = dvcap/6;
lgs_time = 0;

lgs_dvs = dvcap/6;
lgs_times = 0;
```

```
exp_time = mean(obs_dur)*daysec; % Average time from Chris's schedule
exp_dv = avg_acc*exp_time;

obs_asgn = zeros(size(starids));

speed_factor = 0.19; % reduce this to force the LGS to transit more slowly than max
    speed
% Empirically works down to 13 sats required (yay!) at ~0.19

sc_max_acc = (sc_max_thrust_nom/sc_mass_opt_tot)*speed_factor;

for i = 1:numel(starids)


    if (exp_dv > lgs_dv) || (lgs_time + exp_time > total_mission_time/6)
        lgs_dvs(lgs_count) = lgs_dv;
        lgs_times(lgs_count) = lgs_time;
        lgs_count = lgs_count + 1;
        lgs_dv = dvcap/6;
        lgs_time = 0;
    end

    obs_asgn(i) = lgs_count;

    lgs_dv = lgs_dv - exp_dv;
    lgs_time = lgs_time + exp_time;

    curr_starid = starids(hamStark(i));
    curr_starlat = starlats(hamStark(i));
    curr_starlon = starlons(hamStark(i));

    next_starid = starids(hamStark(i+1));
    next_starlat = starlats(hamStark(i+1));
    next_starlon = starlons(hamStark(i+1));

    next_dist = deg2rad(distance(curr_starlat,curr_starlon,next_starlat,next_starlon)
    );

    transit_time = 2*sqrt(range_LGS*next_dist/sc_max_acc);
    transit_dv = transit_time*sc_max_acc;

    if (transit_dv < lgs_dv) && (lgs_time + transit_time < total_mission_time/6)
        lgs_dv = lgs_dv - transit_dv;
        lgs_time = lgs_time + transit_time;
    else
```

```
        lgs_dvs(lgs_count) = lgs_dv;
        lgs_times(lgs_count) = lgs_time;
        lgs_count = lgs_count + 1;
        lgs_dv = dvcap/6;
        lgs_time = 0;
    end
end

lgs_dvs(lgs_count) = lgs_dv;
lgs_times(lgs_count) = lgs_time;

num_sats = max(obs_asgn);

disp(num_sats)

C_obs = cell(num_sats,3);

obs_per_sat = zeros(size(1:num_sats));

for i = 1:num_sats
    obs_per_sat(i) = sum(obs_asgn==i);

    obs_made = starids(hamStark(find(obs_asgn == i)));

    obs_lats = starlats(hamStark(find(obs_asgn == i)));
    obs_lons = starlons(hamStark(find(obs_asgn == i)));
    C_obs{i,1} = obs_made;
    C_obs{i,2} = obs_lats;
    C_obs{i,3} = obs_lons;
end

figOPS = figure;
plot(6*obs_per_sat,'linewidth',2)
ylim([0 6*25])
title('Nr. obs. supported by each LGS satellite')
xlabel('LGS satellite number')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figOPS,'StarkSchedule_tsp_obs_per_sat.png')

figCloud = figure;
plot(obs_asgn,'x','linewidth',2)
title('Observations supported by each LGS satellite')
xlabel('Observation number')
ylabel('LGS satellite number')
set(gca, 'fontsize', 14,'linewidth',2)
```

```
saveas(figCloud,'StarkSchedule_tsp_obs_cloud.png')

figTime = figure;
hold on
plot(6*lgs_times/daysec,'linewidth',2)
plot([0 num_sats],[total_mission_time/(daysec) total_mission_time/(daysec)],'
    linewidth',2)
hold off
ylim([0 365])
title('Days of engagement by LGS spacecraft')
legend('LGS operation time','Max mission duration')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figTime,'StarkSchedule_tsp_time_remaining.png')

figDV = figure;
hold on
plot(lgs_dvs*6,'linewidth',2)
plot([0 num_sats],[dvcap dvcap],'linewidth',2)
hold off
title('dV remaining in each LGS spacecraft')
legend('LGS dV remaining','Initial dV capacity')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figDV,'StarkSchedule_tsp_dv_remaining.png')


colors_for_plot = lines(num_sats);

figureMap = figure;
axesm('MapProjection','robinson','Grid','on','GLineWidth',2)
% axesm('MapProjection','stereo','MapLatLimit',[-83 -90],'PLineLocation',1,'
    ParallelLabel','on','Grid','on','GLineWidth',2)
p1 = scatterm(starlats,starlons,'*', 'linewidth', 2);
p2 = scatterm(deeplats,deeplons,'rv', 'linewidth', 2);
p3 = scatterm(brightlats,brightlons,'g+', 'linewidth', 2);
% legend([p1 p3 p2],{'Stark 2015 targets','Magnitude 2 stars','Hubble/Chandra deep
    fields'})
for i = 9
    plotm(C_obs{i,2},C_obs{i,3},'Color',colors_for_plot(i,:), 'linewidth', 2)

end

set(gca, 'fontsize', 14,'linewidth',2)
saveas(figureMap,'StarkSchedule_tsp_map.png')

figureGlobe = figure;
```

```
axesm('globe','Grid','on','GLineWidth',2,'MeridianLabel','on','MLabelParallel','
    equator','ParallelLabel','on','PLabelMeridian','prime')
p1 = scatterm(starlats,starlons,'*', 'linewidth', 2);
p2 = scatterm(deeplats,deeplons,'rv', 'linewidth', 2);
p3 = scatterm(brightlats,brightlons,'g+', 'linewidth', 2);
% legend([p1 p3 p2],{'Stark 2015 targets','Magnitude 2 stars','Hubble/Chandra deep
    fields'})
set(gca, 'fontsize', 14,'linewidth',2)

for i = 1:num_sats
    plot3m(C_obs{i,2},C_obs{i,3},0.02*ones(size(C_obs{i,1})),'Color',colors_for_plot(
    i,:), 'linewidth', 2)

end

base = zeros(180,360);
baseR = georefcells([-90 90],[0 360],size(base));
copperColor = [0.62 0.38 0.24];
geoshow(base,baseR,'FaceColor',copperColor)
camlight right
material([.8 .9 .4])

saveas(figureGlobe,'StarkSchedule_tsp_globe.png')
```

B.5.10 `seed_tsp_stars`

Uses Joseph Kirk's TSP Genetic Algorithm code [30] to solve for an optimal path among the nodes in a Fibonacci spiral, which MATLAB's `intlinprog` did not handle very well (due to the narrow "dynamic range" between the optimal solution and similar "good enough" solutions). Kirk's code also includes routines for solving the multi-salesman problem, which will be investigated in the future for further optimization of the LGS scheduler.

```
nstars = 259;

idxs = 1:nstars;
phi = acos(1-2.*(idxs-0.5)./nstars);
theta = pi*(1+sqrt(5))*(idxs-0.5);
x = 1.0*cos(theta).*sin(phi);
y = 1.0*sin(theta).*sin(phi);
z = 1.0*cos(phi);
```

```
xyz = [x' y' z'];

resultStruct = tsp_nn('xy',xyz);

%%
resultStruct2 = tsp_ga(resultStruct);

%%
save('tsp_fib.mat','resultStruct','resultStruct2');

%%
srcs = resultStruct2.optSolution(1:nstars);
dests = resultStruct2.optSolution(2:nstars+1);

dists = sqrt((x(srcs)-x(dests)).^2 + (y(srcs)-y(dests)).^2) + (z(srcs)-z(dests)).^2;

disp(rad2deg(mean(dists)))



figureGlobe2 = figure;
axesm('globe','Grid','on','GLineWidth',2,'MeridianLabel','on','MLabelParallel','
    equator','ParallelLabel','on','PLabelMeridian','prime')
set(gca, 'fontsize', 14,'linewidth',2)

plot3(x,y,z,'*','LineWidth',2)

for i = 1:nstars
    plot3(1.02*[x(srcs(i)) x(dests(i))],1.02*[y(srcs(i)) y(dests(i))],1.02*[z(srcs(i)
    ) z(dests(i))],'k-','LineWidth',2);
end

base = zeros(180,360);
baseR = georefcells([-90 90],[0 360],size(base));
copperColor = [0.62 0.38 0.24];
geoshow(base,baseR,'FaceColor',copperColor)
camlight right
material([.8 .9 .4])

%%
saveas(figureGlobe2,'skyglobe2.png')
```

B.6 Pathfinder

B.6.1 SkyCalcs

Calculates the line-of-sight and targets accessible from various ground telescopes through satellites in geostationary orbit.

```
% Geographic calculations -- requires Mapping toolbox

close all;

% Constants
Re = 6371000;
Rgeo = 42164000;
Ageo = 35786000;
E = wgs84Ellipsoid('meter');

% Locations
Klat = dms2degrees([19 49 35]); % Keck, HI
Klon = dms2degrees([-155 28 27]);
Kalt = 4145;
[Kx,Ky,Kz] = geodetic2ecef(E,Klat,Klon,Kalt);

Mlat = dms2degrees([31 41 18]); % MMT, AZ
Mlon = dms2degrees([-110 53 6]);
Malt = 2616;
[Mx,My,Mz] = geodetic2ecef(E,Mlat,Mlon,Malt);

% GSlat = -30.24073; % Gemini South, Chile
% GSlon = -70.73659;
GSlat = -30.24073; % Giant Magellan, Chile
GSlon = -70.73659;
GSalt = 2722;
[GSx,GSy,GSz] = geodetic2ecef(E,GSlat,GSlon,GSalt);

Glons = -180:0.1:180;
[Gx,Gy,Gz] = geodetic2ecef(E,0,Glons,Ageo);

[Kazs,Kels,Kras] = geodetic2aer(0,Glons,Ageo,Klat,Klon,Kalt,E);
[Mazs,Mels,Mras] = geodetic2aer(0,Glons,Ageo,Mlat,Mlon,Malt,E);
[GSazs,GSels,GSras] = geodetic2aer(0,Glons,Ageo,GSlat,GSlon,GSalt,E);

Kdx = Gx-Kx;
Kdy = Gy-Ky;
Kdz = Gz-Kz;
```

```
Kang = rad2deg(atan2(Kdz,sqrt(Kdx.^2+Kdy.^2)));

Mdx = Gx-Mx;
Mdy = Gy-My;
Mdz = Gz-Mz;

Mang = rad2deg(atan2(Mdz,sqrt(Mdx.^2+Mdy.^2)));

GSdx = Gx-GSx;
GSdy = Gy-GSy;
GSdz = Gz-GSz;

GSang = rad2deg(atan2(GSdz,sqrt(GSdx.^2+GSdy.^2)));

figCombo = figure;
hold on;
plot(Glons,Kang.*(Kels>10), 'linewidth', 2)
plot(Glons,Mang.*(Mels>10), 'linewidth', 2)
plot(Glons,GSang.*(GSels>10), 'linewidth', 2)
plot([9 21.5 25 31],[0 0 0 0],'kx', 'linewidth', 2)
plot([134],[0],'ko', 'linewidth', 2)
rectangle('Position',[-120 -0.2 40 0.4],'LineWidth',2)
h=text(9,0.2,'EDRS-A','FontSize',14);
set(h,'Rotation',45);
h=text(19,-0.2,'Artemis','FontSize',14);
set(h,'Rotation',-45);
h=text(25,0.2,'Inmarsat-4A F4','FontSize',14);
set(h,'Rotation',45);
h=text(35,-0.2,'EDRS-C','FontSize',14   );
set(h,'Rotation',-45);
text(134,0.3,'EDRS-D (TBC)','FontSize',14,'HorizontalAlignment','center');
text(-100,0.5,'LCRD (TBD)','FontSize',14,'HorizontalAlignment','center');
title('Sky coverage of demo with ground telescopes and GEO target(s)')
xlabel('Longitude of GEO target [deg]')
xlim([-180 180])
ylabel('Declination of Telescope-LGS line of sight [deg]')
legend('Keck','MMT','Gemini South')
hold off;
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figCombo,'GroundScope-GEO_Sky.png')

set(figCombo,'Units','Inches');
pos = get(figCombo,'Position');
set(figCombo,'PaperPositionMode','Auto','PaperUnits','Inches','PaperSize',[pos(3),
```

```
    pos(4)]);
print(figCombo,'GroundScope-GEO_Sky.pdf','-dpdf','-r0')

figMap = figure;
hold on;
title('Lines of sight from ground telescopes through GEO LGS')
plot([0 Re*cosd(Klat) Rgeo Rgeo+0.5*(Rgeo-Re*cosd(Klat))]/1e6,[0 Re*sind(Klat) 0 -0.5
    *Re*sind(Klat)]/1e6, 'linewidth', 2)
plot([0 Re*cosd(Mlat) Rgeo Rgeo+0.5*(Rgeo-Re*cosd(Mlat))]/1e6,[0 Re*sind(Mlat) 0 -0.5
    *Re*sind(Mlat)]/1e6, 'linewidth', 2)
plot([0 Re*cosd(GSlat) Rgeo Rgeo+0.5*(Rgeo-Re*cosd(GSlat))]/1e6,[0 Re*sind(GSlat) 0
    -0.5*Re*sind(GSlat)]/1e6, 'linewidth', 2)
plot(Rgeo/1e6,0,'kx', 'linewidth', 2)
plot([-1e7,7e7]/1e6,[0 0],'k:', 'linewidth', 2)
legend('Keck/TMT','MMT','GMT')
rectangle('Position',[-Re -Re 2*Re 2*Re]/1e6,'Curvature',[1 1], 'linewidth', 2)
ylim([-10 15])
xlabel('1000''s of km')
ylabel('1000''s of km')
%xlim([-10e6 50e6])
daspect([1 1 1])
hold off;
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figMap,'GroundScope-GEO_LOS.png')
set(figMap,'Units','Inches');
pos = get(figMap,'Position');
set(figMap,'PaperPositionMode','Auto','PaperUnits','Inches','PaperSize',[pos(3), pos
    (4)]);
print(figMap,'GroundScope-GEO_LOS.pdf','-dpdf','-r0')
```

B.6.2 SkyCalcsOffGEO

Calculates the delta-V cost to incline an LGS satellite from GEO to access non-equatorial stars.

```
% Geographic calculations -- requires Mapping toolbox

close all;

% Constants
Re = 6371000;
Rgeo = 42164000;
Ageo = 35786000;
Mu = 3.986e14;
```

```
Vgeo = sqrt(Mu/Rgeo);
E = wgs84Ellipsoid('meter');

% Locations

scopelat = dms2degrees([19 49 35]); % Keck, HI
% scopelon = dms2degrees([-155 28 27]);
scopelon = 70;
scopealt = 4145;
scopename = 'Keck';

% scopelat = dms2degrees([31 41 18]); % MMT, AZ
% % scopelon = dms2degrees([-110 53 6]);
% scopelon = 70;
% scopealt = 2616;
% scopename = 'MMT';

% scopelat = -30.24073; % Giant Magellan, Chile
% % scopelon = -70.73659;
% scopelon = 70;
% scopealt = 2722;
% scopename = 'GMT';

[scopex,scopey,scopez] = geodetic2ecef(E,scopelat,scopelon,scopealt);

lgslatvec = -90:90;
lgslonvec = -180:180;

[lgslat,lgslon] = ndgrid(lgslatvec,lgslonvec);
[lgsx,lgsy,lgsz] = geodetic2ecef(E,lgslat,lgslon,Ageo);

dvs = abs((pi/2)*deg2rad(lgslat)*Vgeo);

dx = lgsx-scopex;
dy = lgsy-scopey;
dz = lgsz-scopez;

dec = rad2deg(atan2(dz,sqrt(dx.^2+dy.^2))); % declination
rtas = rad2deg(atan2(dy,dx)); % Right ascension

[azs,els,ras] = geodetic2aer(lgslat,lgslon,Ageo,scopelat,scopelon,scopealt,E);

figureMap = figure;

% Hubble Deep Field (north), HDF South, HU(X)DF/Chandra South
```

```
deeplons = [189.2058,338.2343,53.1625];
deeplats = [62.2161,-60.5507,-27.7914];

axesm('MapProjection','robinson','Grid','on','GLineWidth',2)
title(sprintf('Delta-V cost to observe LUVOIR targets from %s',scopename))
p1 = scatterm(starlats,starlons,'*', 'linewidth', 2,'DisplayName','Stark 2015 targets
    ');
p2 = scatterm(deeplats,deeplons,'rv', 'linewidth', 2);
p3 = scatterm(brightlats,brightlons,'g+', 'linewidth', 2);
legend([p1 p3 p2],{'Stark 2015 targets','Magnitude 2 stars','Hubble/Chandra deep
    fields'})
% [Cel,hel] = contourm(lgslat,lgslon,els);

% [Cel,hel] = contourm(dec,rtas,els);
% clabelm(Cel,hel);

[Cel,hel] = contourm(dec,rtas,els,[10 10], 'linewidth', 2,'LineColor',[1 0 0]);

[Cdv,hdv] = contourm(dec,rtas,dvs, 'linewidth', 2);

tv = clabelm(Cdv,hdv);
tv.set('FontSize',14);
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figureMap,sprintf('SkyMap_offGEO %s.png',scopename))
```

B.6.3 SkyCalcsOffGEO2

Calculates the line-of-sight and targets accessible from various ground telescopes through a satellite in an inclined geosynchronous orbit.

```
% Geographic calculations -- requires Mapping toolbox and Phased Array?!?!
% (for rotx/roty/rotz)

close all;

% Constants
Re = 6371000;
Rgeo = 42164000;
Ageo = 35786000;
Mu = 3.986e14;
Vgeo = sqrt(Mu/Rgeo);
E = wgs84Ellipsoid('meter');

% Locations
```

```
scopelat = dms2degrees([19 49 35]); % Keck, HI
scopelon = dms2degrees([-155 28 27]);
scopealt = 4145;
scopename = 'Keck';

% scopelat = dms2degrees([31 41 18]); % MMT, AZ
% scopelon = dms2degrees([-110 53 6]);
% scopealt = 2616;
% scopename = 'MMT';

% scopelat = -30.24073; % Giant Magellan, Chile
% scopelon = -70.73659;
% scopealt = 2722;
% scopename = 'GMT';

% daydeg = 0:360;
daydeg = 154.5:0.0001:157; % Zoom in on star 173

scopelons = scopelon + daydeg;
scopelons = scopelons - 360*(scopelons>180);

scopelats = scopelat*ones(size(daydeg));
scopealts = scopealt*ones(size(daydeg));

[scopex,scopey,scopez] = geodetic2ecef(E,scopelats,scopelons,scopealts);

lgsV0 = 68.5; % true anomaly at epoch (i.e. start of day)
lgsVs = lgsV0 + daydeg;
lgsVs = lgsVs - 360*(lgsVs>360);

lgsIP = Rgeo*[cosd(lgsVs);...
    sind(lgsVs);...
    zeros(size(daydeg))]; % Column vectors representing in-plane coordinates of LGS.

lgsinc = 15; % inclination in degrees
lgsRAAN = 32.1; % lgs RAAN, degrees

R1 = rotx(lgsinc);
R2 = rotz(lgsRAAN);

lgsvecs = R2*(R1*lgsIP);

lgsx = lgsvecs(1,:);
lgsy = lgsvecs(2,:);
```

```
lgsz = lgsvecs(3,:);

% plot3(lgsx,lgsy,lgsz)

% dvs = abs(deg2rad(lgslat)*Vgeo);

dx = lgsx-scopex;
dy = lgsy-scopey;
dz = lgsz-scopez;

decs = rad2deg(atan2(dz,sqrt(dx.^2+dy.^2))); % declination
rtas = rad2deg(atan2(dy,dx)); % Right ascension

[azs,els,ras] = geodetic2aer(lgslat,lgslon,Ageo,scopelat,scopelon,scopealt,E);

[decs_mat,starlats_mat] = ndgrid(decs,starlats);
[rtas_mat,starlons_mat] = ndgrid(rtas,starlons);
[seps,~] = distance(decs_mat,rtas_mat,starlats_mat,starlons_mat);
% [closest_to_each_tgt,idx_to_each_tgt] = min(seps,[],2);

% disp(min(min(seps)))

%%
figureApproach = figure;
plot(((daydeg-daydeg(1))*24*60*60/360)*(366.25/365.25),seps(:,173), 'linewidth', 2)
title('Angular sep. of GEO LGS from target star')
xlabel('Time in encounter (sec)')
ylabel('Angle separation (deg)')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figureApproach,'GroundScope_OffGEO_window.png')

%%

figureApproachZoom = figure;
hold on
plot(((daydeg-daydeg(1))*24*60*60/360)*(366.25/365.25),seps(:,173)*3600, 'linewidth',
     2)
plot([272 282],[60 60],'-.', 'linewidth', 2)
plot([272 282],[25 25],'--', 'linewidth', 2)
hold off
legend('GEO LGS separation','Keck max distance to ground LGS','Keck max distance to
    NGS')
ylim([0 70])
title('Angular sep. of GEO LGS from target star')
xlabel('Time in encounter (sec)')
```

```
ylabel('Angle separation (arcsec)')
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figureApproachZoom,'GroundScope_OffGEO_window_1arcmin.png')

%%


daydeg = 0:0.1:360;
% daydeg = 154.5:0.0001:157; % Zoom in on star 173

scopelons = scopelon + daydeg;
scopelons = scopelons - 360*(scopelons>180);

scopelats = scopelat*ones(size(daydeg));
scopealts = scopealt*ones(size(daydeg));

[scopex,scopey,scopez] = geodetic2ecef(E,scopelats,scopelons,scopealts);

lgsV0 = 68.5; % true anomaly at epoch (i.e. start of day)
lgsVs = lgsV0 + daydeg;
lgsVs = lgsVs - 360*(lgsVs>360);

lgsIP = Rgeo*[cosd(lgsVs);...
    sind(lgsVs);...
    zeros(size(daydeg))]; % Column vectors representing in-plane coordinates of LGS.

lgsinc = 15; % inclination in degrees
lgsRAAN = 32.1; % lgs RAAN, degrees

R1 = rotx(lgsinc);
R2 = rotz(lgsRAAN);

lgsvecs = R2*(R1*lgsIP);

lgsx = lgsvecs(1,:);
lgsy = lgsvecs(2,:);
lgsz = lgsvecs(3,:);

% plot3(lgsx,lgsy,lgsz)

% dvs = abs(deg2rad(lgslat)*Vgeo);

dx = lgsx-scopex;
dy = lgsy-scopey;
dz = lgsz-scopez;
```

```
decs = rad2deg(atan2(dz,sqrt(dx.^2+dy.^2))); % declination
rtas = rad2deg(atan2(dy,dx)); % Right ascension

[decs_mat,starlats_mat] = ndgrid(decs,starlats);
[rtas_mat,starlons_mat] = ndgrid(rtas,starlons);
[seps,sep_azs] = distance(decs_mat,rtas_mat,starlats_mat,starlons_mat);
[closest_to_each_tgt,idx_to_each_tgt] = min(seps,[],2);

close_candidates = starids(unique(idx_to_each_tgt(closest_to_each_tgt<0.5)));

figureMap = figure;

axesm('MapProjection','robinson','Grid','on','GLineWidth',2,'MeridianLabel','on','
    MLabelParallel','equator','ParallelLabel','on','PLabelMeridian','prime')
title(sprintf('Line of sight from %s through inclined GEO LGS',scopename))
p1 = scatterm(starlats,starlons,'*', 'linewidth', 2,'DisplayName','Stark 2015 targets
    ');
p2 = scatterm(deeplats,deeplons,'rv', 'linewidth', 2);
p3 = scatterm(brightlats,brightlons,'g+', 'linewidth', 2);
p4 = plotm(decs,rtas,'linewidth',2);
legend([p1 p3 p2 p4],{'Stark 2015 targets','Magnitude 2 stars','Hubble/Chandra deep
    fields','LGS orbit trace'})

set(gca, 'fontsize', 14,'linewidth',2)
saveas(figureMap,sprintf('SkyMap_offGEO_track %s.png',scopename))
```

B.6.4 SkyCalcsHEO

Calculates the line-of-sight from a ground telescope through a satellite in a highly elliptical orbit.

```
% Geographic calculations -- requires Mapping toolbox and Phased Array?!?!
% (for rotx/roty/rotz)

close all;

% Constants
Re = 6371000;
Rgeo = 42164000;
Ageo = 35786000;
Mu = 3.986e14;
Vgeo = sqrt(Mu/Rgeo);
E = wgs84Ellipsoid('meter');
```

```
Tgeo = 2*pi*sqrt(Rgeo^3/Mu);

[earthx,earthy,earthz] = sphere;
earthx = E.MeanRadius*earthx;
earthy = E.MeanRadius*earthy;
earthz = E.MeanRadius*earthz;

% Locations

% scopelat = dms2degrees([19 49 35]); % Keck, HI
% scopelon = dms2degrees([-155 28 27]);
% scopealt = 4145;
% scopename = 'Keck';

scopelat = dms2degrees([0 0 0]); % Geostationary telescope testbed
scopelon = dms2degrees([-120 0 0]);
scopealt = Ageo;
scopename = 'GeoTT';

% scopelat = dms2degrees([31 41 18]); % MMT, AZ
% scopelon = dms2degrees([-110 53 6]);
% scopealt = 2616;
% scopename = 'MMT';

% scopelat = -30.24073; % Giant Magellan, Chile
% scopelon = -70.73659;
% scopealt = 2722;
% scopename = 'GMT';

simtime = 0:10:14*24*60*60; % 6 days, every 10 seconds
% simtime = 2.2*daysec:10:2.4*daysec; % 10 seconds, from 2.2 to 2.4 days after epoch
% simtime = 53*3600:10:55.5*3600;

scopelons = scopelon + 360*(simtime/Tgeo);

scopelats = scopelat*ones(size(simtime));
scopealts = scopealt*ones(size(simtime));

[scopex,scopey,scopez] = geodetic2ecef(E,scopelats,scopelons,scopealts);


% LGS orbit, minimum sidereal case

% lgs_ap = rSIDmin*1000;
% lgs_pe = rLEO*1000;
```

```
% sidereal, periapsis at GEO

% lgs_ap = rSIDopt*1000;
% lgs_pe = Rgeo;

% the above sidereal orbits are attempting to match the motion of Earth *at
% the Equator*, let's try something a little slower

lgs_ap = rSIDopt*1000;
lgs_pe = Rgeo/2;

lgs_sma = (lgs_ap+lgs_pe)/2;
lgs_mam = sqrt(Mu/lgs_sma^3); % mean angular motion

lgs_ecc = (lgs_ap-lgs_pe)/(lgs_ap+lgs_pe);

lgsV0 = deg2rad(68.5); % true anomaly at epoch (i.e. start of simulation)
lgsE0 = atan2(sqrt(1-lgs_ecc^2)*sin(lgsV0),lgs_ecc+cos(lgsV0));
lgsM0 = lgsE0-lgs_ecc*sin(lgsE0);

lgsMs = lgsM0+lgs_mam*simtime;
lgsEs = ecc_from_mean(lgsMs,lgs_ecc);
lgsVs = atan2(sqrt(1-lgs_ecc^2)*sin(lgsEs),cos(lgsEs)-lgs_ecc);

lgsRs = lgs_sma*(1-lgs_ecc^2)./(1+lgs_ecc*cos(lgsVs));

lgsIP = [lgsRs.*cos(lgsVs);...
    lgsRs.*sin(lgsVs);...
    zeros(size(simtime))]; % Column vectors representing in-plane coordinates of LGS.

lgsAPE = 90; % argument of periapsis, degrees
lgsinc = 60; % inclination in degrees
lgsRAAN = 32.1; % lgs RAAN, degrees

R0 = rotz(lgsAPE);
R1 = rotx(lgsinc);
R2 = rotz(lgsRAAN);

lgsvecs = R2*(R1*(R0*lgsIP));

lgsx = lgsvecs(1,:);
lgsy = lgsvecs(2,:);
lgsz = lgsvecs(3,:);
```

```
dx = lgsx-scopex;
dy = lgsy-scopey;
dz = lgsz-scopez;

dists = sqrt(dx.^2+dy.^2+dz.^2);

decs = rad2deg(atan2(dz,sqrt(dx.^2+dy.^2))); % declination
rtas = rad2deg(atan2(dy,dx)); % Right ascension

ddecs = diff(decs)./diff(simtime); % degrees-per-second difference from one moment to
     the next
drtas = diff(rtas)./diff(simtime);

driftrate = sqrt(ddecs.^2+drtas.^2); % this is only valid near the equator, TODO
    improve

%%
figureDrift = figure;
plot(simtime(1:end-1)/daysec,driftrate*3600*1000,'linewidth',2);
ylim([0 35])
set(gca, 'fontsize', 14,'linewidth',2)
xlabel('Time (days)')
ylabel('Drift rate (mas/sec)')
saveas(figureDrift,sprintf('SkyMap_HEO_track_stability_total %s.png',scopename))

%%
figureStab = figure;
hold on
plot(drtas*3600*1000,ddecs*3600*1000,'linewidth',2);
plot([35,35,-35,-35,35],[35,-35,-35,35,35],'linewidth',2);
xlim([-500,500])
ylim([-500,500])
daspect([1 1 1])
set(gca, 'fontsize', 14,'linewidth',2)
hold off
saveas(figureStab,sprintf('SkyMap_HEO_track_stability_2d %s.png',scopename))

%%

obs_idx_start = find(simtime==53.3*3600);
obs_idx_end = find(simtime==55.1*3600);

obsx = [scopex(obs_idx_start) lgsx(obs_idx_start) lgsx(obs_idx_end) scopex(
    obs_idx_end)];
obsy = [scopey(obs_idx_start) lgsy(obs_idx_start) lgsy(obs_idx_end) scopey(
```

```
    obs_idx_end)];
obsz = [scopez(obs_idx_start) lgsz(obs_idx_start) lgsz(obs_idx_end) scopez(
    obs_idx_end)];

figureXYZ = figure;
hold on
plot3(lgsx,lgsy,lgsz,'linewidth',2)
plot3(scopex,scopey,scopez,'linewidth',2)
plot3(obsx,obsy,obsz,'linewidth',2)
surf(earthx,earthy,earthz)
legend('LGS orbit','Telescope latitude','Best observation vector','location','
    southeast')
daspect([1 1 1])
view(45,30)
hold off
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figureXYZ,sprintf('SkyMap_HEO_orbit %s.png',scopename))

%%

rtas_obs = rtas(simtime>53.3*3600 & simtime < 55.1*3600);
decs_obs = decs(simtime>53.3*3600 & simtime < 55.1*3600);
simtime_obs = simtime(simtime>53.3*3600 & simtime < 55.1*3600);
seps_obs = sqrt((rtas_obs-mean(rtas_obs)).^2+(decs_obs-mean(decs_obs)).^2);
figureXY = figure;
hold on
thetas = 0:0.01:2*pi;
plot((rtas_obs-mean(rtas_obs))*3600-10,(decs_obs-mean(decs_obs))*3600-10,'linewidth'
    ,2)
plot(60*cos(thetas),60*sin(thetas),'-.','linewidth',2)
plot(25*cos(thetas),25*sin(thetas),'--','linewidth',2)
title('Angular sep. of HEO LGS from target star')
legend('HEO LGS separation','Keck max distance to ground LGS','Keck max distance to
    NGS')
xlabel('Delta-Dec (arcsec)')
ylabel('Delta-RA (arcsec)')
xlim([-80 80])
ylim([-80 80])
daspect([1 1 1])
hold off
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figureXY,sprintf('SkyMap_HEO_track_zoom %s.png',scopename))

%%
figureMap = figure;
```

```
axesm('MapProjection','robinson','Grid','on','GLineWidth',2,'MeridianLabel','on','
    MLabelParallel','equator','ParallelLabel','on','PLabelMeridian','prime')
title(sprintf('Line of sight from %s through HEO LGS',scopename))
p1 = scatterm(starlats,starlons,'*', 'linewidth', 2,'DisplayName','Stark 2015 targets
    ');
p2 = scatterm(deeplats,deeplons,'rv', 'linewidth', 2);
p3 = scatterm(brightlats,brightlons,'g+', 'linewidth', 2);
p4 = plotm(decs,rtas,'linewidth',2);
legend([p1 p3 p2 p4],{'Stark 2015 targets','Magnitude 2 stars','Hubble/Chandra deep
    fields','LGS orbit trace'})

set(gca, 'fontsize', 14,'linewidth',2)
saveas(figureMap,sprintf('SkyMap_HEO_track %s.png',scopename))
```

B.6.5 SkyCalcsHEO2

Calculates the line-of-sight from a telescope in geostationary orbit through a satellite in a super-geostationary orbit.

```
% Geographic calculations -- requires Mapping toolbox and Phased Array?!?!
% (for rotx/roty/rotz)

close all;

% Constants
Re = 6371000;
Rgeo = 42164000;
Ageo = 35786000;
Mu = 3.986e14;
Vgeo = sqrt(Mu/Rgeo);
E = wgs84Ellipsoid('meter');
Tgeo = 2*pi*sqrt(Rgeo^3/Mu);

[earthx,earthy,earthz] = sphere;
earthx = E.MeanRadius*earthx;
earthy = E.MeanRadius*earthy;
earthz = E.MeanRadius*earthz;

scopelat = dms2degrees([0 0 0]); % Geostationary telescope testbed
scopelon = dms2degrees([-120 0 0]);
scopealt = Ageo;
scopename = 'GeoTT';
```

```
simtime = 0:10:15*24*60*60; % 6 days, every 10 seconds
% simtime = 2.2*daysec:10:2.4*daysec; % 10 seconds, from 2.2 to 2.4 days after epoch
% simtime = 53*3600:10:55.5*3600;

scopelons = scopelon + 360*(simtime/Tgeo);

scopelats = scopelat*ones(size(simtime));
scopealts = scopealt*ones(size(simtime));

[scopex,scopey,scopez] = geodetic2ecef(E,scopelats,scopelons,scopealts);


% LGS super-sync 7/6 resonant orbit, with intervals of matching geo motion

lgs_sma = ((7/6)^(2/3))*Rgeo;
lgs_mean_vel = sqrt(Mu/lgs_sma);

lgs_pe = 2/(Vgeo^2/(Mu)+1/lgs_sma);
lgs_ap = 2*lgs_sma-lgs_pe;

lgs_mam = sqrt(Mu/lgs_sma^3); % mean angular motion

lgs_ecc = (lgs_ap-lgs_pe)/(lgs_ap+lgs_pe);

lgsV0 = deg2rad(-120); % true anomaly at epoch (i.e. start of simulation)
lgsE0 = atan2(sqrt(1-lgs_ecc^2)*sin(lgsV0),lgs_ecc+cos(lgsV0));
lgsM0 = lgsE0-lgs_ecc*sin(lgsE0);

lgsMs = lgsM0+lgs_mam*simtime;
lgsEs = ecc_from_mean(lgsMs,lgs_ecc);
lgsVs = atan2(sqrt(1-lgs_ecc^2)*sin(lgsEs),cos(lgsEs)-lgs_ecc);

lgsRs = lgs_sma*(1-lgs_ecc^2)./(1+lgs_ecc*cos(lgsVs));

lgsIP = [lgsRs.*cos(lgsVs);...
    lgsRs.*sin(lgsVs);...
    zeros(size(simtime))]; % Column vectors representing in-plane coordinates of LGS.

lgsAPE = 0; % argument of periapsis, degrees
lgsinc = 0; % inclination in degrees
lgsRAAN = 0; % lgs RAAN, degrees

R0 = rotz(lgsAPE);
R1 = rotx(lgsinc);
R2 = rotz(lgsRAAN);
```

```
lgsvecs = R2*(R1*(R0*lgsIP));

lgsx = lgsvecs(1,:);
lgsy = lgsvecs(2,:);
lgsz = lgsvecs(3,:);

dx = lgsx-scopex;
dy = lgsy-scopey;
dz = lgsz-scopez;

dists = sqrt(dx.^2+dy.^2+dz.^2);

decs = rad2deg(atan2(dz,sqrt(dx.^2+dy.^2))); % declination
rtas = rad2deg(atan2(dy,dx)); % Right ascension

ddecs = diff(decs)./diff(simtime); % degrees-per-second difference from one moment to
     the next
drtas = diff(rtas)./diff(simtime);

driftrate = sqrt(ddecs.^2+drtas.^2); % this is only valid near the equator, TODO
    improve

%%
figureDrift = figure;
plot(simtime(1:end-1)/daysec,driftrate*3600*1000,'linewidth',2);
ylim([0 35])
set(gca, 'fontsize', 14,'linewidth',2)
xlabel('Time (days)')
ylabel('Drift rate (mas/sec)')
saveas(figureDrift,sprintf('SkyMap_HEO_track_stability_total %s.png',scopename))

%%
figureStab = figure;
hold on
plot(drtas*3600*1000,ddecs*3600*1000,'linewidth',2);
plot([35,35,-35,-35,35],[35,-35,-35,35,35],'linewidth',2);
xlim([-500,500])
ylim([-500,500])
daspect([1 1 1])
set(gca, 'fontsize', 14,'linewidth',2)
hold off
saveas(figureStab,sprintf('SkyMap_HEO_track_stability_2d %s.png',scopename))

%%
```

```
obs_idx_start = 800; % Should be possible to calculate and not hard-code
obs_idx_end = 870;

obsx = [scopex(obs_idx_start) lgsx(obs_idx_start) lgsx(obs_idx_end) scopex(
    obs_idx_end)];
obsy = [scopey(obs_idx_start) lgsy(obs_idx_start) lgsy(obs_idx_end) scopey(
    obs_idx_end)];
obsz = [scopez(obs_idx_start) lgsz(obs_idx_start) lgsz(obs_idx_end) scopez(
    obs_idx_end)];

figureXYZ = figure;
hold on
plot3(lgsx,lgsy,lgsz,'linewidth',2)
plot3(scopex,scopey,scopez,'linewidth',2)
plot3(obsx,obsy,obsz,'linewidth',2)
surf(earthx,earthy,earthz)
legend('LGS orbit','Telescope orbit','Best observation vector','location','southeast'
    )
daspect([1 1 1])
view(45,30)
hold off
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figureXYZ,sprintf('SkyMap_HEO_orbit %s.png',scopename))

%%

rtas_obs = rtas(obs_idx_start:obs_idx_end);
decs_obs = decs(obs_idx_start:obs_idx_end);
simtime_obs = simtime(obs_idx_start:obs_idx_end);
seps_obs = sqrt((rtas_obs-mean(rtas_obs)).^2+(decs_obs-mean(decs_obs)).^2);
figureXY = figure;
hold on
thetas = 0:0.01:2*pi;
plot((rtas_obs-mean(rtas_obs))*3600,(decs_obs-mean(decs_obs))*3600,'linewidth',2)
plot(60*cos(thetas),60*sin(thetas),'-.','linewidth',2)
plot(25*cos(thetas),25*sin(thetas),'--','linewidth',2)
title('Angular sep. of HEO LGS from target star')
legend('HEO LGS separation','Keck max distance to ground LGS','Keck max distance to
    NGS')
xlabel('Delta-Dec (arcsec)')
ylabel('Delta-RA (arcsec)')
xlim([-80 80])
ylim([-80 80])
daspect([1 1 1])
```

```
hold off
set(gca, 'fontsize', 14,'linewidth',2)
saveas(figureXY,sprintf('SkyMap_HEO_track_zoom %s.png',scopename))

%%
figureMap = figure;

axesm('MapProjection','robinson','Grid','on','GLineWidth',2,'MeridianLabel','on','
    MLabelParallel','equator','ParallelLabel','on','PLabelMeridian','prime')
title(sprintf('Line of sight from %s through HEO LGS',scopename))
p1 = scatterm(starlats,starlons,'*', 'linewidth', 2,'DisplayName','Stark 2015 targets
    ');
p2 = scatterm(deeplats,deeplons,'rv', 'linewidth', 2);
p3 = scatterm(brightlats,brightlons,'g+', 'linewidth', 2);
p4 = plotm(decs,rtas,'linewidth',2);
legend([p1 p3 p2 p4],{'Stark 2015 targets','Magnitude 2 stars','Hubble/Chandra deep
    fields','LGS orbit trace'})

set(gca, 'fontsize', 14,'linewidth',2)
saveas(figureMap,sprintf('SkyMap_HEO_track %s.png',scopename))
```

B.6.6 ecc_from_mean

A helper function for calculating the eccentric anomaly from the mean anomaly.

```
function [E] = ecc_from_mean(M,e)
    format long
    E = M;
    k = 1;
    err = 1e-10;
    while (max(k)>err)
        y = E - e*sin(E) - M;
        dy = 1 - e*cos(E);
        k = abs(y./dy);
        E = E-(y./dy);
    end
end
```

B.6.7 HEO_LGS

Calculations of highly-elliptical “sidereal” orbits for Earth-orbiting laser guide stars.

```
close all
```

```
mu = 398600;

RE = 6378;
rLEO = RE+400;
rGEO = RE+35786;
aLUN = 384400;
rLUN1 = 362600;
rLUN2 = 405400;
bLUN = sqrt(rLUN1*rLUN2);
vLEO = sqrt(mu/rLEO);
vESC = sqrt(2*mu/rLEO);
vGEO = sqrt(mu/rGEO);
tGEO = 2*pi*sqrt(rGEO^3/mu); % 1 sidereal day

rVABi1 = (1+1)*RE; % inner Van Allen Belt starts at ~1,000 km altitude (i.e. 1.17 RE
    from center), but it peaks at 1-3 RE altitude = 2-4 from center.
rVABi2 = (1+3)*RE;

rVABo1 = (1+4)*RE; % outer Van Allen Belt peaks 4-6 RE altitude
rVABo2 = (1+6)*RE;

vGTO1 = sqrt(2*mu*(1/rLEO-1/(rLEO+rGEO)));

vGTO2 = sqrt(2*mu*(1/rGEO-1/(rLEO+rGEO)));

aGTO = (rLEO+rGEO)/2;
bGTO = sqrt(rLEO*rGEO);

dV_leo_geo = (vGTO1-vLEO)+(vGEO-vGTO2);



vSID = 2*pi*RE/tGEO;

rSIDcirc = mu/vSID^2;


%minimum sidereal-apogee orbit, so 400 x 150,000 km orbit

rSIDmins = roots([1 rLEO -2*mu*rLEO/vSID^2]);

rSIDmin = max(rSIDmins);

tSIDmin = 2*pi*sqrt(((rSIDmin+rLEO)/2)^3/mu);
```

```
tGTO = 2*pi*sqrt(aGTO^3/mu);

vSIDminP = sqrt(2*mu*(1/rLEO-1/(rLEO+rSIDmin)));

dv_leo_sid_min = vSIDminP-vLEO;

dv_gto_sid_min = vSIDminP-vGTO1;

aSIDmin = (rLEO+rSIDmin)/2;
bSIDmin = sqrt(rLEO*rSIDmin);

max_burn_d = 2*RE/vSIDminP; % Assume spacecraft moves at a constant v at periapsis,
    how long does it take to move across the Earth (i.e. "time close enough to
    periapsis")

num_burns = (dv_gto_sid_min*1000*sc_mass_opt_tot/sc_max_thrust_nom)/max_burn_d; %
    Need to multiply by 1000 to get from km/s to m/s

approx_depl_duration = num_burns*(tGTO+tSIDmin)/2;

%%
% Let's calculate the deployment duration more exactly

test_sc_vel = vGTO1;
depl_duration = 0;
burn_count = 0;

while test_sc_vel < vSIDminP
    burn_count = burn_count + 1;
    burn_d = 2*RE/test_sc_vel;% Assume spacecraft moves at a constant v at periapsis,
     how long does it take to move across the Earth (i.e. "time close enough to
    periapsis")
    test_sc_vel = test_sc_vel + burn_d*sc_max_thrust_nom/(sc_mass_opt_tot*1000); %
    need to divide by 1000 to get from m/s to km/s
    test_sc_sma = (mu)/((2*mu/rLEO)-test_sc_vel^2);
    test_sc_pd = 2*pi*sqrt(test_sc_sma^3/mu);
    depl_duration = depl_duration+test_sc_pd;
end

% End result turns out to be 5168 burns (vs. 5300 preducted above), total
% duration 14.6 years vs. 20 predicted.  Not bad, but good to know to feed
% SPENVIS.

mass_shield = 0.001*(2*20*20+4*20*30)*0.7*2.7; % aluminum 0.7 cm (!) thick
```

```
%% What if we wanted to get there in, say, six months?
% There should be a way to solve for this directly...

test_sc_vel = vGTO1;
depl_duration_ht = 0;
burn_count_ht = 0;
thrust_factor = 30; % 30 turns out to be about right, total duration 181 days

while test_sc_vel < vSIDminP
    burn_count_ht = burn_count_ht + 1;
    burn_d = 2*RE/test_sc_vel;% Assume spacecraft moves at a constant v at periapsis,
     how long does it take to move across the Earth (i.e. "time close enough to
    periapsis")
    test_sc_vel = test_sc_vel + thrust_factor*burn_d*sc_max_thrust_nom/(
    sc_mass_opt_tot*1000); % need to divide by 1000 to get from m/s to km/s
    test_sc_sma = (mu)/((2*mu/rLEO)-test_sc_vel^2);
    test_sc_pd = 2*pi*sqrt(test_sc_sma^3/mu);
    depl_duration_ht = depl_duration_ht+test_sc_pd;
end


%%
rad_count = 0.05*(RE/vSIDminP)*2*num_burns; % Assume Van Allen Belt's dose is
    concentrated at 0.05 rad/sec over a 1 RE thickness, per: https://spacemath.gsfc.
    nasa.gov/Algebra1/3Page7.pdf

% Verifying radiation dosage information
vGTOvab = sqrt(mu*(2/(2*RE)-(1/aGTO))); % velocity of spacecraft in GTO at van Allen
    belt (i.e. 1 RE altitude, 2 RE radius)
rad_gto_yr = 0.05*2*(RE/vGTOvab)*(yrsec/tGTO); % 2 passes through the belts per orbit
    , times ~800 orbits per year, comes out to 100 krad per yer -- but other sources
    suggest more like 2.5 per year?

vSIDminvab = sqrt(mu*(2/(2*RE)-(1/aSIDmin))); % velocity of spacecraft on minimal
    sidereal orbit at van Allen belt (i.e. 1 RE altitude, 2 RE radius)

rad_gto_smad_noshld_yr = (3e6/yrsec)*2*(RE/vGTOvab)*(yrsec/tGTO); % 150 krad/yr,
    using SMAD fig 7-11 p 135.
rad_gto_smad_3mm_yr = (3e4/yrsec)*2*(RE/vGTOvab)*(yrsec/tGTO); % 1.5 krad/yr w/ 0.8 g
    /cm2 of aluminum everywhere (so 3 mm thick -- that's 2.5 kg to cover a whole 12U
    satellite)

rad_sid_smad_noshld = (3e6/yrsec)*2*(RE/vSIDminvab)*num_burns; % 800 krad!!!
rad_sid_smad_maxshld = (1e4/yrsec)*2*(RE/vSIDminvab)*num_burns; % 2 krad, even after
    cladding the spacecraft in 8 kg of aluminum (that's 1-cm-thick aluminum, and also
```

```
    1/3rd the mass budget)

%What if we wanted to keep the periapsis at GEO?

rSIDopts = roots([1 rGEO -2*mu*rGEO/vSID^2]);

rSIDopt = max(rSIDopts); % turns out to be ~lunar distance

aSIDopt = (rGEO+rSIDopt)/2;
bSIDopt = sqrt(rSIDopt*rGEO);

tSIDopt = 2*pi*sqrt(((rSIDopt+rGEO)/2)^3/mu);

vSIDoptP = sqrt(2*mu*(1/rGEO-1/(rGEO+rSIDmin)));

% going from GEO to SID-OPT orbit
dv_geo_sid_opt = vSIDoptP-vGEO;


vST01 = sqrt(2*mu*(1/rLEO-1/(rLEO+rSIDopt)));

vST02 = sqrt(2*mu*(1/rSIDopt-1/(rLEO+rSIDopt)));

dv_gto_sid_opt = (vST01-vGT01) + (vSID-vST02);

dv_inc = (pi/2)*vGEO*deg2rad(15);

time_inc = (dv_inc*1000*sc_mass_opt_tot/sc_max_thrust_nom);

% Make inner and outer boundaries of each belt
t = linspace(0,2*pi,100);
xi1 = rVABi1*cos(t);
xi2 = rVABi2*cos(t);
yi1 = rVABi1*sin(t);
yi2 = rVABi2*sin(t);

xo1 = rVABo1*cos(t);
xo2 = rVABo2*cos(t);
yo1 = rVABo1*sin(t);
yo2 = rVABo2*sin(t);


%%
figureOrbits = figure;
hold on
```

```
axis equal
rectangle('Position',[-RE -RE 2*RE 2*RE],'Curvature',[1 1],'FaceColor',[0 .5 .5],'
    LineStyle','none'); % Earth
% rectangle('Position',[-rLEO -rLEO 2*rLEO 2*rLEO],'Curvature',[1 1],'LineWidth',2);
rectangle('Position',[-rLEO -bGTO 2*aGTO 2*bGTO],'Curvature',[1 1],'LineWidth',2,'
    LineStyle',':'); % GTO
rectangle('Position',[-rGEO -rGEO 2*rGEO 2*rGEO],'Curvature',[1 1],'LineWidth',2,'
    LineStyle','--'); % GEO
title('Comparison of different LGS orbits')
xlabel('Distance from Earth center (km)')
ylabel('Distance from Earth center (km)')

plot([(aGTO-rLEO) 1.3e5],[-bGTO -1.85e5],'k:','linewidth',1.5)
plot([0 -0.7e5],[rGEO 1.35e5],'k--','linewidth',1.5)

plot([-3*RE -0.9e5 -6*RE],[0 -1e5 0],'k')

text(-1e5,1.5e5,'GEO','FontSize',14)
text(-1e5,-1e5,'Van Allen belts','FontSize',14,'HorizontalAlignment','right')
text(1e5,-2e5,'GTO','FontSize',14)

text(3e5,2.3e5,'Moon','FontSize',14)
text(1e5,0.5e5,'Sidereal orbit 1','FontSize',14)
text(1e5,1.5e5,'Sidereal orbit 2','FontSize',14)

set(gca, 'fontsize', 14,'linewidth',2)

rectangle('Position',[-rLEO -bSIDmin 2*aSIDmin 2*bSIDmin],'Curvature',[1 1],'
    LineWidth',2);
rectangle('Position',[-rGEO -bSIDopt 2*aSIDopt 2*bSIDopt],'Curvature',[1 1],'
    LineWidth',2);
rectangle('Position',[-rLUN2 -bLUN 2*aLUN 2*bLUN],'Curvature',[1 1],'LineWidth',2,'
    EdgeColor',[0.5 0.5 0.5]);

hp1 = patch([xi2,xi1],[yi2,yi1],'r','linestyle','none','facealpha',0.3);
hp2 = patch([xo2,xo1],[yo2,yo1],'g','linestyle','none','facealpha',0.3);

hold off
saveas(figureOrbits,'lgs-orbit-comparison.png')
```

B.7 hamiltonian

Code for computing a Hamiltonian path from a graph, by Pramit Biswas. [8] Used according to the terms of the 2-clause BSD license.

```
%%
% Let us create the following graph
%       (1)--(2)--(3)-------(4)
%        |   / \   |      |
%        |  /   \  |      |
%        | /     \ |      |
%       (5)-------(6)     |
%   |           |
%   |           |
%   |           |
%  (7)-----------------(8)
%
% g=[0 1 0 0 1 0 0 0;
%    1 0 1 0 1 1 0 0;
%    0 1 0 1 0 1 0 0;
%    0 0 1 0 0 0 0 1;
%    1 1 0 0 0 1 1 0;
%  0 1 1 0 1 0 0 0;
%  0 0 0 0 1 0 0 1;
%  0 0 0 1 0 0 1 0]
% s=5; % Source
% d=1; % Destination
%
% P = hamiltonianPath(g,s,d);
%
% P will be an array mentioning the path/cycle, if path/cycle found; or a
% string: 'No Path Found', if path/cycle not found
%
% #Note: This code can be used for finding Hamiltonian cycle also. For
% that, make sure Source and Destination are same.
%%
%{
    Main Function
%}
function hamPath = hamiltonian(Graph, Source, Destination)

% Input Checking
if ~isreal(Graph)
    error('Graph must be in real form');
elseif ~isnumeric(Graph)
```

```
    error('Matrix must be numeric');
elseif ~ismatrix(Graph)
    error('Check Matrix Dimensions');
else
    [r, c] = size (Graph);
    if r~=c
        error('Matrix must be square matrix');
    end
end

if ~(isreal(Source)||isreal(Destination)||(Source>0 && Source<=r) || (Destination>0
    && Destination<=r))
    error('improper Source/Destination');
end

clear c;

% Function call
hamPath = findHam(Graph, Source, Destination, r);

end

%%
%{
    This functions sets some initial parameters, and calls the actual
    function.
%}
function hamPath = findHam(Graph, Source, Destination, totalNodes)

hamPath = zeros(size(Graph(1,:)));

hamPath(1) = Source;

[Status, hamPath] = hamRec(Graph, hamPath, Source, Destination, totalNodes, 1);

if Status == 0
    if Source ~= Destination
        hamPath = 'No Path Found';
    else
        hamPath = 'No Cycle Found';
    end
    return;
end

end
```

```
%%
%{
    This function recursively call itself, hence finding the solution
%}
function [Status, hamPath] = hamRec(Graph, hamPath, Source, Destination, totalNodes,
    nodesFound)

% Ending Condition check
if ( (nodesFound == totalNodes-1 && Source~=Destination) || (nodesFound == totalNodes
     && Source==Destination) )
    if ( Graph(hamPath(nodesFound), Destination) ~= 0)
        hamPath(nodesFound+1) = Destination;
        Status = 1;
        return;
    else
        Status = 0;
        return;
    end
end

for i=1:totalNodes
    if i==Destination
        continue;
    end

    if isSafe(Graph, hamPath, nodesFound, i)
        hamPath(nodesFound+1) = i;

        [Status, hamPath] = hamRec(Graph, hamPath, Source, Destination, totalNodes,
    nodesFound+1);
        if Status
            return;
        end

        hamPath(nodesFound+1) = 0;
    end
end

Status = 0;

end

%%
%{
```

```
    This function is used to check whether the current node can be added
    or not for making the path/cycle.
%}
function Flag = isSafe(Graph, hamPath, nodesFound, i)

if Graph(hamPath(nodesFound),i) == 0
    Flag = 0;
    return;
end

for ii=1:nodesFound
    if hamPath(ii) == i
        Flag = 0;
        return;
    end
end

Flag = 1;

end
```

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix C

LGS Data Files

These are the CSV files that are ingested by `StarkSkymap.m` to provide the locations of the target stars and the list of observations.

C.1 Targets

The list of stars which are baselined to be observed by LUVOIR, per Stark *et al.* 2015 [57], received from Chris Stark via personal communication. Each row consists of the star ID (Hipparcos Catalogue), followed by the hour angle and declination, as sourced from SIMBAD.

`simbad-trim.csv`

```
HIP 439    ;00 05 24.4278018131 -37 21 26.504223709
HIP 544    ;00 06 36.7840943605 +29 01 17.410390108
HIP 910    ;00 11 15.8508998160 -15 28 04.740699873
HIP 950    ;00 11 44.0207940487 -35 07 59.231973741
HIP 1292   ;00 16 12.6791488703 -79 51 04.244738563
HIP 1475   ;00 18 22.8849667932 +44 01 22.637271919
HIP 1599   ;00 20 04.2599713 -64 52 29.254760
HIP 1803   ;00 22 51.7883318927 -12 12 33.972374893
HIP 3093   ;00 39 21.8055114029 +21 15 01.716052732
HIP 3583   ;00 45 45.5929656213 -47 33 07.146506536
HIP 3765   ;00 48 22.9763438736 +05 16 50.209562003
HIP 3821   ;00 49 06.2907161 +57 48 54.675740
HIP 3909   ;00 50 07.5885892159 -10 38 39.584811061
HIP 4151   ;00 53 04.1958850619 +61 07 26.301822158
HIP 5336   ;01 08 16.39470 +54 55 13.2264
HIP 5862   ;01 15 11.1225892742 -45 31 54.007631265
HIP 7339   ;01 34 33.2643888977 +68 56 53.288116227
HIP 7513   ;01 36 47.84216 +41 24 19.6443
HIP 7751   ;01 39 47.53953 -56 11 47.0997
HIP 7978   ;01 42 29.3148822519 -53 44 26.991165270
HIP 7981   ;01 42 29.7634932653 +20 16 06.660242064
HIP 8102   ;01 44 04.0834226 -15 56 14.926552
HIP 8362   ;01 47 44.8336250528 +63 51 09.007331430
HIP 10138  ;02 10 25.9190575041 -50 49 25.467227759
HIP 10644  ;02 17 03.2353876410 +34 13 27.243444846
HIP 10798  ;02 18 58.5046943868 -25 56 44.473476858
HIP 12114  ;02 36 04.9023844802 +06 53 12.431588708
HIP 12444  ;02 40 12.4221262303 -09 27 10.336039665
HIP 12653  ;02 42 33.4664838826 -50 48 01.056222150
HIP 12777  ;02 44 11.9870420 +49 13 42.411120
HIP 12843  ;02 45 06.2034405658 -18 34 21.478629972
HIP 13402  ;02 52 32.1281853696 -12 46 10.970649463
HIP 14632  ;03 09 04.0198629 +49 36 47.799638
HIP 15330  ;03 17 46.1632605674 -62 34 31.154247481
HIP 15371  ;03 18 12.8185412558 -62 30 22.917300282
HIP 15457  ;03 19 21.6963205 +03 22 12.715139
HIP 15510  ;03 19 55.6509352 -43 04 11.217495
```

```
HIP 16537  ;03 32 55.8449634 -09 27 29.731165
HIP 16852  ;03 36 52.1448022536 +00 23 58.536917738
HIP 17378  ;03 43 14.9008787 -09 45 48.208444
HIP 17651  ;03 46 50.8881911 -23 14 59.004585
HIP 18859  ;04 02 36.7451443573 -00 16 08.118566607
HIP 19076  ;04 05 20.2584363613 +22 00 32.055953585
HIP 19335  ;04 08 36.6168168411 +38 02 23.055834828
HIP 19849  ;04 15 16.3197260 -07 39 10.338087
HIP 19859  ;04 15 28.8004956278 +06 11 12.699896698
HIP 21770  ;04 40 33.71305 -41 51 49.5075
HIP 22263  ;04 47 36.2917607568 -16 56 04.041927355
HIP 22449  ;04 49 50.4109057 +06 57 40.588294
HIP 23311  ;05 00 48.9991346145 -05 45 13.220341956
HIP 23693  ;05 05 30.6561782324 -57 28 21.728931803
HIP 24186  ;05 11 40.5893175460 -45 01 06.353955874
HIP 24813  ;05 19 08.4754630747 +40 05 56.589643877
HIP 25110  ;05 22 33.5290089287 +79 13 52.142657543
HIP 25278  ;05 24 25.4629773342 +17 23 00.729080928
HIP 25878  ;05 31 27.3958475290 -03 40 38.021551815
HIP 26394  ;05 37 09.8851202601 -80 28 08.831347245
HIP 26779  ;05 41 20.3357283721 +53 28 51.810629854
HIP 27072  ;05 44 27.7908940 -22 26 54.180763
HIP 27435  ;05 48 34.9401453172 -04 05 40.719529406
HIP 28103  ;05 56 24.2930042 -14 10 03.718884
HIP 29271  ;06 10 14.4741194137 -74 45 10.963587125
HIP 29295  ;06 10 34.6152510171 -21 51 52.658021185
HIP 29525  ;06 13 12.5027477683 +10 37 37.713412362
HIP 29568  ;06 13 45.2955618090 -23 51 42.968779940
HIP 29650  ;06 14 50.8755168504 +19 09 23.210642162
HIP 29800  ;06 16 26.6187785712 +12 16 19.790924635
HIP 30503  ;06 24 43.8797456693 -28 46 48.416253741
HIP 32439  ;06 46 14.1490160697 +79 33 53.316713406
HIP 32480  ;06 46 44.3375552599 +43 34 38.742949290
HIP 32984  ;06 52 18.0504546991 -05 10 25.366165190
HIP 33226  ;06 54 48.9577150656 +33 16 05.438570309
HIP 33277  ;06 55 18.6667052671 +25 22 32.503802377
HIP 33817  ;07 01 13.7244998356 -25 56 55.467043102
HIP 34017  ;07 03 30.4588797095 +29 20 13.496987787
HIP 34065  ;07 03 57.3152423334 -43 36 28.921620874
HIP 35136  ;07 15 50.1392154188 +47 14 23.875808075
HIP 36208  ;07 27 24.49975 +05 13 32.8332
HIP 36366  ;07 29 06.71887 +31 47 04.3773
HIP 36439  ;07 29 55.9568245403 +49 40 20.861583932
HIP 38784  ;07 56 17.2280848677 +80 15 55.954609278
HIP 38908  ;07 57 46.9145300342 -60 18 11.051586167
HIP 39903  ;08 09 00.5695810134 -61 18 08.583553860
HIP 40035  ;08 10 39.8257869468 -13 47 57.138604864
HIP 40693  ;08 18 23.9469692487 -12 37 55.810202572
HIP 40843  ;08 20 03.8606964664 +27 13 03.746404123
HIP 41926  ;08 32 51.4958311521 -31 30 03.062906467
HIP 42438  ;08 39 11.7043265307 +65 01 15.268273190
HIP 42808  ;08 43 18.0304005386 -38 52 56.570025600
HIP 43587  ;08 52 35.8113282132 +28 19 50.956901366
HIP 43726  ;08 54 17.9469808897 -05 26 04.046487788
HIP 43797  ;08 55 11.7820589468 -54 57 56.762642209
HIP 44075  ;08 58 43.9328634171 -16 07 57.805389219
HIP 44897  ;09 08 51.0705427688 +33 52 55.988119240
HIP 45333  ;09 14 20.5373862843 +61 25 23.953809117
HIP 45343  ;09 14 22.7754519714 +52 41 11.792840757
HIP 120005 ;09 14 24.6830609365 +52 41 10.906120364
HIP 46509  ;09 29 08.9408976854 -02 46 08.208984804
HIP 47080  ;09 35 39.5018053382 +35 48 36.484070234
HIP 47592  ;09 42 14.4165254340 -23 54 56.054079074
HIP 48113  ;09 48 35.3713005275 +46 01 15.633773390
HIP 49081  ;10 01 00.6567799273 +31 55 25.216848499
HIP 49908  ;10 11 22.1400209197 +49 27 15.249158750
HIP 49986  ;10 12 17.6684359061 -03 44 44.393843397
HIP 50384  ;10 17 14.53796 +23 06 22.3876
HIP 50564  ;10 19 44.1668778 +19 28 15.294314
HIP 50954  ;10 24 23.7059714 -74 01 53.803578
HIP 51317  ;10 28 55.5513010331 +00 50 27.601817738
HIP 51459  ;10 30 37.5803678549 +55 58 49.936416690
HIP 51502  ;10 31 04.5420225146 +82 33 31.255563167
HIP 51523  ;10 31 21.8213012849 -53 42 55.737315449
HIP 53721  ;10 59 27.9738644892 +40 25 48.922388918
HIP 54035  ;11 03 20.19400 +35 58 11.5682
HIP 54211  ;11 05 28.57798 +43 31 36.3914
HIP 55846  ;11 26 45.3217209044 +03 00 47.158549988
HIP 56452  ;11 34 29.4862840828 -32 49 52.819893272
HIP 56997  ;11 41 03.0159358291 +34 12 05.882438337
HIP 57443  ;11 46 31.0719919710 -40 30 01.279976346
HIP 57507  ;11 47 15.8077490164 -30 17 11.437355211
HIP 57548  ;11 47 44.3968668170 +00 48 16.404931305
HIP 57757  ;11 50 41.7182390 +01 45 52.991019
HIP 57939  ;11 52 58.7683801554 +37 43 07.240082865
HIP 58345  ;11 57 56.2063624042 -27 42 25.364242481
HIP 58576  ;12 00 44.4611593182 -10 26 46.055023587
HIP 59199  ;12 08 24.8165241 -24 43 43.950354
HIP 61174  ;12 32 04.2265302 -16 11 45.616473
HIP 61317  ;12 33 44.5448195 +41 21 26.924857
HIP 62207  ;12 44 59.4050673140 +39 16 44.098316643
HIP 62523  ;12 48 47.0482457374 +24 50 24.820252955
HIP 62951  ;12 53 58.8003153880 -18 02 05.677093667
HIP 63721  ;13 03 29.0648355662 +25 47 47.866025476
HIP 64394  ;13 11 52.3937856 +27 52 41.453553
HIP 64583  ;13 14 15.1459391643 -59 06 11.652757891
HIP 64792  ;13 16 46.5161591991 +09 25 26.967205938
HIP 64797  ;13 16 51.05313 +17 01 01.8441
HIP 64924  ;13 18 24.3142756 -18 18 40.304648
HIP 65721  ;13 28 25.8086 +13 46 43.637
HIP 65859  ;13 29 59.7856884759 +10 22 37.784822214
HIP 67155  ;13 45 43.7754528655 +14 53 29.473195472
HIP 67275  ;13 47 15.74340 +17 27 24.8552
HIP 68184  ;13 57 32.0591733968 +61 29 34.299358045
HIP 68682  ;14 03 32.3512374876 +10 47 12.439942066
HIP 69671  ;14 15 38.6854180858 -45 00 02.728009204
HIP 69965  ;14 19 00.8956424822 -25 48 55.531887272
HIP 69972  ;14 19 04.8341367419 -59 22 44.535027477
HIP 70319  ;14 23 15.2847534665 +01 14 29.641784590
HIP 70497  ;14 25 11.7970287 +51 51 02.676894
HIP 70890  ;14 29 42.9451234609 -62 40 46.170818907
HIP 71284  ;14 34 40.8171808483 +29 44 42.463685978
HIP 72567  ;14 50 15.8107686483 +23 54 42.633203114
HIP 72603  ;14 50 41.1746746757 -15 59 50.028744741
HIP 72659  ;14 51 23.37993 +19 06 01.6994
HIP 72848  ;14 53 23.7667403 +19 09 10.081308
HIP 73184  ;14 57 28.0008928545 -21 24 55.726897561
HIP 73996  ;15 07 18.0658711040 +24 52 09.095241798
HIP 75181  ;15 21 48.1511171112 -48 19 03.462483973
HIP 76074  ;15 32 12.9326267337 -41 16 32.131578815
HIP 76829  ;15 41 11.3769921 -44 39 40.342813
HIP 77052  ;15 44 01.8189306784 +02 30 54.600773948
```

```
HIP 77257  ;15 46 26.6144291449 +07 21 11.041647444
HIP 77358  ;15 47 29.1006767984 -37 54 58.722306957
HIP 77760  ;15 52 40.5411372117 +42 27 05.451105113
HIP 77801  ;15 53 12.0969759597 +13 11 47.842699872
HIP 78072  ;15 56 27.1826948 +15 39 41.820500
HIP 78459  ;16 01 02.6608052290 +33 18 12.642245533
HIP 78775  ;16 04 56.7938239883 +39 09 23.434791067
HIP 79248  ;16 10 24.3152754550 +43 49 03.498734567
HIP 79537  ;16 13 48.5584718157 -57 34 13.843908786
HIP 79672  ;16 15 37.2703721200 -08 22 09.981989277
HIP 80337  ;16 24 01.2905970257 -39 11 34.734611237
HIP 80459  ;16 25 24.6233091302 +54 18 14.765751243
HIP 80686  ;16 28 28.1396465547 -70 05 03.822057572
HIP 80824  ;16 30 18.0582010683 -12 39 45.323235188
HIP 81300  ;16 36 21.4492997704 -02 19 28.512485729
HIP 82588  ;16 52 58.8025427362 -00 01 35.116299669
HIP 82860  ;16 56 01.6892483 +65 08 05.263139
HIP 83541  ;17 04 27.8432064954 -28 34 57.639798036
HIP 83601  ;17 05 16.8186294192 +00 42 09.217571716
HIP 83609  ;17 05 20.7263053042 -33 46 00.025833174
HIP 84478  ;17 16 13.3627374963 -26 32 46.133091527
HIP 84720  ;17 19 03.83574 -46 38 10.4467
HIP 84862  ;17 20 39.5675395123 +32 28 03.877348066
HIP 84893  ;17 21 00.3752009 -21 06 46.566283
HIP 85042  ;17 22 51.2877095327 -02 23 17.439826176
HIP 85235  ;17 25 00.0982708577 +67 18 24.150141561
HIP 85295  ;17 25 45.2323043669 +02 06 41.123668100
HIP 85523  ;17 28 39.9455601300 -46 53 42.693246243
HIP 86162  ;17 36 25.8991699744 +68 20 20.904135942
HIP 86400  ;17 39 16.9163262 +03 33 18.875718
HIP 86486  ;17 40 23.8255179884 -49 24 56.104103575
HIP 86620  ;17 41 58.1041581273 +72 09 24.836647212
HIP 86736  ;17 43 25.7937012 -21 40 59.497954
HIP 86796  ;17 44 08.7036342277 -51 50 02.591049123
HIP 86974  ;17 46 27.5266778 +27 43 14.437984
HIP 88574  ;18 05 07.5787546501 -03 01 52.753216025
HIP 88601  ;18 05 27.28518 +02 30 00.3558
HIP 88694  ;18 06 23.7191233831 -36 01 11.229463639
HIP 88745  ;18 07 01.53971 +30 33 43.6896
HIP 88972  ;18 09 37.4162810870 +38 27 27.995921559
HIP 89042  ;18 10 26.1511772984 -62 00 08.059294747
HIP 89348  ;18 13 53.8327884671 +64 23 50.225272101
HIP 89805  ;18 19 40.13138 -63 53 11.6282
HIP 90790  ;18 31 18.9612203327 -18 54 31.732564617
HIP 91768  ;18 42 46.7048533410 +59 37 49.411785651
HIP 91772  ;18 42 46.8942178098 +59 37 36.723092516
HIP 93858  ;19 06 52.4643138830 -37 48 38.372801262
HIP 94761  ;19 16 55.2565250393 +05 10 08.038856129
HIP 95149  ;19 21 29.7277723051 -34 59 00.356331012
HIP 95319  ;19 23 34.0133626629 +33 13 19.074920892
HIP 95447  ;19 24 58.2002406780 +11 56 39.882312617
HIP 96100  ;19 32 21.5902990 +69 39 40.234737
HIP 96441  ;19 36 26.5343563 +50 13 15.964573
HIP 96895  ;19 41 48.9539315338 +50 31 30.218780803
HIP 97295  ;19 46 25.5997765489 +33 43 39.342745085
HIP 97675  ;19 51 01.6437560393 +10 24 56.595177367
HIP 97944  ;19 54 17.7452778997 -23 56 27.862976419
HIP 98036  ;19 55 18.7925630 +06 24 24.342501
HIP 98470  ;20 00 20.2490835420 -33 42 12.427710024
HIP 98767  ;20 03 37.4049065078 +29 53 48.495330996
HIP 98959  ;20 05 32.7652367983 -67 19 15.228862402
HIP 99240  ;20 08 43.6094697 -66 10 55.443275
HIP 99461  ;20 11 11.9384853308 -36 06 04.353559455
HIP 99701  ;20 13 53.3963903475 -45 09 50.473460317
HIP 99825  ;20 15 17.3916558603 -27 01 58.713584596
HIP 100017 ;20 17 31.3282807419 +66 51 13.281593047
HIP 100925 ;20 27 44.2428093744 -30 52 04.246660714
HIP 101997 ;20 40 11.7546336973 -23 46 25.924674471
HIP 102040 ;20 40 45.1407864181 +19 56 07.928511642
HIP 102485 ;20 46 05.7326265 -25 16 15.231155
HIP 103096 ;20 53 19.7890710784 +62 09 15.813717444
HIP 103389 ;20 56 47.3303687041 -26 17 46.969399809
HIP 104214 ;21 06 53.9396100677 +38 44 57.897024357
HIP 104217 ;21 06 55.2640651855 +38 44 31.362140913
HIP 105090 ;21 17 15.2688576495 -38 52 02.510022611
HIP 105858 ;21 26 26.6048372 -65 21 58.314484
HIP 106440 ;21 33 33.9749932664 -49 00 32.403471949
HIP 107350 ;21 44 31.3299733695 +14 46 18.982331039
HIP 107649 ;21 48 15.7510673466 -47 18 13.020109072
HIP 108870 ;22 03 21.6542294981 -56 47 09.537018193
HIP 109176 ;22 07 00.6620572743 +25 20 42.376138495
HIP 109422 ;22 10 08.7803395485 -32 32 54.275502381
HIP 110649 ;22 24 56.3720716646 -57 47 50.824747845
HIP 111449 ;22 34 41.6367033191 -20 42 29.574530725
HIP 112447 ;22 46 41.5811758 +12 10 22.385447
HIP 112460 ;22 46 49.7311740821 +44 20 02.372223230
HIP 113020 ;22 53 16.7323107416 -14 15 49.303409936
HIP 113283 ;22 56 24.0532946025 -31 33 56.035064606
HIP 113357 ;22 57 27.9804167474 +20 46 07.782240714
HIP 113421 ;22 58 15.5411942071 -02 23 43.387117210
HIP 113576 ;23 00 16.1224771183 -22 31 27.651428500
HIP 114046 ;23 05 52.0354545522 -35 51 11.058757520
HIP 114622 ;23 13 16.9747821012 +57 10 06.076520993
HIP 114924 ;23 16 42.3028134294 +53 12 48.514282989
HIP 114948 ;23 16 57.6873978547 -62 00 04.318777045
HIP 116085 ;23 31 22.2087185647 +59 09 55.866485371
HIP 116745 ;23 39 37.3871312596 -72 43 19.757339908
HIP 116771 ;23 39 57.0413764 +05 37 34.647529
HIP 117473 ;23 49 12.5251391270 +02 24 04.403044891
HIP 117712 ;23 52 25.4066877164 +75 32 40.368591630
```

C.2 Bright Stars

The celestial coordinates (hour angle and declination) of the stars of apparent magnitude 2 and brighter, as sourced from SIMBAD.

`bright_stars_simbad_trim.csv`

```
05 23 34.6 -69 45 22
05 40 45.527 -01 56 33.26
01 37 42.84548 -57 14 12.3101
12 54 01.7495922 +55 57 35.362645
12 47 43.26877 -59 41 19.5792
07 08 23.4840514 -26 23 35.518484
14 39 35.06311 -60 50 15.0992
05 40 45.52666 -01 56 33.2649
08 09 31.95013 -47 20 11.7108
05 25 07.86325 +06 20 58.9318
09 13 11.97746 -69 43 01.9473
07 12 36.0 -27 40 00
07 46 53.081 +39 00 52.52
06 37 42.71050 +16 23 57.4095
05 14 32.27210 -08 12 05.8981
14 39 36.49400 -60 50 02.3737
12 26 35.896 -63 05 56.73
17 33 36.52012 -37 06 13.7648
20 41 25.91514 +45 16 49.2197
07 46 54.220 +39 00 21.25
10 19 58.354462 +19 50 29.35920
12 31 09.95961 -57 06 47.5684
07 46 58.210 +39 00 51.24
05 59 31.7229284 +44 56 50.757259
19 50 46.99855 +08 52 05.9563
14 39 36.204 -60 50 08.23
18 36 56.33635 +38 47 01.2802
22 57 39.04625 -29 37 20.0533
13 47 32.43776 +49 18 47.7602
17 23 41.71 +30 29 50.6
08 22 30.83526 -59 30 34.1431
07 46 53.320 +39 00 18.25
04 35 55.23907 +16 30 33.4885
06 45 08.91728 -16 42 58.0171
17 37 19.12985 -42 59 52.1808
07 45 18.94987 +28 01 34.3160
10 08 22.31099 +11 58 01.9516
12 26 36.442 -63 05 58.28
14 03 49.404440 -60 22 22.94186
05 16 41.35871 +45 59 52.7693
16 29 24.45970 -26 25 55.2094
20 25 38.85705 -56 44 06.3230
22 08 13.98473 -46 57 39.5078
05 36 12.81335 -01 12 06.9089
06 45 08.917 -16 42 58.02
07 34 35.863 +31 53 17.79
03 24 19.3700924 +49 51 40.245455
14 15 39.67207 +19 10 56.6730
06 58 37.54876 -28 58 19.5102
05 55 10.30536 +07 24 25.4304
06 22 41.9853527 -17 57 21.307352
07 34 35.87319 +31 53 17.8160
05 26 17.51312 +28 36 26.8262
07 46 54.220 +39 01 18.25
13 25 11.57937 -11 09 40.7501
22 47 27.89448 +58 07 23.6820
17 37 19.131 -42 59 52.17
08 44 42.22658 -54 42 31.7493
07 46 53.970 +39 01 06.25
16 48 39.89508 -69 01 39.7626
14 03 49.40535 -60 22 22.9266
09 27 35.24270 -08 39 30.9583
07 39 18.11950 +05 13 29.9552
11 03 43.67152 +61 45 03.7249
06 23 57.10988 -52 41 44.3810
18 24 10.31840 -34 23 04.6193
```

C.3 Observations

This is the list of observations baselined for LUVOIR, per Stark *et al.* 2015 [57], received from Chris Stark via personal communication. Each row consists of the star ID (Hipparcos Catalogue), the count of how many times the star has been observed (including the current one, *i.e.* starting at 1), the desired number of years since the first observation of that star, and the duration of the observation in days.

LUVOIR-Architecture_A-NOMINAL_OCCRATES-observations-trim.csv

```
108870,1,0,0.287309
104214,1,0,0.283274
19849,1,0,0.281086
54035,1,0,0.287174
15510,1,0,0.272661
104217,1,0,0.307915
114046,1,0,0.296112
96100,1,0,0.294242
105090,1,0,0.308159
73184,1,0,0.332606
49908,1,0,0.335085
61317,1,0,0.269823
1599,1,0,0.268259
84478,1,0,0.331173
1475,1,0,0.32297
99461,1,0,0.334955
114622,1,0,0.344392
64394,1,0,0.270016
105858,1,0,0.267811
3765,1,0,0.348172
64924,1,0,0.31245
12114,1,0,0.350594
7751,1,0,0.353935
16537,1,0,0.248323
15457,1,0,0.322745
57443,1,0,0.327535
7981,1,0,0.350556
113283,1,0,0.360968
84720,1,0,0.375995
63721,1,0,0.307463
99825,1,0,0.376448
56452,1,0,0.366809
91772,1,0,0.352881
29271,1,0,0.353868
86162,1,0,0.337981
10644,1,0,0.323017
56997,1,0,0.383947
70890,1,0,0.379192
8102,1,0,0.238586
88601,1,0,0.309483
23311,1,0,0.38509
81300,1,0,0.396736
8362,1,0,0.395069
23693,1,0,0.308712
32984,1,0,0.377723
14632,1,0,0.261069
72659,1,0,0.404621
80686,1,0,0.333923
13402,1,0,0.398152
12777,1,0,0.265108
27072,1,0,0.241134
3821,1,0,0.235953
85523,1,0,0.375158
68184,1,0,0.395846
57939,1,0,0.361294
99240,1,0,0.241339
77257,1,0,0.289611
15330,1,0,0.41951
36208,1,0,0.395047
3093,1,0,0.423569
72848,1,0,0.412738
45343,1,0,0.385149
88972,1,0,0.405776
51459,1,0,0.328894
24813,1,0,0.316676
47080,1,0,0.4362
64797,1,0,0.396933
15371,1,0,0.445264
42808,1,0,0.38504
41926,1,0,0.389346
78072,1,0,0.251129
86400,1,0,0.41743
40693,1,0,0.430989
80337,1,0,0.455107
10798,1,0,0.40478
26779,1,0,0.432715
25878,1,0,0.397967
120005,1,0,0.402678
94761,1,0,0.360449
22263,1,0,0.450719
43587,1,0,0.45974
117473,1,0,0.355723
58345,1,0,0.411353
85235,1,0,0.415489
58576,1,0,0.4824
69972,1,0,0.418291
82860,1,0,0.330239
10138,1,0,0.445339
109176,1,0,0.246461
47592,1,0,0.347382
12843,1,0,0.28986
544,1,0,0.454649
5862,1,0,0.354734
29295,1,0,0.39596
79672,1,0,0.489218
24186,1,0,0.345354
105858,2,1.01724,0.261734
1599,2,0.935237,0.260386
```

```
76074,1,0,0.388955
64394,2,1.00688,0.265323
42438,1,0,0.497364
84862,1,0,0.494575
61317,2,0.911963,0.26078
57757,1,0,0.242968
16852,1,0,0.276378
80824,1,0,0.443085
7513,1,0,0.26447
99701,1,0,0.38925
116771,1,0,0.267416
75181,1,0,0.521747
27072,2,1.23515,0.23919
53721,1,0,0.445572
3583,1,0,0.500943
85295,1,0,0.430794
25278,1,0,0.437287
14632,2,1.33144,0.260536
27435,1,0,0.504808
12777,2,1.3581,0.262945
117712,1,0,0.49157
3909,1,0,0.468201
101997,1,0,0.459937
77358,1,0,0.500423
71284,1,0,0.285117
49081,1,0,0.527914
107649,1,0,0.499384
88745,1,0,0.450312
78072,2,1.41411,0.249056
106440,1,0,0.372038
38908,1,0,0.513021
77760,1,0,0.305009
62951,1,0,0.417408
77052,1,0,0.5304
102485,1,0,0.265403
86974,1,0,0.248049
113357,1,0,0.536784
77257,2,1.28747,0.283427
15510,2,0.686794,0.244463
83609,1,0,0.398241
98767,1,0,0.508383
86796,1,0,0.462937
23693,2,1.0239,0.302723
95447,1,0,0.47914
35136,1,0,0.49761
95319,1,0,0.476392
439,1,0,0.316517
113020,1,0,0.470363
3821,2,0.923223,0.231018
7981,2,0.494539,0.263352
63721,2,0.0205623,0.289695
97944,1,0,0.538366
22449,1,0,0.230071
64924,2,0.799815,0.279279
109176,2,1.54814,0.246244
88601,2,0.245413,0.270366
34065,1,0,0.521634
70497,1,0,0.260704
99240,2,0.952997,0.234232
32480,1,0,0.501204
78775,1,0,0.446782
10644,2,0.883751,0.300433
99461,2,0.403789,0.246728
7978,1,0,0.521441
57757,2,1.41536,0.241918
62207,1,0,0.492487
19076,1,0,0.530805
33277,1,0,0.51482
80686,2,0.992291,0.321531
24813,2,1.16942,0.312446
59199,1,0,0.258692
100017,1,0,0.515059
12653,1,0,0.526788
78459,1,0,0.519206
7751,2,0.45474,0.270132
56997,2,0.612957,0.297468
57443,2,0.28758,0.296207
43726,1,0,0.493179
12114,2,0.392664,0.25893
51459,2,1.08223,0.313439
107350,1,0,0.48514
114622,2,0.421855,0.260216
91772,2,0.0224414,0.309339
3765,2,0.416885,0.263602
32439,1,0,0.517121
84720,2,0.511269,0.276939
33226,1,0,0.466903
93858,1,0,0.50626
86162,2,0.0337295,0.318728
77801,1,0,0.496697
91768,1,0,0.299609
99825,2,0.519176,0.296893
84478,2,0.287978,0.254505
15457,2,0.293851,0.306016
16537,2,0.126501,0.248036
89042,1,0,0.537312
29568,1,0,0.493956
76829,1,0,0.302581
56452,2,0.479852,0.295789
88694,1,0,0.524859
17378,1,0,0.243648
12843,2,1.47677,0.288203
98959,1,0,0.488198
46509,1,0,0.301578
64792,1,0,0.484355
70319,1,0,0.511524
73184,2,0.343498,0.246195
8362,2,0.614296,0.312373
62523,1,0,0.522658
81300,2,0.549817,0.301627
69965,1,0,0.544422
7513,2,1.55401,0.264789
23311,2,0.429375,0.29373
8102,2,0.576285,0.230802
68682,1,0,0.515639
16852,2,1.63184,0.275947
116771,2,1.54647,0.265467
113283,2,0.325525,0.275452
96100,2,0.241784,0.256374
29271,2,0.821086,0.31638
26394,1,0,0.542257
72659,2,0.669996,0.312252
40843,1,0,0.464369
1475,2,0.0563663,0.295632
114046,2,0.056646,0.254391
19849,2,0.51282,0.229
57939,2,0.124324,0.3334
13402,2,0.470668,0.314424
104214,2,0.0921989,0.24943
54035,2,0.0417176,0.243678
49908,2,0.226519,0.243881
36208,2,0.0239883,0.336401
105090,2,0.186514,0.240245
82860,2,1.27898,0.326647
86736,1,0,0.412895
```

```
45343,2,0.0812126,0.333691
85523,2,0.0861027,0.330459
15330,2,0.687956,0.354485
32984,2,0.12973,0.319837
18859,1,0,0.53028
47592,2,1.23524,0.335897
65721,1,0,0.364877
72567,1,0,0.512848
4151,1,0,0.33037
33817,1,0,0.504598
72848,2,0.543319,0.348585
3093,2,0.557556,0.349506
15371,2,0.805188,0.365335
42808,2,0.157121,0.342445
109422,1,0,0.436412
5862,2,1.22773,0.342462
112447,1,0,0.272542
117473,2,0.096795,0.321111
47080,2,0.801144,0.364216
94761,2,0.0409875,0.34642
25878,2,0.075789,0.365196
104217,2,0.202454,0.226041
120005,2,0.0885721,0.360153
71284,2,1.65225,0.28552
102485,2,1.6634,0.263533
41926,2,0.183966,0.347469
70890,2,0.0104425,0.310202
64797,2,0.164738,0.348478
88972,2,0.169857,0.335572
68184,2,0.15904,0.330091
48113,1,0,0.472037
29800,1,0,0.434222
38784,1,0,0.485325
24186,2,0.0462832,0.330924
86400,2,0.162925,0.350314
44897,1,0,0.559038
34017,1,0,0.555441
70497,2,1.5473,0.260565
40693,2,0.594656,0.374474
77760,2,1.60274,0.301685
22263,2,0.794425,0.378825
58345,2,0.168735,0.367252
80337,2,0.810337,0.396304
29525,1,0,0.465457
67155,1,0,0.345049
108870,2,0.159081,0.243134
29295,2,0.0733788,0.369499
97675,1,0,0.472063
57507,1,0,0.478183
10798,2,0.19389,0.374255
86974,2,1.33981,0.247633
26779,2,0.52205,0.364666
59199,2,1.57807,0.258431
22449,2,1.36707,0.229572
116745,1,0,0.45707
910,1,0,0.428435
85235,2,0.186144,0.368298
36439,1,0,0.524608
43587,2,0.614463,0.388867
99701,2,0.076072,0.354604
45333,1,0,0.472097
53721,2,0.990708,0.383971
58576,2,0.770102,0.394286
113576,1,0,0.445019
69972,2,0.172865,0.38432
65859,1,0,0.346812
25278,2,1.01774,0.399058
114948,1,0,0.556759
10138,2,0.48612,0.395822
73996,1,0,0.438534
114924,1,0,0.52862
82588,1,0,0.481385
76074,2,0.101221,0.362143
84862,2,0.925033,0.390939
79672,2,0.841008,0.399025
50954,1,0,0.256836
90790,1,0,0.453855
544,2,0.625035,0.407895
17378,2,1.17426,0.24244
29650,1,0,0.472944
79248,1,0,0.475815
88745,2,1.2121,0.413688
42438,2,0.809544,0.420043
76829,2,1.71169,0.302158
85042,1,0,0.509825
46509,2,1.74712,0.298422
106440,2,0.0528322,0.349613
28103,1,0,0.244978
55846,1,0,0.515275
75181,2,0.81841,0.380223
51502,1,0,0.479891
3909,2,1.13869,0.42832
950,1,0,0.480503
96441,1,0,0.31353
439,2,0.0460831,0.307986
85295,2,0.11101,0.413505
83601,1,0,0.573866
44075,1,0,0.561584
110649,1,0,0.497046
113421,1,0,0.493946
86796,2,1.07913,0.414218
95447,2,1.1695,0.401376
25110,1,0,0.445736
95149,1,0,0.464092
80824,2,0.0212749,0.410272
57548,1,0,0.509294
49081,2,1.00063,0.425583
116085,1,0,0.492402
112447,2,1.62782,0.270548
17651,1,0,0.271671
12444,1,0,0.505504
19335,1,0,0.579666
27435,2,0.718265,0.402312
77358,2,0.723751,0.385808
3583,2,0.784805,0.453885
107649,2,0.946343,0.44749
77052,2,0.737362,0.385226
91768,2,0.0333147,0.284724
117712,2,0.493981,0.420537
83541,1,0,0.488139
38908,2,0.948802,0.441634
61174,1,0,0.279366
103389,1,0,0.514069
83609,2,0.0203872,0.393578
62951,2,0.0357359,0.392856
98470,1,0,0.522834
54211,1,0,0.326936
98767,2,0.907949,0.429769
4151,2,1.74206,0.324739
97295,1,0,0.446683
32480,2,1.19373,0.452713
113357,2,0.996879,0.471112
34065,2,1.0181,0.418652
39903,1,0,0.393675
28103,2,1.63483,0.244851
```

```
101997,2,0.228338,0.444394
35136,2,1.02236,0.470729
40035,1,0,0.547812
49986,1,0,0.364675
86736,2,1.38662,0.412492
65721,2,1.61264,0.354853
50954,2,1.55951,0.256717
78775,2,0.235272,0.446314
7978,2,1.05716,0.451834
64792,2,1.30468,0.420652
32439,2,1.14486,0.426785
12653,2,1.14075,0.446127
33277,2,0.936804,0.437033
78459,2,1.03472,0.411603
62207,2,0.848625,0.437677
69671,1,0,0.510138
19076,2,0.835474,0.435158
64583,1,0,0.341991
80459,1,0,0.435894
40843,2,1.24722,0.441921
86620,1,0,0.546745
98036,1,0,0.250678
89042,2,1.12134,0.421284
100017,2,0.87727,0.424125
109422,2,1.38818,0.423558
96895,1,0,0.519502
112460,1,0,0.446882
43726,2,0.82894,0.430529
97944,2,0.756853,0.452449
5336,1,0,0.521079
107350,2,0.867583,0.43992
113020,2,0.0333749,0.437147
36366,1,0,0.271073
19859,1,0,0.488642
95319,2,0.252519,0.435401
77801,2,0.811697,0.419521
84893,1,0,0.376625
88694,2,0.864825,0.428065
93858,2,0.756561,0.421411
111449,1,0,0.492231
98959,2,0.826579,0.432611
910,2,1.3081,0.398714
88601,3,0.9203,0.257341
29800,2,1.38966,0.426949
26394,2,1.06065,0.432385
48113,2,1.33144,0.430705
51317,1,0,0.440805
103096,1,0,0.433797
69965,2,0.908226,0.433419
33226,2,0.0830554,0.422152
102040,1,0,0.498666
88574,1,0,0.365602
17651,2,1.75542,0.271497
50384,1,0,0.52862
68682,2,0.788128,0.416177
98036,2,1.70761,0.251157
72603,1,0,0.471069
73996,2,1.33906,0.400997
18859,2,1.25833,0.479392
67155,2,0.0694365,0.323158
96441,2,1.77375,0.311287
79537,1,0,0.307262
97675,2,1.35591,0.443778
29568,2,0.265219,0.421971
72567,2,0.934133,0.442445
100925,1,0,0.490093
1803,1,0,0.499305
61174,2,1.74327,0.277071
1292,1,0,0.485848
45333,2,1.35524,0.458058
70319,2,0.284431,0.420341
33817,2,0.267202,0.406141
30503,1,0,0.47761
50564,1,0,0.407686
3821,3,1.20019,0.228939
21770,1,0,0.286311
65859,2,0.140357,0.316912
7339,1,0,0.519697
67275,1,0,0.542536
62523,2,0.278984,0.465474
86486,1,0,0.403112
89805,1,0,0.498628
89348,1,0,0.443271
99240,3,1.2389,0.23303
51523,1,0,0.42797
44897,2,0.950103,0.446347
36439,2,1.38321,0.495221
39903,2,1.33671,0.38931
36366,2,1.60764,0.270571
43797,1,0,0.531987
116745,2,0.159274,0.424565
34017,2,0.959621,0.488094
29650,2,1.39746,0.469423
113576,2,0.101204,0.413181
114924,2,1.26074,0.457136
38784,2,0.307961,0.469719
25110,2,1.51447,0.441355
114948,2,1.21053,0.440794
51502,2,1.40856,0.466118
950,2,1.40469,0.470458
57507,2,0.328952,0.4229
29525,2,0.272803,0.464771
97295,2,1.42775,0.446251
64583,2,1.56894,0.341275
110649,2,1.35551,0.480724
27072,3,2.19582,0.238647
90790,2,0.205027,0.426551
82588,2,0.28597,0.487804
79248,2,0.739098,0.4543
113421,2,0.92681,0.471817
57443,3,1.07843,0.276402
54211,2,0.0578372,0.312249
84893,2,1.75421,0.379272
85042,2,0.837287,0.414341
15457,3,1.10194,0.275594
44075,2,1.16264,0.498911
10644,3,1.23725,0.301787
17378,3,1.67752,0.242958
83601,2,1.01003,0.429614
19335,2,1.34951,0.473705
12444,2,1.19495,0.423343
8102,3,0.733453,0.229955
94761,3,0.081975,0.334639
103389,2,1.26564,0.495983
98470,2,1.31072,0.506115
50564,2,1.41934,0.397226
86486,2,1.44116,0.396398
21770,2,1.79765,0.282917
22449,3,2.27844,0.229568
45343,3,0.162425,0.306485
68184,3,0.556641,0.319894
23693,3,1.43346,0.298457
95149,2,0.348955,0.46072
40035,2,1.42677,0.495584
116085,2,0.282084,0.431768
```

```
55846,2,0.282856,0.456163
49986,2,0.12591,0.361804
32984,3,0.486488,0.296452
88972,3,0.594498,0.32044
64797,3,0.617769,0.335019
57757,3,2.6538,0.241335
41926,3,0.689874,0.374448
29271,3,1.11966,0.287333
113283,3,0.443897,0.269617
86162,3,0.060713,0.290768
51459,3,1.51512,0.320256
111449,2,1.54187,0.46284
57939,3,0.279729,0.310105
117473,3,0.123194,0.317088
51523,2,1.40515,0.422844
58345,3,0.303723,0.336619
5336,2,0.490042,0.501293
72603,2,1.61553,0.470688
64924,3,1.09066,0.274288
86974,3,1.78642,0.247545
89348,2,1.56926,0.446288
96895,2,1.06601,0.444748
42808,3,0.314243,0.320954
85523,3,0.114804,0.32666
86400,3,0.61097,0.336894
86620,2,1.25818,0.452902
79537,2,0.277545,0.292678
16537,3,0.506005,0.214274
29295,3,0.146758,0.336989
10798,3,0.727089,0.357188
63721,3,0.0308434,0.25804
83541,2,0.334429,0.451836
91772,3,0.0403945,0.282026
112460,2,0.0332761,0.425258
57548,2,0.0219921,0.458526
120005,3,0.177144,0.319399
69671,2,0.880143,0.428332
88574,2,0.127174,0.359845
103096,2,0.0878813,0.402694
19859,2,0.964353,0.430998
105858,3,1.42413,0.259018
50384,2,1.24338,0.457297
25878,3,0.13642,0.313495
36208,3,0.043179,0.317769
109176,3,2.75224,0.246223
80686,3,1.38921,0.323569
85295,3,0.222019,0.365955
80459,2,0.0413867,0.417976
85235,3,0.372289,0.341528
64394,3,1.40964,0.262003
1599,3,1.30933,0.26074
78072,3,2.51397,0.248998
99701,3,0.152144,0.330407
67275,2,1.63648,0.46239
61317,3,1.27675,0.259858
51317,2,0.0439498,0.421851
69972,3,0.388946,0.343035
30503,2,0.983024,0.458623
24813,3,1.63718,0.303805
43797,2,1.44331,0.531565
1803,2,0.941629,0.446038
102040,2,0.895662,0.405468
89805,2,1.18019,0.49761
62951,3,0.0643246,0.380482
76074,3,0.147231,0.38037
56452,3,0.654343,0.290134
23311,3,0.585511,0.287712
80824,3,0.0850996,0.38159
77257,3,1.67371,0.285895
13402,3,0.658936,0.332321
26779,3,0.73087,0.363401
70497,3,2.90118,0.262431
14632,3,2.66288,0.258637
59199,3,2.95888,0.256283
1292,2,0.250672,0.458953
100925,2,0.345438,0.43389
106440,3,0.116231,0.321235
7513,3,3.2807,0.264872
101997,3,0.856269,0.391398
116771,3,3.26478,0.265607
24186,3,0.0848525,0.276725
12777,3,1.90134,0.261882
47592,3,1.72934,0.335153
3821,4,1.93877,0.226617
7339,2,0.309436,0.499051
99240,4,1.90599,0.229755
12843,3,1.9198,0.285836
16852,3,3.26368,0.276162
5862,3,1.71882,0.340101
102485,3,2.95716,0.264993
82860,3,1.66268,0.32414
95319,3,0.946945,0.436193
84478,3,0.392697,0.248631
83609,3,0.0407745,0.356995
78775,3,0.470544,0.339473
71284,3,3.30449,0.289635
113020,3,0.0611873,0.428897
1475,3,0.101459,0.257468
86974,4,2.82849,0.247772
33817,3,0.534405,0.391058
22449,4,3.34172,0.228958
112447,3,2.23825,0.275339
72848,3,0.760646,0.333359
33226,3,0.113257,0.466612
29568,3,0.530439,0.383932
70319,3,0.995509,0.417368
12114,3,0.53545,0.267374
439,3,0.0614441,0.299982
96100,3,0.483567,0.238135
72659,3,0.91363,0.308496
7981,3,0.692355,0.263498
7751,3,0.6201,0.264363
81300,3,0.749751,0.287638
49908,3,0.30889,0.236516
40693,3,0.832518,0.365537
77760,3,3.20548,0.298847
50954,3,3.11903,0.260499
46509,3,3.49424,0.297441
76829,3,3.42337,0.302113
28103,3,3.26966,0.244941
10138,3,0.680568,0.382473
57507,3,0.592113,0.387325
3765,3,0.56848,0.256915
544,3,0.875049,0.394372
91768,3,0.0555244,0.28069
62523,3,1.04619,0.417073
99825,3,0.707967,0.294515
3093,3,0.780578,0.338168
67155,3,0.138873,0.304061
99461,3,0.58733,0.245119
38784,3,0.615922,0.361656
90790,3,0.410054,0.384186
8362,3,0.837677,0.301517
17378,4,2.68403,0.241058
15510,3,0.936537,0.233849
```

```
17651,3,3.29142,0.271517
114046,3,0.113292,0.242014
84720,3,0.715777,0.272926
61174,3,2.397,0.278451
16537,4,0.632507,0.2199
19849,3,0.6993,0.224063
29525,3,0.613807,0.379087
8102,4,1.10018,0.212649
95149,3,0.628119,0.460597
113576,3,0.222648,0.387534
116745,3,0.318549,0.378604
43587,3,0.860248,0.38081
82588,3,0.571939,0.374321
77358,3,0.940876,0.426786
56997,3,0.85814,0.298746
55846,3,0.636426,0.457522
15330,3,0.963139,0.354122
83541,3,0.601973,0.441826
4151,3,3.48411,0.321023
105090,3,0.254337,0.228985
36366,3,3.21528,0.271037
117712,3,0.673611,0.342884
27435,3,0.933745,0.388644
68682,3,1.07472,0.370357
65859,3,0.191396,0.311309
439,4,0.0921661,0.275197
116085,3,0.564167,0.384214
96441,3,2.36499,0.318085
54211,3,0.115674,0.317745
98036,3,3.41522,0.250287
93858,3,0.983529,0.431506
80459,3,0.0827734,0.426398
8102,5,1.46691,0.21249
114622,3,0.575256,0.260495
21770,3,2.47178,0.280786
77052,3,0.95857,0.403787
22263,3,1.11219,0.391746
73184,3,0.468406,0.234354
62207,3,1.18808,0.45239
49986,3,0.171696,0.376941
65721,3,3.22528,0.349514
75181,3,1.06393,0.427394
98959,3,1.15721,0.408397
63721,4,0.0411245,0.259395
79248,3,1.00786,0.381964
```

```
47080,3,1.09247,0.355254
54035,3,0.0834351,0.231917
25278,3,2.03547,0.416723
88694,3,1.12427,0.525908
43726,3,1.16052,0.423071
53721,3,1.98142,0.38425
42438,3,1.13336,0.419374
77801,3,1.05521,0.376521
58576,3,1.07814,0.447351
3583,3,1.09873,0.399256
57548,3,0.0408426,0.45852
100017,3,1.22818,0.43535
15371,3,1.12726,0.354763
91768,4,0.0721818,0.259435
64583,3,3.13789,0.341326
39903,3,2.67342,0.389442
910,3,2.45269,0.411936
86736,3,2.4651,0.418776
104217,3,0.294479,0.225255
79672,3,1.17741,0.399195
88574,3,0.173419,0.365141
107350,3,1.12786,0.389275
94761,4,0.122962,0.339275
12843,4,3.10121,0.288872
84862,3,1.29505,0.439614
112447,4,3.66259,0.272418
19076,3,1.08612,0.38669
80337,3,1.13447,0.362927
33277,3,1.21785,0.479713
51317,3,0.0966896,0.402177
3821,5,2.58502,0.224858
99240,5,2.66839,0.228605
103096,3,0.175763,0.374201
1292,3,0.940022,0.441202
57939,4,0.404053,0.306919
112460,3,0.0617985,0.413087
69965,3,1.18069,0.389865
106440,4,0.17963,0.34013
100925,3,0.621789,0.390852
78459,3,1.37963,0.411565
79537,3,0.360808,0.291788
73996,3,2.51074,0.381102
88745,3,2.303,0.404591
84893,3,3.70334,0.373348
86796,3,1.43884,0.41715
```

```
70890,3,0.0301671,0.281937
38908,3,1.23344,0.42685
109422,3,2.77635,0.423741
34065,3,1.32352,0.459997
99701,4,0.228216,0.321522
98767,3,1.18033,0.412074
45343,4,0.243638,0.306539
97944,3,0.963268,0.367586
95447,3,1.6373,0.4585
85042,3,1.1722,0.360765
50564,3,2.83867,0.407455
120005,4,0.265716,0.317317
61174,4,3.92236,0.277727
22449,5,3.64551,0.228404
7978,3,1.37431,0.538242
35136,3,1.4313,0.494268
89042,3,1.56988,0.554827
32439,3,1.6028,0.528984
29800,3,2.77932,0.411077
72567,3,1.21437,0.396177
3909,3,1.59416,0.444327
67155,4,0.19839,0.309332
77257,4,2.70369,0.284972
107649,3,1.32488,0.399896
97295,3,2.67703,0.446405
42808,4,0.471364,0.32114
86486,3,2.88232,0.384824
69972,4,0.561811,0.345699
51523,3,2.81029,0.423837
117473,4,0.158392,0.317203
48113,3,2.36701,0.463272
25110,3,2.6924,0.459304
26394,3,1.37884,0.442316
85295,4,0.333029,0.349556
69671,3,1.05617,0.465941
40843,3,1.66296,0.442008
102040,3,1.22136,0.386754
12777,4,2.85202,0.264699
96441,4,3.74458,0.309812
88601,4,1.10436,0.249726
19859,3,1.31503,0.377962
64792,3,2.60936,0.45622
49081,3,1.40088,0.425502
86974,5,3.27509,0.247471
85235,4,0.558433,0.35619
```

1803,3,1.28404,0.405533
54211,4,0.165249,0.309959
104214,3,0.414895,0.218724
27072,4,3.1565,0.23878
113421,3,1.29753,0.397672
29295,4,0.220136,0.330077
108870,3,0.318163,0.226836
21770,4,4.04472,0.287624
97675,3,2.56116,0.453826
89348,3,2.94236,0.435016
83609,4,0.0611617,0.344778
62951,4,0.0857661,0.385671
12653,3,1.59705,0.447844
57757,4,3.89224,0.242101
78775,4,0.705816,0.36883
17378,5,3.01953,0.240611
102485,4,3.51163,0.26673
113357,3,1.39563,0.416187
113020,4,0.0945623,0.431057
86620,3,1.76145,0.575255
45333,3,2.5599,0.461759
59199,4,3.55065,0.260057
29650,3,1.86328,0.469269
109176,4,3.95635,0.24773
82860,4,2.68586,0.326039
58345,4,0.472457,0.330468
22449,6,4.55689,0.228764
114948,3,1.57368,0.440844
83601,3,1.41404,0.380333
32480,3,1.7906,0.473805
51502,3,2.5041,0.495294
70497,4,3.48142,0.260698
44897,3,1.23513,0.345372
111449,3,3.25505,0.491079
950,3,1.87292,0.461767
50384,3,1.61639,0.507075
78072,4,2.98533,0.249098
72603,3,3.41057,0.470679
7339,3,0.618872,0.367998
114924,3,1.63896,0.452318
110649,3,2.71102,0.481669
33817,4,0.748167,0.391039
113576,4,0.323851,0.38911
65859,4,0.242434,0.320953
71284,4,4.95674,0.28966
103389,3,1.89846,0.630629
29525,4,0.886611,0.379125
16537,5,0.801175,0.214509
30503,3,1.42985,0.403232
116745,4,0.477823,0.400583
80824,4,0.101056,0.373195
57507,4,0.921065,0.408731
46509,4,5.24136,0.301318
25878,4,0.212209,0.310437
76829,4,5.13506,0.300688
51459,4,1.94801,0.320205
116771,4,4.98308,0.268967
24186,4,0.123422,0.27082
17651,4,3.94971,0.272278
77760,4,4.9685,0.306114
49986,4,0.228927,0.390478
19335,3,1.88931,0.538318
38784,4,0.86229,0.379258
16852,4,4.89552,0.285128
40035,3,2.85354,0.491558
12444,3,1.55343,0.381707
7513,4,5.00738,0.265885
29568,4,0.861963,0.393554
96895,3,1.38581,0.391046
79537,4,0.555089,0.291809
88972,4,0.721891,0.307198
98470,3,1.83501,0.48807
50954,4,3.78739,0.256788
24813,4,2.10495,0.313277
23693,4,1.84302,0.293846
67275,3,2.12742,0.462379
18859,3,2.13916,0.544064
86162,4,0.0944425,0.278519
8102,6,1.57169,0.212781
14632,4,3.99433,0.260537
41926,4,0.873841,0.328951
36366,4,3.90427,0.273579
34017,3,1.34347,0.331168
76074,4,0.19324,0.366872
103096,4,0.263644,0.376584
5862,4,2.20991,0.344928
47592,4,2.22343,0.337956
36439,3,1.79818,0.435216
68184,4,0.675921,0.304176
10644,4,1.59075,0.295138
4151,4,5.22617,0.325801
33226,4,0.143459,0.466662
64797,4,0.782507,0.335106
36208,4,0.0623697,0.303067
80459,4,0.12416,0.426449
43797,3,1.87631,0.489779
44075,3,1.51144,0.367412
101997,4,1.02752,0.445637
105858,4,2.23792,0.263627
28103,4,3.9703,0.247855
82588,4,0.800715,0.380706
86400,4,0.773895,0.324454
117473,5,0.184791,0.314671
95149,4,0.7677,0.486398
65721,4,4.99919,0.375534
51317,4,0.131849,0.40238
89805,3,1.50206,0.37441
64394,4,2.21515,0.264225
65859,5,0.280713,0.316645
88574,4,0.219663,0.373408
83541,4,0.936402,0.405961
10798,4,0.920979,0.318362
64583,4,4.86373,0.341406
86974,6,4.1683,0.249003
89805,4,1.82393,0.543609
12444,4,2.2704,0.492625
80686,4,1.78612,0.316697
68682,4,1.50461,0.537335
40843,4,2.7716,0.49524
99240,6,2.85899,0.229722
950,4,3.12153,0.508177
62951,5,0.10006,0.415685
57443,4,1.36601,0.269766
72567,4,2.05509,0.550832
112460,4,0.0903209,0.393193
57548,4,0.0628347,0.429493
36439,4,2.90475,0.49954
3821,6,2.76967,0.227973
91772,4,0.0628359,0.273233
102485,5,4.99021,0.268419
32984,4,0.583786,0.284449
85523,4,0.143504,0.308903
70319,4,1.27994,0.395211
27072,5,3.97993,0.241373
67275,4,2.61836,0.543669

```
98036,4,4.87888,0.255607
116085,4,0.902668,0.392055
15457,4,1.39579,0.270355
78072,5,4.39944,0.251156
70890,4,0.051052,0.286495
17378,6,4.02604,0.241039
7339,4,0.77359,0.397471
24186,5,0.161991,0.277302
44897,4,1.33014,0.512915
88574,5,0.254347,0.398963
1599,4,2.05752,0.26071
7981,4,1.08799,0.253447
84893,4,5.65247,0.381801
90790,4,0.574076,0.31915
69965,4,1.36234,0.509898
10138,4,0.875016,0.394868
86796,4,2.15826,0.417203
50954,5,5.3469,0.262325
59199,5,4.93146,0.256784
26779,4,0.939689,0.356175
69671,4,1.32021,0.342548
5336,3,0.686059,0.337244
117712,4,0.808333,0.439279
70497,5,4.8353,0.264713
96441,5,4.33582,0.325391
38784,5,0.923882,0.55654
17651,5,5.4857,0.273931
78459,4,1.72454,0.497363
100925,4,0.829052,0.384492
100925,5,0.89814,0.609397
34017,4,1.63136,0.446116
61317,4,2.00632,0.259963
106440,5,0.221895,0.318787
28103,5,5.60513,0.249559
77052,4,1.17978,0.421262
43797,4,3.03096,0.477543
107649,4,2.08196,0.602246
25878,5,0.27284,0.327589
63721,5,0.0514057,0.257301
44075,4,2.20902,0.477282
109176,5,5.33247,0.248514
29295,5,0.293515,0.335366
72848,4,0.977973,0.361429
7513,5,6.21606,0.268215
76829,5,5.64856,0.301972
29650,4,2.79491,0.453412
29525,5,1.02301,0.505803
49986,5,0.25182,0.382774
36366,5,5.51191,0.274611
116771,5,6.18589,0.268267
50384,4,2.61109,0.420872
71284,5,5.45241,0.280444
46509,5,7.1632,0.302562
83609,5,0.0713553,0.355133
77760,5,6.73151,0.305888
95447,4,2.57291,0.387097
79537,5,0.638352,0.295552
47592,5,2.59401,0.337369
12444,5,2.74838,0.609279
91768,5,0.0943915,0.245564
73996,4,3.18026,0.432496
12843,5,3.54424,0.285974
114948,4,2.5421,0.372261
86736,4,3.54358,0.423395
12653,4,2.39557,0.519411
100017,4,1.40363,0.559406
38908,4,1.61296,0.512353
57757,5,5.13067,0.247919
910,4,3.59729,0.436131
39903,4,3.24629,0.402272
98767,4,1.63431,0.522015
77801,4,1.70456,0.307771
5862,5,2.57822,0.333706
64583,5,6.58957,0.341454
21770,5,4.71884,0.293945
27435,4,1.0774,0.406183
16852,5,5.38507,0.27853
64792,4,4.0445,0.495419
79248,4,1.41101,0.362355
88601,6,2.02466,0.252479
544,4,1.06256,0.406485
93858,4,1.2105,0.339859
112447,5,4.27302,0.280597
25278,4,3.27937,0.465654
82860,5,3.06956,0.329933
88745,4,3.5151,0.452276
55846,4,0.919282,0.308538
61174,5,4.57608,0.283691
24813,5,2.45577,0.307677
439,5,0.115208,0.277233
99701,5,0.304288,0.318233
54211,5,0.181774,0.315847
51459,5,2.3809,0.313586
34065,4,1.62895,0.455661
29271,4,1.41824,0.299062
3583,4,1.49113,0.488603
98959,4,1.48784,0.35641
98959,5,2.1491,0.650785
43587,4,1.10603,0.392513
78775,5,0.941088,0.372436
86486,4,3.49997,0.4179
40693,4,1.07038,0.346581
51502,4,2.97362,0.467367
32439,4,2.4042,0.489254
113283,4,0.591863,0.259345
114924,4,2.77363,0.402083
14632,5,5.05948,0.260569
89042,4,1.90628,0.578118
29800,4,3.24253,0.450605
16537,6,0.927676,0.234769
50564,4,4.46077,0.426854
85295,5,0.421836,0.392427
77358,4,1.01325,0.584889
25110,4,3.3655,0.487242
94761,5,0.16395,0.328847
113357,4,2.09345,0.545263
75181,4,1.47314,0.474772
4151,5,7.14243,0.325348
13402,4,0.847203,0.308044
23693,5,2.25258,0.300767
42808,5,0.667765,0.329122
69671,5,1.49624,0.465973
58345,5,0.641192,0.357764
53721,4,2.31165,0.464496
51523,4,3.61324,0.450031
77257,5,3.08993,0.288954
67155,5,0.257907,0.302224
76074,5,0.294461,0.389395
48113,4,2.81083,0.467803
12777,5,3.25945,0.265991
65721,5,6.61183,0.375534
3093,4,1.05936,0.338722
33226,5,0.166111,0.466684
79672,4,1.51382,0.442027
109422,4,3.23907,0.473358
```

```
97295,4,3.21243,0.431379
3909,4,2.27738,0.455874
23311,4,0.741647,0.294597
80337,4,1.45861,0.405043
96895,4,2.13202,0.34467
96895,5,2.66502,0.560809
22263,4,1.42996,0.407744
49081,4,1.80113,0.5154
1803,4,2.56808,0.405734
64924,4,1.3815,0.27424
89348,4,4.31547,0.453628
111449,4,4.79692,0.521805
34017,5,3.07079,0.647386
97675,4,3.01313,0.429163
83609,6,0.0815489,0.333635
76829,6,7.01791,0.310486
49986,6,0.286159,0.40573
1475,4,0.157826,0.248981
7339,5,1.00567,0.356882
54211,6,0.223086,0.315846
53721,5,3.30236,0.466313
71284,6,6.77421,0.284387
98470,4,2.88358,0.413064
16852,6,6.85373,0.282477
56452,4,0.828835,0.288037
40035,4,3.7096,0.646362
102040,4,1.62848,0.328835
102040,5,2.0356,0.426157
84893,5,7.79651,0.379355
95319,4,1.13633,0.296332
82588,5,0.972297,0.435692
57939,5,0.559459,0.310354
32480,4,2.62621,0.503631
72603,4,5.2056,0.490299
15330,4,1.23832,0.347676
98036,5,7.31832,0.255338
69972,5,0.734676,0.377644
81300,4,0.899701,0.292715
19335,4,2.15921,0.576716
19076,4,1.75449,0.297374
67275,5,3.10931,0.56933
85235,5,0.744577,0.345484
84862,4,2.12758,0.533069
15371,4,1.77141,0.396454
56997,4,1.34851,0.294341
```

```
110649,4,4.06654,0.509618
12444,6,3.70433,0.682271
35136,4,1.73801,0.462275
34017,6,3.93445,0.647374
42438,4,1.781,0.437621
93858,5,1.74009,0.551047
29800,5,4.47779,0.49096
96895,6,3.09143,0.625456
79537,6,0.832634,0.31948
98470,5,3.40787,0.654302
103389,4,2.53128,0.49634
97944,4,1.51371,0.280321
80686,5,2.18304,0.328579
114948,5,3.02631,0.610144
120005,5,0.354288,0.312694
33277,4,1.31153,0.637283
51317,5,0.184589,0.453196
102040,6,2.44272,0.431823
84720,4,1.12479,0.281422
45333,4,3.16223,0.450159
109422,5,4.47301,0.473358
8362,4,1.06106,0.304527
86620,4,2.1389,0.373429
86620,5,2.39054,0.579977
113020,5,0.122375,0.394719
57507,5,1.18423,0.347163
57507,6,1.84213,0.610484
88574,6,0.300592,0.402829
47080,4,1.31096,0.356299
114924,5,3.15185,0.593232
86486,5,4.94113,0.431115
103389,5,3.1641,0.630717
10644,5,1.94425,0.313431
97675,5,4.36904,0.518899
44897,5,1.99522,0.359242
44897,6,2.37526,0.631723
67155,6,0.297585,0.305136
116771,6,6.70138,0.268734
26394,4,2.22736,0.300053
58576,4,1.38618,0.419827
95149,5,1.5354,0.414767
72659,4,1.21817,0.302987
7513,6,6.73406,0.266473
73996,5,4.68671,0.486595
46509,6,7.68733,0.301319
```

```
27072,6,5.35232,0.246106
82588,6,2.00179,0.488152
18859,4,2.39083,0.438841
65859,6,0.370031,0.378748
84478,4,0.523596,0.279322
113576,5,0.384573,0.387708
93858,6,1.96706,0.505589
70890,5,0.0707766,0.342009
439,6,0.15361,0.27067
39903,5,4.583,0.393437
95319,5,1.32572,0.480437
45343,5,0.32485,0.285906
59199,6,5.52323,0.260542
62523,4,1.25543,0.288171
544,5,1.31257,0.453995
98767,5,2.08828,0.683394
62523,5,2.02263,0.429223
102485,6,5.54468,0.268643
86796,5,2.51797,0.534439
57757,6,5.66143,0.24791
27435,5,1.29288,0.411214
114924,6,4.16044,0.593267
107350,4,1.82192,0.280984
107350,5,1.90868,0.439963
50384,5,2.73543,0.596879
70497,6,5.22213,0.272352
26394,5,2.65162,0.506124
80459,5,0.182102,0.470009
88745,5,4.84842,0.484957
89805,5,2.25309,0.37211
49908,4,0.411853,0.241534
7751,4,0.78546,0.27808
116085,5,1.6925,0.438464
116085,6,1.97459,0.438464
95447,5,3.50851,0.51612
97944,5,2.20175,0.529359
26779,5,1.14851,0.363988
25110,5,4.71169,0.473977
109176,6,6.02053,0.253577
29568,5,0.928268,0.507565
1803,5,2.91049,0.40569
1803,6,3.76652,0.400043
18859,5,3.27166,0.513667
43797,5,4.04128,0.668414
91768,6,0.111049,0.252339
```

103096,5,0.307585,0.37415	62207,5,2.37615,0.539645	106440,6,0.243028,0.305253
116745,5,0.637098,0.305804	62207,6,3.05505,0.586677	83541,5,1.27083,0.345677
86736,5,4.77613,0.454274	65721,6,7.09562,0.375536	83541,6,2.07346,0.47374
98767,6,2.99623,0.60746	44075,5,3.13914,0.557169	78459,6,3.44908,0.640425
1292,4,1.31603,0.378541	44075,6,4.53431,0.635478	84862,5,3.05261,0.527028
1292,5,1.3787,0.590748	36366,6,6.2009,0.27897	43726,4,1.65788,0.282236
34065,5,2.138,0.582961	78459,5,2.06945,0.54814	43726,5,1.90656,0.334823
99825,4,0.896758,0.337865	21770,6,6.06708,0.295668	43726,6,2.48682,0.51994
35136,5,2.55589,0.586062	62951,6,0.142944,0.43321	29271,5,1.71682,0.274861
17651,6,5.92456,0.273957	67275,6,4.41849,0.593932	78072,6,4.71369,0.258066
64792,5,4.95778,0.575381	98036,6,10.0017,0.257388	112447,6,5.69736,0.282511
33817,5,1.01537,0.367934	12114,4,0.713934,0.249947	30503,4,1.69795,0.296335
77760,6,7.21233,0.306423	32439,5,2.74766,0.55672	30503,5,1.96605,0.493799
107649,5,3.0283,0.619969	40035,5,4.70834,0.56589	43797,6,4.47428,0.654595
4151,6,7.66504,0.333179	64583,6,8.31541,0.348779	47592,6,3.70572,0.350676
51523,5,4.81765,0.457602	7978,5,1.90289,0.554591	36439,5,3.31972,0.562502
49081,5,2.90182,0.547146	88694,4,1.4702,0.290792	103096,6,0.336878,0.317441
25278,5,4.29711,0.434288	88601,5,1.22707,0.246474	32480,5,3.58119,0.505924
910,5,5.23241,0.442599	19335,5,2.69902,0.56537	80824,5,0.122331,0.335209
51502,5,4.22567,0.467402	14632,6,6.25778,0.260935	5862,6,3.5604,0.354247
19849,4,0.9324,0.239952	110649,5,5.12082,0.537055	34065,6,3.25791,0.620201
72603,5,6.64163,0.528653	96100,4,0.676994,0.23067	114622,4,0.88206,0.242216
48113,5,3.8464,0.463345	64394,5,2.51721,0.261591	113283,5,0.680643,0.265046
7978,4,1.58574,0.493939	84893,6,9.94055,0.381847	13402,5,1.03547,0.299685
117473,6,0.263986,0.422989	68682,5,1.7912,0.31027	47080,5,2.11211,0.395032
111449,5,5.99615,0.529437	68682,6,2.36438,0.558428	98470,6,3.93216,0.654312
3765,4,0.720075,0.273992	114948,6,4.11579,0.610218	110649,6,6.17511,0.580806
45333,5,4.06573,0.492145	89348,5,6.27704,0.453638	76074,6,0.322067,0.380366
77358,5,1.23038,0.366334	36208,5,0.0815604,0.329289	5336,4,1.42112,0.337374
96441,6,5.7154,0.318136	99461,4,0.807578,0.250429	5336,5,1.66614,0.504768
38784,6,1.78617,0.556594	38908,5,2.08736,0.57033	77358,6,1.37513,0.487943
101997,5,1.19878,0.447543	12843,6,4.57798,0.291205	117712,5,1.12268,0.423621
105090,4,0.339116,0.24753	12777,6,4.21012,0.270979	117712,6,1.30231,0.47583
97295,5,4.46171,0.453819	28103,6,6.07223,0.246787	33226,6,0.226515,0.466657
29650,5,3.26073,0.458846	24813,6,3.3913,0.330728	86620,6,2.64217,0.5238
85523,5,0.172205,0.312147	55846,5,1.76785,0.428554	40843,5,3.18733,0.508894
51317,6,0.202169,0.520363	116745,6,0.764517,0.449676	3093,5,1.22662,0.348122
75181,5,1.71866,0.424509	73184,4,0.718223,0.234039	35136,6,2.8626,0.636865
77052,5,1.54846,0.484032	40693,5,1.30824,0.342728	950,5,3.74584,0.567413
107350,6,2.68951,0.485064	15510,4,1.31115,0.23598	89042,5,2.69123,0.507832
50564,5,6.08287,0.4311	61174,6,6.10144,0.284165	51459,6,2.81379,0.329834
105858,5,2.54309,0.256885	72848,5,1.1953,0.392587	24186,6,0.262271,0.27229
50954,6,5.79248,0.264517	40035,6,5.27905,0.646408	19859,4,1.92871,0.287283
26394,6,3.39407,0.554268	53721,6,4.07291,0.477283	19859,5,2.45472,0.371984
62207,4,1.78211,0.283678	3909,5,3.52994,0.575425	12653,5,3.19409,0.545407

```
83601,4,2.42406,0.309088
83601,5,3.9391,0.669328
85042,4,1.67457,0.288214
85042,5,1.7583,0.367582
85042,6,3.01423,0.402456
1599,5,2.33809,0.259767
42438,5,2.02386,0.50429
95447,6,4.91191,0.524535
19076,5,2.17223,0.375521
19076,6,2.58997,0.550375
72567,5,2.70898,0.499374
72567,6,3.17605,0.624358
45333,6,4.66806,0.518783
79672,5,1.76612,0.437345
29525,6,1.43222,0.526889
22263,5,1.74773,0.425143
18859,6,3.77499,0.513696
99701,6,0.349931,0.312515
10798,5,1.11487,0.348992
23693,6,3.0717,0.31296
77052,6,1.76967,0.577275
95149,6,2.44268,0.486443
56452,5,1.00333,0.298368
3583,5,1.88353,0.507587
29800,6,4.941,0.464215
41926,5,1.05781,0.343518
107649,6,3.5961,0.619976
12653,6,4.33484,0.545414
63721,6,0.0616868,0.302491
82860,6,4.09275,0.341283
78775,6,1.0352,0.432845
111449,6,7.02406,0.573428
32439,6,3.66354,0.556736
86796,6,3.47719,0.532622
79672,6,2.52303,0.499189
104217,4,0.404908,0.221729
80459,6,0.206934,0.426484
89042,6,3.02763,0.520561
61317,5,2.27991,0.257251
36439,6,4.42629,0.566662
15330,5,1.5135,0.374602
41926,6,1.47173,0.350622
72603,6,8.25716,0.567589
86162,5,0.121426,0.273169
7978,6,2.74862,0.549335
50564,6,8.31326,0.431107
80337,5,1.70171,0.426505
94761,6,0.204937,0.357678
109422,6,4.93573,0.473362
77257,6,3.99116,0.295146
86486,6,6.17641,0.444048
25878,6,0.333472,0.305445
73996,6,5.85838,0.486596
114046,4,0.169938,0.223151
39903,6,5.53779,0.423944
10138,5,1.02085,0.273101
70319,5,1.49326,0.320957
70319,6,2.13323,0.510627
113357,5,2.69157,0.602051
95319,6,1.45198,0.475725
23311,5,0.85875,0.337464
72659,5,1.4009,0.290535
25110,6,5.21652,0.502755
97295,6,4.99712,0.490598
38908,6,2.56177,0.580271
80824,6,0.132968,0.36989
51523,6,5.82132,0.450032
98959,6,2.23176,0.745535
32480,6,4.77492,0.581666
950,6,4.83837,0.567414
88972,5,0.806819,0.321793
48113,6,4.43815,0.473748
86736,6,5.23833,0.445195
84862,6,3.88514,0.524353
64797,5,0.947245,0.329411
64797,6,1.35909,0.345076
51502,6,4.53869,0.505019
64924,5,1.67234,0.254132
75181,6,2.53707,0.4748
64792,6,5.34918,0.575381
29295,6,0.352218,0.324267
89348,6,7.65015,0.493596
42808,6,0.864167,0.322693
22263,6,2.54216,0.463226
81300,5,1.04965,0.326733
19335,6,3.37377,0.576707
105858,6,3.35688,0.268163
88694,5,1.72965,0.266474
88694,6,2.24854,0.525907
49081,6,4.00251,0.602003
29650,6,4.34764,0.526884
85235,6,0.977258,0.356164
69671,6,1.76029,0.325706
33277,5,1.87361,0.282531
33277,6,2.24833,0.609656
80686,6,2.97687,0.328612
910,6,7.35808,0.475042
85523,6,0.215257,0.316388
58345,6,0.77618,0.338935
3583,6,2.74682,0.511409
10138,6,1.16669,0.409872
97675,6,5.8756,0.523407
3909,6,4.78249,0.541516
25278,6,5.42793,0.504247
68184,5,0.755441,0.314767
99461,5,1.13795,0.239414
86400,5,0.977552,0.331179
86400,6,1.50706,0.364929
113357,6,3.09033,0.624068
64394,6,3.32272,0.267467
101997,6,1.31295,0.442534
544,6,1.8751,0.426843
8362,5,1.28444,0.278224
12114,5,0.821024,0.257898
42438,6,2.67149,0.518445
70890,6,0.0905013,0.287871
40843,6,4.01881,0.542609
58576,5,1.77123,0.577126
27435,6,1.3647,0.469164
88745,6,6.90899,0.484956
77801,5,1.94807,0.267755
77801,6,2.92211,0.610644
57939,6,0.683783,0.314343
69965,5,1.63481,0.307385
69965,6,2.17974,0.70337
32984,5,0.648651,0.265788
7751,5,0.950819,0.268721
69972,6,0.821109,0.368378
49908,5,0.473631,0.23691
79248,5,2.08291,0.276944
79248,6,2.35168,0.552757
7981,5,1.53307,0.266068
3765,5,0.871669,0.251614
58576,6,2.61835,0.479283
15371,5,2.5766,0.438399
```

```
113421,4,1.85362,0.27038
113421,5,2.68775,0.318111
84720,5,1.27817,0.262151
91772,5,0.080789,0.331659
104214,4,0.484044,0.232483
15371,6,3.30127,0.415649
10644,6,2.828,0.321871
7981,6,2.22543,0.274665
113020,6,0.161312,0.400632
40693,6,1.78397,0.383858
72848,6,1.73862,0.390552
56997,5,1.53239,0.280372
99825,5,1.08555,0.267259
120005,6,0.425146,0.314119
84478,5,0.628316,0.247248
36208,6,0.100751,0.397195
88972,6,0.934211,0.330106
43587,5,1.35182,0.358851
43587,6,1.96628,0.435262
10798,6,1.4057,0.352988
73184,5,0.811904,0.229945
15330,6,2.13266,0.448282
3093,6,1.72842,0.368402
29568,6,1.12718,0.260404
1599,6,2.99276,0.268114
81300,6,1.54949,0.293324
26779,6,1.30512,0.375865
8362,6,1.84289,0.365685
68184,6,0.874722,0.335308
45343,6,0.373578,0.346033
86162,6,0.141664,0.31106
```

```
72659,6,1.94908,0.364736
54035,4,0.116809,0.220275
114622,5,0.997111,0.236998
15510,5,1.49846,0.243336
85295,6,0.444038,0.239176
13402,6,1.45907,0.356272
61317,6,2.91828,0.260953
57548,5,0.0659764,0.458816
57443,5,1.65359,0.298359
32984,6,0.745949,0.374893
57443,6,2.58822,0.290483
23311,6,1.24909,0.324483
100017,5,1.75454,0.260951
100017,6,2.19317,0.486756
80337,6,2.18791,0.51219
105090,5,0.406939,0.22638
15457,5,1.68964,0.292513
15457,6,2.64466,0.30024
56997,6,2.63572,0.375795
62523,6,2.23187,0.261458
47080,6,2.3306,0.436509
29271,6,2.38862,0.323216
84720,6,1.99395,0.375739
19849,5,1.07226,0.221861
90790,5,0.615082,0.233273
90790,6,0.697092,0.40204
1475,5,0.191645,0.247594
99825,6,1.46313,0.316808
108870,4,0.445428,0.214609
104214,5,0.645392,0.29391
103389,6,3.92349,0.32802
```

```
113576,6,0.445296,0.232633
64924,6,2.32674,0.291262
91772,6,0.10323,0.312294
99461,6,1.21137,0.248953
104217,5,0.478528,0.233807
113283,6,0.739829,0.259757
112460,5,0.0950746,0.405367
112460,6,0.152119,0.390091
33817,6,1.12225,0.23776
56452,6,1.26506,0.474161
104214,6,0.783691,0.214492
104217,6,0.607363,0.221299
7751,6,1.28154,0.292951
3765,6,1.21276,0.272645
12114,6,1.21369,0.298775
84478,6,0.863934,0.306483
114622,6,1.22721,0.262038
105090,6,0.542585,0.229679
96100,5,1.06385,0.220416
49908,6,0.535409,0.225733
1475,6,0.214192,0.235886
73184,6,0.936813,0.637053
114046,5,0.260572,0.223553
15510,6,1.7482,0.211906
19849,6,1.21212,0.21533
96100,6,1.40235,0.211269
114046,6,0.385193,0.211304
108870,5,0.50906,0.212473
54035,5,0.208588,0.219265
```

Bibliography

[1] Accion Systems Inc. TILE, 2017.

[2] Aerojet. Monopropellant Data Sheets, July 2009.

[3] Sonja Alexander. NASA Awards Launch Services Contract for TESS, June 2016. Library Catalog: www.nasa.gov.

[4] Alluxa. ULTRA Series Notch Filters, June 2016. Section: Products.

[5] Inc Apollo Fusion. Apollo Constellation Engine (ACE).

[6] Gomspace A/S. GOMspace | NanoPower P60 System.

[7] Alessandra Babuscia, Craig Hardgrove, Kar-Ming Cheung, Paul Scowen, and Jim Crowell. Telecommunication system design for interplanetary CubeSat missions: LunaH-Map. In *2017 IEEE Aerospace Conference*, pages 1–9, March 2017.

[8] Pramit Biswas. hamiltonian(Graph, Source, Destination), June 2015.

[9] Matthew R. Bolcar. The Large UV/Optical/Infrared Surveyor (LUVOIR): Decadal Mission concept technology development overview. In *UV/Optical/IR Space Telescopes and Instruments: Innovative Technologies and Concepts VIII*, volume 10398, page 103980A. International Society for Optics and Photonics, September 2017.

[10] Matthew R. Bolcar, Steve Aloezos, Vincent T. Bly, Christine Collins, Julie Crooke, Courtney D. Dressing, Lou Fantano, Lee D. Feinberg, Kevin France, Gene Gochar, Qian Gong, Jason E. Hylan, Andrew Jones, Irving Linares, Marc Postman, Laurent Pueyo, Aki Roberge, Lia Sacks, Steven Tompkins, and Garrett West. The Large UV/Optical/Infrared Surveyor (LUVOIR): Decadal Mission concept design update. In *UV/Optical/IR Space Telescopes and Instruments: Innovative Technologies and Concepts VIII*, volume 10398, page 1039809. International Society for Optics and Photonics, September 2017.

[11] X. Bonfils, N. Astudillo-Defru, R. Díaz, J.-M. Almenara, T. Forveille, F. Bouchy, X. Delfosse, C. Lovis, M. Mayor, F. Murgas, F. Pepe, N. C. Santos, D. Ségransan, S. Udry, and A. Wünsche. A temperate exo-Earth around a quiet M dwarf at 3.4 parsec,. *Astronomy & Astrophysics*, 613:A25, May 2018. Publisher: EDP Sciences.

[12] Natasha Bosanac, Andrew D. Cox, Kathleen C. Howell, and David C. Folta. Trajectory design for a cislunar CubeSat leveraging dynamical systems techniques: The Lunar IceCube mission. *Acta Astronautica*, 144:283–296, March 2018.

[13] Busek Co. Inc. BIT-3 RF Ion Thruster, 2017.

[14] Kerri Cahoy, Peter Grenfell, Angela Crews, Michael Long, Paul Serra, Anh Nguyen, Riley Fitzgerald, Christian Haughwout, Rodrigo Diez, Alexa Aguilar, John Conklin, Cadence Payne, Joseph Kusters, Chloe Sackier, Mia LaRocca, and Laura Yenchesky. The CubeSat Laser Infrared CrosslinK Mission (CLICK). In *International Conference on Space Optics — ICSO 2018*, volume 11180, page 111800Y. International Society for Optics and Photonics, July 2019.

[15] Richard Capek. Which is the best projection for the world map? In *Proceedings of the 20th international Cartographic Conference*, volume 5, pages 3084–3093, 2001.

[16] James R. Clark and Kerri Cahoy. Optical Detection of Lasers with Near-term Technology at Interstellar Distances. *The Astrophysical Journal*, 867(2):97, 2018.

[17] Darren Rowen, Rick Dolphus, Patrick Doyle, and Addison Faler. OCSD-A / AeroCube 7-A Status Update. In *Cal Poly CubeSat Developer's Workshop*, 2016.

[18] David Le Mignant. Keck LGS AO Performance, 2007.

[19] E. S. Douglas, J. R. Males, J. Clark, O. Guyon, J. Lumbres, W. Marlow, and K. L. Cahoy. Laser Guide Star for Large Segmented-aperture Space Telescopes. I. Implications for Terrestrial Exoplanet Detection and Observatory Stability. *The Astronomical Journal*, 157(1):36, January 2019.

[20] Ewan S. Douglas, Gregory Allan, Derek Barnes, Joseph S. Figura, Christian A. Haughwout, Jennifer N. Gubner, Alex A. Knoedler, Sarah LeClair, Thomas J. Murphy, and Nikolaos Skouloudis. Design of the deformable mirror demonstration CubeSat (DeMi). In *Techniques and Instrumentation for Detection of Exoplanets VIII*, volume 10400, page 1040013. International Society for Optics and Photonics, 2017.

[21] Bernard Edwards and Andrew Fletcher. NASA's LCRD, Laser Communications Relay Demonstration, October 2013. Library Catalog: NASA NTRS.

[22] Enpulsion GmbH. IFM Nano Thruster, 2018.

[23] Lee Feinberg, Matthew Bolcar, Scott Knight, and David Redding. Ultra-stable segmented telescope sensing and control architecture. In *UV/Optical/IR Space Telescopes and Instruments: Innovative Technologies and Concepts VIII*, volume 10398, page 103980E. International Society for Optics and Photonics, September 2017.

[24] David C. Folta, Natasha Bosanac, Andrew Cox, and Kathleen C. Howell. The Lunar IceCube Mission Design: Construction of Feasible Transfer Trajectories with a Constrained Departure. In *26th AAS/AIAA Spaceflight Mechanics Meeting*, Napa, CA, United States, February 2016.

[25] Gabriel Soto, Amlan Sinha, Dmitry Savransky, Daniel Garrett, and Christian Delacroix. Starshade Orbital Maneuver Study for WFIRST, January 2017.

[26] Alain H. Greenaway and Stuart E. Clark. PHAROS: an agile satellite-borne laser guidestar. In *Laser Beam Propagation and Control*, volume 2120, pages 206–211. International Society for Optics and Photonics, June 1994.

[27] Craig Hardgrove, Joe DuBois, Lena Heffern, Ernest Cisneros, James Bell, Teri Crain, Richard Star, Thomas Prettyman, Igor Lazbin, Bob Roebuck, Nathaniel Struebel, Ethan Clark, Derek Nelson, Jeremy Bauman, Bobby Williams, Michael Tsay, Joshua Model, Pete Hruby, Alessandra Babuscia, Steve Stem, Devon Sanders, Elliot Hegel, Mitchell Wiens, Sean Parlapiano, Patrick Hailey, Tyler O'Brien, Katherine Mesick, and Dan Coupland. The Lunar Polar Hydrogen Mapper (LunaH-Map) Mission. *Small Satellite Conference*, August 2019.

[28] J. D. Myers. LUVOIR Design, July 2019.

[29] Kevin Burns. Ancient Aliens, 2010.

[30] Joseph Kirk. Traveling Salesman Problem (TSP) Genetic Algorithm Toolbox, June 2020.

[31] Andrew Klesh, Brian Clement, Cody Colley, John Essmiller, Daniel Forgette, Joel Krajewski, Anne Marinan, and Tomas Martin-Mur. MarCO: Early Operations of the First CubeSats to Mars. *Small Satellite Conference*, August 2018.

[32] Marcus T. Knopp, Dirk Giggenbach, Ramon Mata Calvo, Christian Fuchs, Karen Saucke, Frank Heine, Florian Sellmaier, and Felix Huber. Connectivity services based on optical ground-to-space links. *Acta Astronautica*, 148:369–375, July 2018.

[33] David Krejci, Alexander Reissner, Bernhard Seifert, David Jelem, Thomas Hörbe, Florin Plesescu, Pete Friedhoff, and Steve Lai. Demonstration of the ifm nano feep thruster in low earth orbit. In *The 4S Symposium 2018, Sorrento, Italy*, 2018.

[34] Paul A. Lightsey, Allison A. Barto, and James Contreras. Optical performance for the James Webb Space Telescope. In *Optical, Infrared, and Millimeter Space Telescopes*, volume 5487, pages 825–832. International Society for Optics and Photonics, October 2004.

[35] LISA Pathfinder Collaboration, M. Armano, H. Audley, J. Baird, P. Binetruy, M. Born, D. Bortoluzzi, E. Castelli, A. Cavalleri, A. Cesarini, A. M. Cruise,

K. Danzmann, M. de Deus Silva, I. Diepholz, G. Dixon, R. Dolesi, L. Ferraioli, V. Ferroni, E. D. Fitzsimons, M. Freschi, L. Gesa, F. Gibert, D. Giardini, R. Giusteri, C. Grimani, J. Grzymisch, I. Harrison, G. Heinzel, M. Hewitson, D. Hollington, D. Hoyland, M. Hueller, H. Inchauspé, O. Jennrich, P. Jetzer, N. Karnesis, B. Kaune, N. Korsakova, C. J. Killow, J. A. Lobo, I. Lloro, L. Liu, J. P. López-Zaragoza, R. Maarschalkerweerd, D. Mance, N. Meshksar, V. Martín, L. Martin-Polo, J. Martino, F. Martin-Porqueras, I. Mateos, P. W. McNamara, J. Mendes, L. Mendes, M. Nofrarias, S. Paczkowski, M. Perreur-Lloyd, A. Petiteau, P. Pivato, E. Plagnol, J. Ramos-Castro, J. Reiche, D. I. Robertson, F. Rivas, G. Russano, J. Slutsky, C. F. Sopuerta, T. Sumner, D. Texier, J. I. Thorpe, D. Vetrugno, S. Vitale, G. Wanner, H. Ward, P. J. Wass, W. J. Weber, L. Wissel, A. Wittchen, and P. Zweifel. LISA Pathfinder platform stability and drag-free performance. *Physical Review D*, 99(8):082001, April 2019. Publisher: American Physical Society.

[36] Philip Lubin. A Roadmap to Interstellar Flight. *arXiv:1604.01356 [astro-ph, physics:physics]*, April 2016. arXiv: 1604.01356.

[37] Marcos van Dam. W.M. Keck Observatory Natural Guide Star Adaptive Optics, 2007.

[38] Weston Marlow, Ashley Carlton, Yoon Hyosang, James Clark, Christian Haughwout, Kerri Cahoy, Jared Males, Laird Close, and Katie Morzinski. Laser-Guide-Star Satellite for Ground-Based Adaptive Optics Imaging of Geosynchronous Satellites. *Journal of Spacecraft and Rockets*, pages 1–18, May 2017.

[39] James Mason, Matthew Baumgart, Thomas Woods, Daniel Hegel, Bryan Rogler, George Stafford, Stanley Solomon, and Phillip Chamberlin. MinXSS CubeSat On-Orbit Performance and the First Flight of the Blue Canyon Technologies XACT 3- axis ADCS. *Small Satellite Conference*, August 2016.

[40] Colin R. McInnes. Low-Thrust Orbit Raising with Coupled Plane Change and J Precession. *Journal of Guidance, Control, and Dynamics*, 20(3):607–609, 1997. Publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/2.4084.

[41] National Geographic Maps. World Classic Map [Enlarged], 2017.

[42] Mamadou N'Diaye, Rémi Soummer, Laurent Pueyo, Alexis Carlotti, Christopher C. Stark, and Marshall D. Perrin. APODIZED PUPIL LYOT CORONAGRAPHS FOR ARBITRARY APERTURES. V. HYBRID SHAPED PUPIL DESIGNS FOR IMAGING EARTH-LIKE PLANETS WITH FUTURE SPACE OBSERVATORIES. *The Astrophysical Journal*, 818(2):163, February 2016. Publisher: American Astronomical Society.

[43] Dennis Overbye. NASA Again Delays Launch of Troubled Webb Telescope; Cost Estimate Rises to $9.7 Billion. *The New York Times*, June 2018.

[44] Sang Park, Michael J. Eisenhower, Matthew R. Bolcar, Marcel Bluth, Julie Crooke, Lee D. Feinberg, Jason E. Hylan, William Hayden, J. Scott Knight, Bryan Matonak, and Kan Yang. LUVOIR Thermal Architecture Overview and Enabling Technologies for Picometer-Scale WFE Stability. In *2019 IEEE Aerospace Conference*, pages 1–13, March 2019. ISSN: 1095-323X.

[45] Eliad Peretz, John Mather, Richard Slonaker, John O'Meara, Sara Seager, Randy Campbell, Tiffany Hoerbelt, and Isabel Kain. Orbiting Configurable Artificial Star (ORCAS) for Visible Adaptive Optics from the Ground. *Bulletin of the American Astronomical Society*, 51(7):284, September 2019. Conference Name: Bulletin of the American Astronomical Society.

[46] Phase Four, Inc. Phase Four Radio Frequency Thruster, May 2017.

[47] L. Pueyo, N. Zimmerman, M. Bolcar, T. Groff, C. Stark, G. Ruane, J. Jewell, R. Soummer, K. St Laurent, J. Wang, D. Redding, J. Mazoyer, K. Fogarty, Roser Juanola-Parramon, S. Domagal-Goldman, A. Roberge, O. Guyon, and A. Mandell. The LUVOIR architecture "A" coronagraph instrument. In *UV/Optical/IR Space Telescopes and Instruments: Innovative Technologies and Concepts VIII*, volume 10398, page 103980F. International Society for Optics and Photonics, September 2017.

[48] Jordi Puig-Suari, Guy Zohar, and Kyle Leveque. Deployment of CubeSat Constellations Utilizing Current Launch Opportunities. *Small Satellite Conference*, August 2013.

[49] Aki Roberge, Maxime J. Rizzo, Andrew P. Lincowski, Giada N. Arney, Christopher C. Stark, Tyler D. Robinson, Gregory F. Snyder, Laurent Pueyo, Neil T. Zimmerman, Tiffany Jansen, Erika R. Nesvold, Victoria S. Meadows, and Margaret C. Turnbull. Finding the Needles in the Haystacks: High-fidelity Models of the Modern and Archean Solar System for Simulating Exoplanet Observations. *Publications of the Astronomical Society of the Pacific*, 129(982):124401, 2017.

[50] Garreth Ruane, Dimitri Mawet, Jeffrey Jewell, and Stuart Shaklan. Performance and sensitivity of vortex coronagraphs on segmented space telescopes. In *Techniques and Instrumentation for Detection of Exoplanets VIII*, volume 10400, page 104000J. International Society for Optics and Photonics, 2017.

[51] Thomas S. Schwarze, Germán Fernández Barranco, Daniel Penkert, Marina Kaufer, Oliver Gerberding, and Gerhard Heinzel. Picometer-Stable Hexagonal Optical Bench to Verify LISA Phase Extraction Linearity and Precision. *Physical Review Letters*, 122(8):081104, February 2019. Publisher: American Physical Society.

[52] Sara Seager, Margaret Turnbull, William Sparks, Mark Thomson, Stuart B. Shaklan, Aki Roberge, Marc Kuchner, N. Jeremy Kasdin, Shawn Domagal-Goldman, Webster Cash, Keith Warfield, Doug Lisman, Dan Scharf, David

Webb, Rachel Trabert, Stefan Martin, Eric Cady, and Cate Heneghan. The Exo-S probe class starshade mission. In *Techniques and Instrumentation for Detection of Exoplanets VII*, volume 9605, page 96050W. International Society for Optics and Photonics, September 2015. tex.ids: seager_exo-s_2015-1.

[53] Seth Shostak. Signals from A Nearby Star System?, April 2018.

[54] Garrett Shea. NASA Systems Engineering Handbook Revision 2, June 2017.

[55] Doug Sinclair and Jonathan Dyer. Radiation Effects and COTS Parts in SmallSats. *Small Satellite Conference*, August 2013. tex.ids: sinclair_radiation_nodate.

[56] H. Philip Stahl. Survey of cost models for space telescopes. *Optical Engineering*, 49(5):053005, May 2010. Publisher: International Society for Optics and Photonics.

[57] Christopher C. Stark, Aki Roberge, Avi Mandell, Mark Clampin, Shawn D. Domagal-Goldman, Michael W. McElwain, and Karl R. Stapelfeldt. LOWER LIMITS ON APERTURE SIZE FOR AN EXOEARTH DETECTING CORONAGRAPHIC MISSION. *The Astrophysical Journal*, 808(2):149, July 2015.

[58] John Steeves, Hyeong Jae Lee, Evan Hilgemann, Dylan McKeithen, Christine Bradley, David Webb, Stuart Shaklan, Stefan Martin, and Douglas Lisman. Development of low-scatter optical edges for starshades. In *Advances in Optical and Mechanical Technologies for Telescopes and Instrumentation III*, volume 10706, page 107065K. International Society for Optics and Photonics, July 2018.

[59] David Sternberg, John Essmiller, Cody Colley, Andrew Klesh, and Joel Krajewski. Attitude Control System for the Mars Cube One Spacecraft. In *2019 IEEE Aerospace Conference*, pages 1–10, March 2019. tex.ids: sternberg_attitude_2019-1 ISSN: 1095-323X.

[60] Stuart Shaklan and Nick Siegler. Starshade Technology: Current Status at JPL, November 2015.

[61] Richard Swinbank and R. James Purser. Fibonacci grids: A novel approach to global modelling. *Quarterly Journal of the Royal Meteorological Society*, 132(619):1769–1793, 2006. _eprint: https://rmets.onlinelibrary.wiley.com/doi/pdf/10.1256/qj.05.227.

[62] John T. Trauger and Wesley A. Traub. A laboratory demonstration of the capability to image an Earth-like extrasolar planet. *Nature*, 446(7137):771–773, April 2007. Number: 7137 Publisher: Nature Publishing Group.

[63] Robert K. Tyson and Benjamin W. Frazier. Focal Anisoplanatism (the "Cone Effect"). *Field Guide to Adaptive Optics, Second Edition*, pages 41–42, April 2012. Publisher: International Society for Optics and Photonics.

[64] United States Air Force. Airborne Laser (YAL-1), March 2007.

[65] Ezinne Uzo-Okoro, Christian Haughwout, Emily Kiley, Mary Dahl, and Kerri Cahoy. Ground-Based 1U CubeSat Robotic Assembly Demonstration. *Small Satellite Conference*, August 2020.

[66] VACCO Industries. JPL MarCO Micro CubeSat Propulsion System, November 2015.

[67] VACCO Industries. Green Propulsion System, August 2019. Library Catalog: www.cubesat-propulsion.com.

[68] Xu Wang, Fang Shi, and J. Kent Wallace. Zernike wavefront sensor (ZWFS) development for Low Order Wavefront Sensing (LOWFS). In *Space Telescopes and Instrumentation 2016: Optical, Infrared, and Millimeter Wave*, volume 9904, page 990463. International Society for Optics and Photonics, July 2016.

[69] John Noble Wilford. The Impossible Quest for the Perfect Map. *The New York Times*, October 1988.

[70] Laura Yenchesky, Ondrej Cierny, Peter Grenfell, William Kammerer, Paula Periera, Tao Sevigny, and Kerri Cahoy. Optomechanical Design and Analysis for Nanosatellite Laser Communications. *Small Satellite Conference*, August 2019.

[71] H. Zech, F. Heine, D. Tröndle, S. Seel, M. Motzigemba, R. Meyer, and S. Philipp-May. LCT for EDRS: LEO to GEO optical communications at 1,8 Gbps between Alphasat and Sentinel 1a. In *Unmanned/Unattended Sensors and Sensor Networks XI; and Advanced Free-Space Optical Communication Techniques and Applications*, volume 9647, page 96470J. International Society for Optics and Photonics, October 2015.

[72] Ondrej Čierny and Kerri L. Cahoy. On-orbit beam pointing calibration for nanosatellite laser communications. *Optical Engineering*, 58(4):041605, November 2018.