Cislunar Space Situational Awareness Architecture Design and Analysis

by

Michael Klonowski

B.A., St. John's University (MN), 2021

M.S., University of Colorado Boulder, 2023

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Ann & H.J. Smead Department of Aerospace Engineering Sciences

2025

Committee Members:

Marcus J. Holzinger, Chair

Prof. Hanspeter Schaub, Co-Chair

Dr. Naomi Owens-Fahrner

Prof. Jay McMahon

Prof. Zachary Sunberg

Prof. Ryan Russell

Klonowski, Michael (Ph.D., Aerospace Engineering Sciences)

Cislunar Space Situational Awareness Architecture Design and Analysis

Thesis directed by Prof. Marcus J. Holzinger

As interest in missions within Cislunar space increases, so does the need for robust space situational awareness (SSA) capabilities. The chaotic multi-body dynamics of the regime, combined with the massive volume of space when compared to the near-Earth regime present unique challenges for Cislunar SSA. Earth-based sensors, already overwhelmed due to the increasing numbers of objects in LEO, are not well-suited to meet the SSA demands of Cislunar space. In this thesis, we present multiple methods for designing and analyzing architectures consisting of space-based observers for Cislunar SSA.

The first two contributions in this thesis focus on the multi-objective design of distributed architectures. First, multi-objective Monte Carlo Tree Search (MO-MCTS) is leveraged to identify Pareto-optimal architectures, maximizing for coverage of a trajectory and volumes of Cislunar space while minimizing a total cost representing the number of observers as well as their capabilities. The second contribution extends this work by including a third objective which represents the cost of a cooperative agent to use a distributed architecture. The cooperative agent generates a unique transfer trajectory from GEO to L1 or L2 attempting to maximize its visibility to the architecture while minimizing its total control expenditure. The inclusion of this objective demonstrates how cost may be distributed between an architecture and cooperative agents relying on it for ciritical mission support.

The final two contributions in this thesis present novel tools for analyzing the performance of architectures for use in real-world scenarios. The third contribution builds on insights from the first two contributions and presents a systematic method by which the evolution of coverage provided by an architecture is encoded in a mathematical graph. Through this graph, we demonstrate how to uncover paths that remain persistently detectable by the architecture. These paths are

then refined using collocation techniques generating trajectories that represent so-called persistent detection corridors, within which a high value asset such as a crewed spacecraft can operate within knowing it will remain detectable, even in the event of a small perturbation. This process is demonstrated on multiple architectures, and its efficacy for use in architecture optimization and mission planning is discussed.

Finally, the fourth contribution leverages reachability theory to present a framework that identifies the ability of an architecture to continuously detect a low-thrust spacecraft throughout Cislunar space. In this framework, we introduce resilience maps, a heatmap that shows how well reachable sets from any position in Cislunar space can be detected by an architecture over time. This is in contrast to a volume coverage metric, which only tracks how well a distributed architecture detects static points in Cislunar space. A resilience map is able to capture the dynamic nature of Cislunar SSA by displaying in a single plot how well an architecture's coverage evolution matches that of a low-thrust spacecraft. This framework and method of analysis is demonstrated on sample architectures, and its utility for architecture fine-tuning and resource allocation is discussed.

Dedication

For Sarah and Robert, Maggie and Anna, and Sanna.

Acknowledgements

I would first like to acknowledge my advisor, Marcus Holzinger, for his guidance and support throughout my PhD journey. His technical leadership provided me with a strong foundation upon which this thesis was constructed. Furthermore, he often gave me valuable advice and persepctives related to my career and life in general.

I would also like to thank my mentors at BAE (formerly Ball Aerospace), in particular Dr. Naomi Owens-Fahrner, who continuously advocated for the funding of this research, and always provided me with valuable feedback on my work. Others providing invaluable feedback from BAE include Josh Wysack, Jake Griesbach, Geoff Lake, Amy Bloom, and many others.

I also want to express my gratitude to all the folks in the VADeR lab and other labs at CU Boulder who have provided me with support and friendship throughout my time here. There are innumerable individuals in the aerospace department who have contributed to my growth, and I am truly thankful for their encouragement and camaraderie.

Finally, I would not be here without the unwavering support of my family and friends. From a young age I was obsessed with science and technology, and my parents fostered that interest by providing me with opportunities to explore my interests like gifting me older computers that I could safely tinker with without risk of breaking anything too important. Most importantly, they instilled in me the values and morals that guide me to this day. I am also grateful to my sisters, my first friends, with whom I shared many adventures and who continue to support me. Finally, I am forever grateful to my best friend, partner, and wife, Sanna, who supports and loves me unconditionally. I look forward to a lifetime of adventures together.

Contents

Tables

Table

Figures

Figure

Chapter 1

Cislunar Space Situational Awareness

In 2022, the White House Office of Science and Technology Policy released its first document outlining national strategic policies and goals for development of the Cislunar regime [2]. This document, titled *National Cislunar Science and Technology Strategy*, outlines how "U.S. science and technology leadership in Cislunar space will support the **responsible, peaceful, and sustainable exploration and use of Cislunar space**...by all spacefaring nations and entities" (emphasis added). Interest in the region is indeed growing, with sources indicating over 100 missions within the Cislunar regime planned for the next decade [76], spanning the spectrum from commerical mining missions on the Lunar surface, to scientific exploration, to national security missions [13]. The potential for economic growth in the region is immese, with many private companies already investing in building out capabilities to support missions such as the Artemis program [90, 76], and other companies such as Interlune [1] , which is developing capabilities for mining Helium-3 on the Lunar surface. The broad range of interest in the region among many different entities - commercial, scientific, governmental, and academic - means that Cislunar space is a key strategic domain for the United States and other nations [13].

Repeatedly emphasized by the White House [2] and industry leaders in space policy [1, 56, 13, 76, 44, 3] is how imperative it is to expand Space Situational Awareness (SSA) capabilities into Cislunar space. SSA can be defined as "the actionable knowledge required to predict, avoid, deter, operate through, recover from, and/or attribute cause to the loss and/or degradation of space

[1] `https://www.interlune.space/`

capabilities and services" [1]. Put another way, SSA aims to provide decision makers with as much relevant information as possible in order to ensure the safety, security, and sustainability of space operations. Given the cost, complexity, and difficulty of operating within the Cislunar regime, the more information decision makers can obtain, disseminate, and implement into real world problems, the better. However, information in Cislunar space is notoriously difficult to obtain, as the region represents a massive volume of space not easily observable from Earth-based sensors, which are increasingly over burdened by the growing number of spacecraft in Earth orbit. Dedicated space-based sensors are an obvious solution to this problem, but a space-based SSA architecture must be designed to account for the challenging and complex dynamics of the regime.

Holzinger & Jah [1] apply a Johari Window framework to the categorization of information in SSA domains, which identifies that all knowledge relevant to a decision maker is based on two factors: whether a fact is *known* and whether or not it is *understood*. Using this framework, they identify that ideal decision making in SSA "occurs when one knows and understands all relevant information" - this type of information is referred to as the "known-knowns." In reality, information is often incomplete, and thus falls into the three other categories of information: known-unknowns, unknown-knowns, and unknown-unknowns. Much of Cislunar SSA research explores hypothesis-driven methods which predict behaviors, falling into the known-unknowns category - things that are known to be unknown, but predictable. On the other hand, unknown-unknowns refer to unpredictable events which necessitate a reactive response. It is a goal of research in the field of Cislunar SSA, and a key concept that underpins the direction of this thesis, to identify processes and tools to transfer reactive SSA to predictive SSA [1].

A number of recent works have explored aspects of Cislunar SSA, including the design of distributed observer constellations [89, 88, 77, 9, 10, 97, 34, 43, 28, 24], the analysis of specific periodic orbits for Cislunar SSA [29, 102, 46, 48, 47, 6, 38, 94, 40, 101, 105, 103, 11, 65], and the development of tools and metrics to aid in the analysis of architectures [104, 39, 45, 35].

Even with the often-assumed dynamics of the circular-restricted three body problem (CR3BP), motion in the Cislunar regime is chaotic and often unstable, making studying SDA architectures difficult. Much of the work within the literature focuses on analyzing the performance of sensors in specific Cislunar periodic orbits with respect to SDA tasks such as initial orbit determination or custody maintenance. Vendl and Holzinger investigated the accessibility of a trapezoidal volume of space to various observers with electro-optical (EO) sensors given an apparent magnitude cut-off [94]. This work showed that a single observer's ability to detect Cislunar space objects (SOs) over long periods of time depended greatly on both the resonance of an orbit's period with the Earth-Moon synodic period, and the initial phasing along its orbit with respect to the incoming solar rays. Frueh et al. have proposed methodologies for the manual construction of Cislunar SDA architectures by placing observers in 2:1 planar resonant orbits [40]. Such a setup can be favorable in that the 2:1 resonance with the Moon can help with scheduling recurring observations of lunar-based assets, albeit with the loss of resonance with the incoming solar rays. Fowler et al. take a more general approach by using metrics attempting to estimate the numerical observability capabilities of Cislunar observers, as well as other heuristic metrics [39]. With metrics considering relative angular rates and the condition number of the empirical local observability gramian similar to [45], Fowler et al. perform a study over multiple combinations of observers in different orbits to demonstrate the utility of their metrics. Another study of interest addressed the architecture design problem within a multi-stakeholder framework [71]. Here, a tabletop simulation consisting of multiple agents with shared and differing interests is proposed. Within the simulation, stakeholders will collaborate to envision the future of Cislunar space with respect to conflicting preferences and real world limitations. This preliminary study emphasizes that multi-domain knowledge and collaboration is a key aspect of SDA in Cislunar space. Many other studies within these veins have been conducted, including work by Gupta et al. [47, 48, 46], Wilmer et al. [103, 102], Dhalke et al. [29], Bhadauria et al. [9, 10] and others, highlighting the growing interest and importantance of effective SSA in Cislunar space via distributed space-based architectures.

Another vein of research within the Cislunar SSA literature focuses on a multi-objective optimization approach to constructing optimal Cislunar SSA architectures. This involves formulating the design problem as a multi-objective optimization problem, where various conflicting objectives must be balanced. For example, one objective may be to maximize coverage of key regions in Cislunar space, while another may be to minimize the cost of deploying and maintaining the sensor network. Techniques such as genetic algorithms [34, 97, 28, 24] and linear programming techniques [77, 89] have been employed to explore the design space and identify optimal or near-optimal solutions. The range of techniques highlights how difficult the underlying problem is, as the placement of observers in Cislunar space is highly nonlinear, consisting of integer and continuous variables, and subject to various constraints. This results in a mixed-integer nonlinear programming (MINLP) problem that is difficult to solve. Patel et al. [77] formulated the problem as an integer linear programming problem by discretizing the state space and using minimizing the number of observers in a constellation with a minimum coverage requirement enforced via a set of linear constraints. Shimane et al. [89] take a similar approach, but use a facility location problem to determine the optimal placement of observers in Cislunar space. This work is especially interesting in that it attempts to tackle the placement of observers while also optimizing the tasking of individual observers to meet coverage demand, two activities that are highly coupled. Inevitably, even linear approximations of the problem require dealing with the nonlinearities of the Cislunar regime as well as the variable solar illumination over time.

Many other studies employ genetic algorithms to optimize Cislunar SSA architectures. Visonneau et al. [97] use a genetic algorithm to optimize for observer placement, relying on the accessibility metric from Vendl [94] that uses a minimum visibility based on apparent magnitude, and minimizes the number of observers and the cummulative linear stability indices of the orbits. Dhalke et al. [28] also use a genetic algorithm in a similar formulation to Visonneau. Most of these works use a similar set of objectives, such as minimizing the number of observers, maximizing coverage of the so called “cone of shame” or “Earth-Moon corridor” region, and minimizing the cummulative linear stability indices of the orbits. The cone of shame is a region in Cislunar space generally encapsulates the volume of space a spacecraft would traverse in a high energy direct transfer from the Earth to the Moon, and thus is a key region of interest for Cislunar SSA.

While these works represent significant contributions to the field of Cislunar SSA, they often rely on similar heuristic methods to model architecture performance, with most using apparent magnitude as a metric for detectability. While beneficial for high level analyses of observer performance, this approach is not able to account for individual observer capabilities such as telescope diamater, sensor noise, and other factors that significantly affect sensing capabilities. Additionally, there is a gap in the literature regarding how different entities will inevitably use Cislunar SSA architectures for their own mission objectives, and the implications of real world usage of these architectures on the design of the architecture itself. Given the immense complexities of the Cislunar regime, and the high-stakes associated with future missions within the region, it is imperative that Cislunar SSA architectures are designed to meet the needs of decision makers, mission operators, and cooperative agents across a broad spectrum of potential objectives, and that the architectures are able to adapt to unforseen events and disruptions.

There are some studies that explore how best to task Cislunar SSA architectures to meet unique mission objectives, including how to respond to disruptions across a variety of categories. An increasing number of studies attempt to identify common definitions of resilience applied to the Cislunar domain, with Hay et al. [54] partitioning resilience into four categories reflecting the origin of disruptions: environment, human, system, and support. Hay and others [106] note that the key of resilient systems is the ability of a system to minimize, respond to, recover from, and evolve from disruptions across all these categories. Not considered in the literature, however, is how to construct an architecture that is resilient to disruptions to spacecraft that are being tracked by the architecture. A potential approach to this problem is to leverage reachability theory to quantify the ability of an architecture to maintain coverage of a spacecraft in the event of a disruption to its trajectory. Reachability theory has been applied to Cislunar SSA in previous work, notably by Hall et al. [50, 52] and Schwab et al. [86] where reachable sets are used to determine the probability of detecting SOs over time, in turn informing the intelligent tasking of observers for generating precise tracks. While greatly beneficial for lower-level tasks of generating tracks, it remains to be seen how one can effectively quantify what kind of architectures are best suited to implement such tasks.

The major goal of this thesis is to develop a framework for the design and analysis of Cislunar Space Domain Awareness (SDA) architectures that can be used to support the growing number of missions in the region. In particular, this thesis aims to explore novel avenues of analysis that will aid in Cislunar SSA, from the beginning stages of planning and designing an architecture based on informative mission objectives, to the real world usage and interaction with a distributed architecture by decision makers, mission operators, and cooperative agents.

1.1 Optimal Architecture Design Using Monte Carlo Tree Search

This first contribution lays the foundation for the design and analysis of optimal Cislunar SSA architectures. Here, we propose the use of a multi-objective Monte Carlo Tree Search (MO-MCTS) algorithm to return Pareto-optimal solutions to the observer placement problem for various mission scenarios. Rather than using an apparent magnitude threshold to determine detectability, an electro-optical (EO) sensor model is used to simulate a signal-to-noise ratio (SNR) threshold for detectability. This is a more realistic representation of the Cislunar SSA environment, and expands the state space of the problem to include observer characteristics such as telescope diameter which directly impacts observer performance and cost. The MO-MCTS algorithm is used to solve two types of architecture design problems: the first seeks to maximize coverage of a predetermined Earth-Moon transfer trajectory, and the second maximizes coverage of two volumes of Cislunar space representing performance over the entire Cislunar volume and the Earth-Moon corridor. This contribution represents one of the first in-depth analysis of the Cislunar SSA architecture design problem, and provides a foundation for the proceeding contributions in this thesis.

1.2 Architecture Design for Cooperative Agents

This contribution expands on the previous contribution by introducing a novel method for the design of Cislunar SDA architectures that are tailored to the needs of cooperative agents (CAs) operating in the region. The previous contribution focused on the design of architectures that maximize coverage of a specific trajectory or volume, but did not consider how cooperative agents would interact with said architecture. As the number of Cislunar missions increases, so will the number of cooperative agents operating in the region, each with their own mission objectives and requirements. It is imperative that Cislunar SDA architectures are designed to be accessible and useful for these agents, while also satisfying high level mission objectives such as coverage and cost.

A cooperative agent is modeled as a spacecraft departing a GEO orbit and traveling to L1 or L2, whose trajectory is directly affected by the coverage profile of the architecture iteration. The cooperative agent is tasked with generating a transfer trajectory that minimizes its delta-v while maximizing its photometric detection threshold based on visual apparent magnitude as seen by the architecture. This is done by iteratively generating optimal transfer trajectories that minimize the delta-v of the cooperative agent while maximizing its photometric detection threshold. The resultant multi-objective optimization problem (MOOP) is solved using a hybrid multi-objective MO-MCTS/NSGA II algorithm, which allows for efficient exploration of the design space and identification of Pareto-optimal solutions. The solution space is analyzed using clustering methods to identify distinct architecture categories, and the tradespace shows how cost is distributed between architectures and cooperative agents. This contribution highlights the importance of considering how different stakeholders will interact with future architectures and how this interaction can inform the design of architectures and Cislunar missions.

1.3 Persistent Detection Corridors for Crewed Missions

In this contribution, we expand the study of detection corridors, introduced in the previous contribution, and present a novel tool to aid in uncovering corridors that provide minimum levels of persistent coverage, termed *persistent detection corridors* (PDCs). This work stems directly from the observervation in the previous two contributions that the distribution of coverage provided by an architecture may or may not admit opportunities for spacecraft in the Cislunar regime to remain persistently detectable by an architecture. PDCs are volumes of space where a spacecraft remains within the detectable regions generated by an SSA architecture over time. These corridors represent feasible pathways through which crewed missions and critical assets can traverse that guarantee minimum coverage requirements. A detection corridor is defined along a reference trajectory, allowing for the analysis of how state perturbations along said trajectory could potentially impact asset coverage. The identification of PDCs involves an analysis of the spatial and temporal dynamics of detectable regions resulting from observer motion and the varying solar illumination. By mapping these corridors, mission planners can rapidly enumerate trajectories that optimize for coverage, minimizing the risk associated with detection gaps.

Here, we introduce a novel method for the automated generation and analysis of detection corridors within Cislunar space for any arbitrary Cislunar SSA architecture. By leveraging these corridors, mission designers can ensure that trajectories not only meet mission objectives but also maintain the minimum coverage requirements necessary for safe human spaceflight and coverage of high-value assets. The results of this study also offer critical insights into the design and optimization of SSA architectures, providing a foundation for future mission planning that prioritizes the safety and success of crewed missions in Cislunar space.

Moreover, this work contributes to the broader understanding of how SSA architectures can designed so that they are realistically useful for cooperative agents wishing to utilize established SSA capabilities. The ability to maintain persistent detection of assets in Cislunar space is not only a technical challenge but also a strategic imperative as interest in Cislunar missions expands. By addressing these challenges, our methods set the stage for more resilient and reliable Cislunar SSA operations, paving the way for a safe and sustained human presence on the Moon and beyond.

1.4 Resilience of Architectures Using Reachability

The final contribution of this thesis focuses on the resilience of Cislunar SSA architectures to disruptions in the spacecraft being tracked by the architecture. As the number of missions in Cislunar space increases, so does the need for resilient architectures that can remain performant under unforseen circumstances. Resilience is a key aspect of any SSA architecture, as it allows for continued operation and coverage even in the face of unexpected events or disruptions. This contribution leverages reachability theory to quantify the resilience of Cislunar SSA architectures to astrodynamic disruptions in spacecraft being tracked by the architecture.

In missions with high value assets, such as crewed missions to the Moon, it is critical to ensure that spacecraft can be detectable by mission operators at all times, even in the event of loss of control or deviation from a reference trajectory. Observers with electro-optical (EO) sensors rely on solar illumintation to detect space objects (SOs), and thus the distribution of coverage an architecture provides evolves over time as a function of the relative motion of observers and solar illumination geometry. This evolution ocassionally leads to a mismatch in coverage and spacecraft dynamics. Consequently, typical metrics for determining Cislunar architecture performance, such as volume coverage or coverage of a select number of predetermined trajectories lacks the nuance needed to encapsulate the dynamic nature of Cislunar SSA. By leveraging reachability theory, this potential mismatch of dynamics can be exposed, allowing for more localized analysis of architecture performance.

This contribution leverages the many advances in reachable set computation methods to provide a foundation for quantifying the resilience of Cislunar SSA architectures to astrodynamic disruptions in the spacecraft being tracked. This novel framework tracks the coverage of reachable sets over time, resulting in heat maps that illustrate the temporal evolution of coverage and its alignment with the CR3BP dynamics.

1.5 Contributions and Outline

The contributions in this thesis represent a significant step forward in the design and analysis of Cislunar SSA architectures, and provide a variety of novel tools to aid in such. Chapter 2 outlines the first contribution, which formulates the architecture design problem as a multi-objective optimization problem and is solved using an implementation of the multi-objective Monte Carlo Tree Search (MO-MCTS) algorithm. Chapter 3 expands on the previous contribution by introducing a novel method for the design of Cislunar SDA architectures that are tailored to the needs of cooperative agents (CAs) operating in the region. Chapter 4 introduces a novel method for the automated generation and analysis of persistent detection corridors (PDCs) within Cislunar space for any arbitrary Cislunar SSA architecture, which can be used in both architecture optimization and for mission planning within the regime. Finally, Chapter 5 leverages reachability theory to quantify the resilience of Cislunar SSA architectures to astrodynamic disruptions in spacecraft being tracked by the architecture.

1.6 Publications

The research presented here has been presented in a number of publications, including journals and conference papers. The following sections outline these publications.

1.6.1 Journal Papers

- Michael Klonowski, Marcus J. Holzinger, and Naomi Owens Fahrner. Optimal Cislunar Architecture Design Using Monte Carlo Tree Search Methods. The Journal of the Astronautical Sciences, 70(3):17, June 2023.

- Michael Klonowski, Naomi Owens-Fahrner, Casey Heidrich, and Marcus Holzinger. Cislunar space domain awareness architecture design and analysis for cooperative agents. The Journal of the Astronautical Sciences, 71(5):47, 2024.

- Michael Klonowski, Casey Heidrich, Naomi Owens-Fahrner, and Marcus J. Holzinger. Persistent Detection Corridors for Crewed Missions and Cislunar Space Situational Awareness. The Journal of Spacecraft and Rockets. **Submitted for review, April 2025.**

- Michael Klonowski, and Marcus J. Holzinger. Resilience of Architectures for Cislunar Space Situational Awareness Using Low-Thrust Reachable Sets. The Journal of Spacecraft and Rockets. **Submitted for review, July 2025.**

1.6.2 Conference Papers

- Michael Klonowski, Marcus J. Holzinger, and Naomi Owens Fahrner. Optimal Cislunar Architecture Design Using Monte Carlo Tree Search Methods. In The Advanced Maui

Optical and Space Surveillance Conference, volume 70, page 17, September 2022.

- M. Klonowski, N. O. Fahrner, C. Heidrich, and M. Holzinger. Robust Cislunar Architecture Design Optimization for Cooperative Agents. In Proceedings of the Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference, page 15, September 2023.

- Michael Klonowski, Casey Heidrich, Naomi Owens-Fahrner, and Marcus J Holzinger. Analysis of Persistent Detection Corridors for Cislunar Space Situational Awareness. In The Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference, 2024.

Chapter 2

Background

2.1 Architectures for Cislunar Space Situational Awareness

2.1.1 The Circular Restricted Three-Body Problem

Throughout this thesis, we model the complex dynamics of Cislunar space using the Circular Restricted Three-Body Problem (CR3BP). The CR3BP is a simplified model of the gravitational interactions between three bodies, where two of the bodies are massive and the third body is a negligible mass. In this case, the Earth and the Moon are the two primary bodies, and the third body is a spacecraft or space object (SO) of negligible mass. The CR3BP assumes that the two primaries are in circular orbits around their common center of mass (barycenter) unaffected by the third body. Because of these assumptions, the CR3BP is an autonomous system, meaning that the equations of motion do not depend on time, allowing the system to be described in a rotating frame, with the Earth and Moon stationary. While other higher-fidelity models exist, such as the bi-circular restricted four-body problem (BR4BP), the CR3BP is widely used due to its simplicity and useful properties, and in many of the novel algorithms used in this thesis, the Cislunar dynamics can be modeled with any model according to an analyst's needs.

As is common in astrodynamics, the distance, time, and mass quantities are nondimensionalized to reduce the potential for poor conditioning in numerical simulations. The characteristic length is set to the distance between the Earth and the Moon, or the mean semi-major axis of the Moon's orbit about the Earth:

$$l^* = 3.9860044 \times 10^5 \text{ km} \tag{2.1}$$

The characteristic mass is set as the sum of the masses of the two primaries, or the mass of the Earth plus the mass of the Moon, which leads to the nondimensionalized mass ratio, μ:

$$\mu = \frac{M_M}{M_E + M_M} \approx 0.012150584... \tag{2.2}$$

The characteristic time is set such that in the nondimensionalized system, the period of the Moon's orbit is 2π TU

$$t^* = \left(\frac{l^3}{\tilde{G}(M_E + M_M)} \right)^{1/2} = 3.75190261 \times 10^5 \text{ s} \approx 4.342 \text{ days} \tag{2.3}$$

where $\tilde{G}$ is the universal gravitational constant. As a result of these characteristic quantities, the Earth and Moon lie on the x axis at $x = -\mu$ and $x = 1-\mu$, respectively. Throughout this thesis, we often use the nondimensional quantities for time and distance, marked as TU and DU, respectively. Where appropriate, some quantities are discussed in their dimensional form.

The equations of motion for the CR3BP can be expressed in a compact form via the pseudo-potential, U, which captures the effective potential of the system including the standard gravitational potential and the centrifugal force due to the rotating frame:

$$U = \frac{1}{2}\left(x^2 + y^2\right) + \frac{(1-\mu)}{r_1} + \frac{\mu}{r_2} \tag{2.4}$$

$$r_1 = \sqrt{(x+\mu)^2 + y^2 + z^2} \tag{2.5}$$

$$r_2 = \sqrt{(x-1+\mu)^2 + y^2 + z^2} \tag{2.6}$$

$$\tag{2.7}$$

where r_1 and r_2 are the distances from the spacecraft to the Earth and Moon, respectively. The equations of motion can then be expressed as the following system of second-order ordinary differential equations (ODEs):

$$\ddot{x} = \frac{\partial U}{\partial x} + 2\dot{y} \tag{2.8}$$

$$\ddot{y} = \frac{\partial U}{\partial y} - 2\dot{x} \tag{2.9}$$

$$\ddot{z} = \frac{\partial U}{\partial z} \tag{2.10}$$

where $\dot{x}$, $\dot{y}$, and $\dot{z}$ are the time derivatives of the position coordinates in the rotating frame. In vector form, the second order ODE is

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} \dot{\boldsymbol{r}} \\ \ddot{\boldsymbol{r}} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \frac{\partial U}{\partial x} + 2\dot{y} \\ \frac{\partial U}{\partial y} - 2\dot{x} \\ \frac{\partial U}{\partial z} \end{bmatrix} \tag{2.11}$$

The assumptions in the CR3BP admit five equilibrium points where a spacecraft can remain stationary with zero velocity unperturbed. These equilibrium points are often referred to as *Lagrange points* or *libration points*, two of which can be solved for analytically (L4 and L5) and three of which must be solved for numerically. The Lagrange points within the Earth-Moon CR3BP are defined in Table 2.1 and shown in Figure 2.1.

Table 2.1: Lagrange points in the Earth-Moon CR3BP. The coordinates are given in nondimensionalized units (DU) in the rotating frame.

Lagrange Point	x (DU)	y (DU)	z (DU)
L1	0.836915	0.0	0.0
L2	1.155682	0.0	0.0
L3	-1.005063	0.0	0.0
L4	0.487849	0.866025	0.0
L5	0.487849	-0.866025	0.0

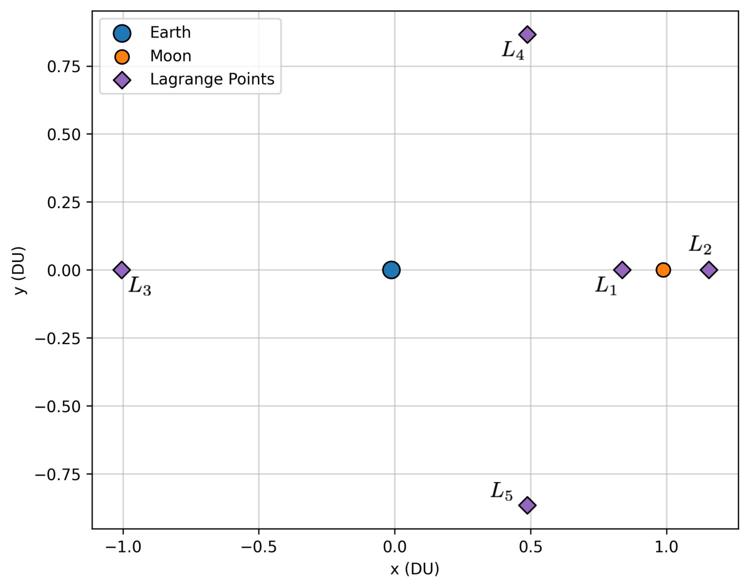


Figure 2.1: Lagrange points in the Earth-Moon CR3BP rotating nondimensional frame.

In the CR3BP, there exists a single constant of motion C_J which is a constant quantity along a trajectory in the CR3BP. This constant, the Jacobi constant, is defined as

$$C_J = 2U - \dot{x}^2 - \dot{y}^2 - \dot{z}^2 = 2U - v^2 \tag{2.12}$$

where $v^2 = \dot{x}^2 + \dot{y}^2 + \dot{z}^2$. The Jacobi constant is a measure of the total energy of the system, and by abuse of language in this thesis, we often refer to the Jacobi constant as the *Jacobi energy* of a trajectory. The Jacobi energy is extremely useful in analysis as it allows for the categorization of regions within Cislunar space. For example, given that the pseudo-potential is only a function of position, we can select a Jacobi energy value C_J^* and solve for v^2 over a meshgrid of positions in the rotating frame. The manifold where $v^2 = 0$ is referred to as the *zero-velocity surface* and demarks the boundary of regions of Cislunar space that are accessible to a spacecraft with Jacobi energy C_J^*. Thus, regions where $v^2 < 0$ are inaccessible to a spacecraft with Jacobi energy C_J^*, while regions where $v^2 > 0$ are accessible. The Jacobi constant is often used as a first principles informed method of sampling velocities for trajectory generation in the CR3BP, as it restricts the velocities to those that are physically realizable in the system. An example of these regions is shown for increasing

Jacobi energies in Figure 2.2.

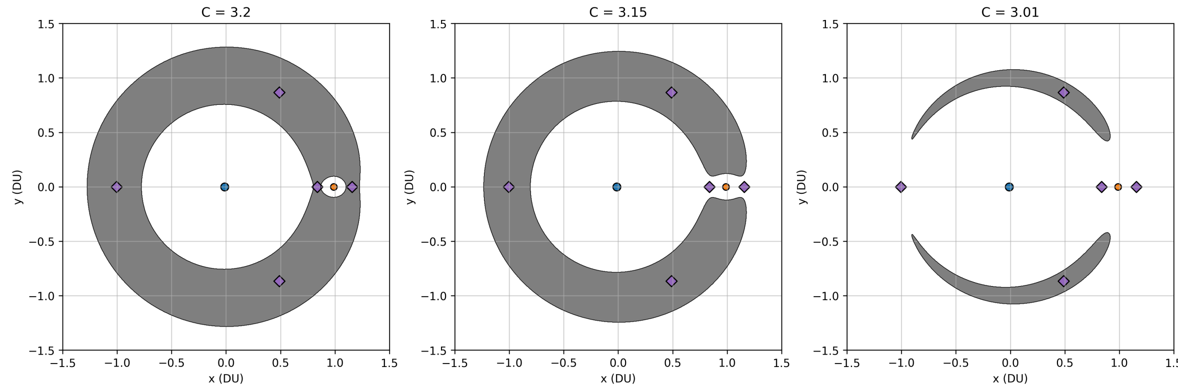


Figure 2.2: Zero-velocity surfaces and inaccessible regions over increasing energy. The zero-velocity surface is shown in black, with inaccessible regions shaded in grey.

2.1.1.1 Periodic Orbits

The CR3BP admits a wide variety of fundamental solutions including periodic orbits that exactly repeat after a fixed period. There are an infinite number of periodic orbits in the CR3BP, but since there are no analytic solutions to the CR3BP, these orbits must be solved for numerically. Most often, a shooting method is used to solve a boundary value problem (BVP) to obtain a periodic orbit starting near an equilibrium point or from a bifurcation from an existing orbit. Then, additional memebers of an orbit family can be obtained using continuation methods. A number of procedures for computing periodic orbits are discussed in detail in [36].

There are many orbital families discussed throughout the literature for various purposes, each with varying geometries, periods, and stability properties. The Jet Propulsion Laboratory (JPL) [1] maintains a massive database of precomputed periodic orbits in the Earth-Moon CR3BP, from which many of the periodic orbits discussed in this thesis are obtained. Additional periodic orbits are generated using a multiple shooting method. Orbits from the following families are used throughout this thesis:

- L1 and L2 Lyapunov, Halo, Axial, and Vertical

[1] `https://ssd.jpl.nasa.gov/tools/periodic_orbits.html`

- L4 and L5 Axial, Planar, and Long
- Distant Retrograde and Prograde, Low Prograde
- 1:2, 1:3, 2:3, 3:2 Resonant Orbits (resonance with the CR3BP period)
- Dragonfly and Butterfly Orbits
- GEO Orbit

These orbits are shown in Figure 2.3 with initial conditions reported in Appendix **??**.

2.1.2 Illumination and Pointing Constraints

Within this thesis, we consider illumination and pointing constraints as they relate to the ability of observers to detect SOs in Cislunar space. These constraints are widely used across the field, and are taken from [25, 94, 35]. A SO is only non-illuminated when it is in the Earth or Moon's shadow. To check for illumination, the positions of the SO, Earth, Moon, and Sun within the CR3BP rotating frame are $\boldsymbol{r}_{SO}$, $\boldsymbol{r}_E$, $\boldsymbol{r}_M$, $\boldsymbol{s}$. The SO is considered illuminated if the following conditions are met:

$$\|\hat{\boldsymbol{s}} \times (\boldsymbol{r}_{SO} - \boldsymbol{r}_E)\| > R_E \textbf{ or } \hat{\boldsymbol{s}} \cdot (\boldsymbol{r}_{SO} - \boldsymbol{r}_E) < 0 \tag{2.13}$$

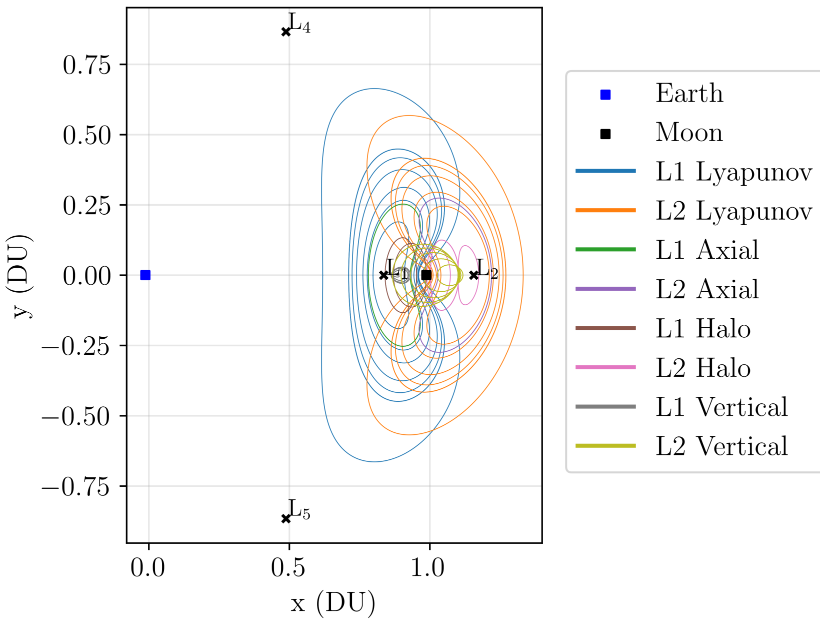
and

$$\|\hat{\boldsymbol{s}} \times (\boldsymbol{r}_{SO} - \boldsymbol{r}_M)\| > R_M \textbf{ or } \hat{\boldsymbol{s}} \cdot (\boldsymbol{r}_{SO} - \boldsymbol{r}_M) < 0 \tag{2.14}$$

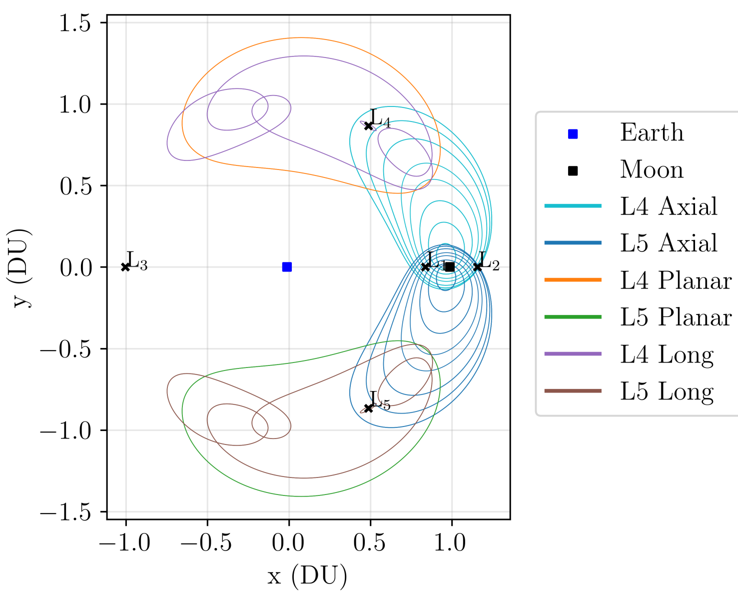
where $\hat{\boldsymbol{s}}$ is the unit vector pointing from CR3BP system barycenter to the sun defined as

$$\hat{\boldsymbol{s}} = \frac{\boldsymbol{s}}{\|\boldsymbol{s}\|} \tag{2.15}$$

and R_E and R_M are the radii of the Earth and Moon, respectively. The first expression in Eqn. 2.13 checks if the SO is outside of a cylinder with radius R_E extending from the Sun through the Earth, while the second expression checks if the SO is in front of the Earth relative to the Sun.

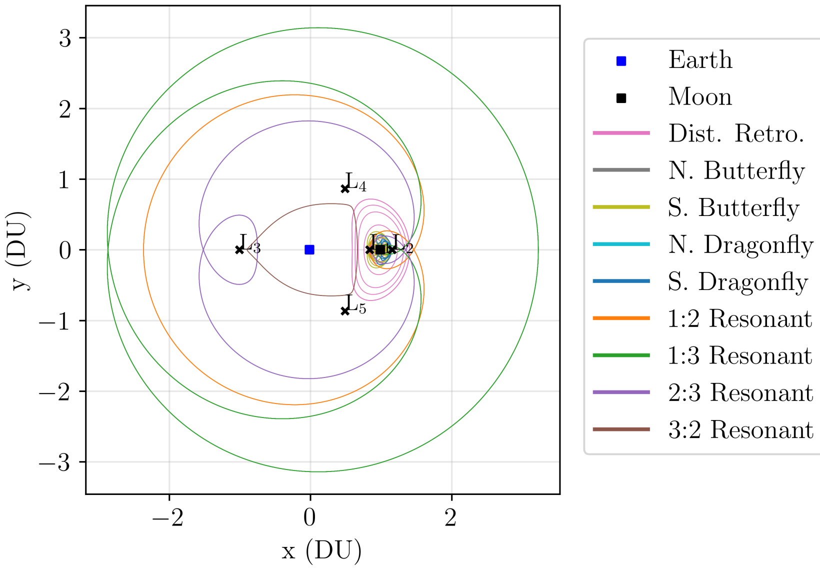


(a)

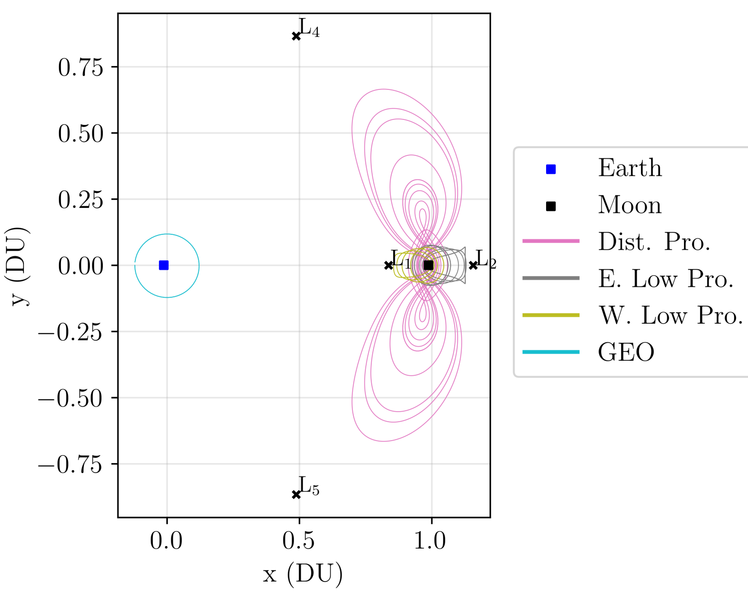


(b)

Figure 2.3: Candidate observer orbits in the CR3BP. Split into four figures for clarity (continued on next page).



(c)



(d)

Figure 2.3: Candidate observer orbits in the CR3BP. Split into four figures for clarity.

The same logic applies to the Moon in Eqn. 2.14. If both conditions are met, the SO is considered illuminated. If either condition is not met, the SO is considered non-illuminated. This illumination condition can be seen in Figure 2.4 adopted from [35].

Pointing constraints are also considered in this thesis, representing the inability of an observer to detect a target if the target is within some angle of the Earth, Moon, and Sun, as well as additional angular exclusion zones around these bodies. Here, the observer position is represented as $\boldsymbol{r}_{obs}$ and the pointing constraints are defined as follows:

$$\arccos\left(\frac{\boldsymbol{e}\cdot\boldsymbol{v}}{\|\boldsymbol{e}\|\|\boldsymbol{v}\|}\right) \leq \epsilon + \beta_E \tag{2.16}$$

where $\boldsymbol{e} = \boldsymbol{r}_E - \boldsymbol{r}_{obs.}$, $\boldsymbol{v} = \boldsymbol{r}_{SO} - \boldsymbol{r}_{obs.}$, β_E defines the additional angular exclusion extended from the Earth, and ϵ defines the angular exclusion as a result of the Earth defined as

$$\epsilon = \arctan\left(\frac{R_E}{\|\boldsymbol{r}_{obs} - \boldsymbol{r}_E\|}\right) \tag{2.17}$$

Similarly, for the Moon's exclusion cone, the SO is occluded if

$$\arccos\left(\frac{\boldsymbol{m}\cdot\boldsymbol{v}}{\|\boldsymbol{m}\|\|\boldsymbol{v}\|}\right) \leq \eta + \beta_M \tag{2.18}$$

where $\boldsymbol{m} = \boldsymbol{r}_M - \boldsymbol{r}_{obs.}$, β_M defines the additional angular exclusion extended from the Moon, and η defines the angular exclusion as a result of the Moon defined as

$$\eta = \arctan\left(\frac{R_M}{\|\boldsymbol{r}_{obs} - \boldsymbol{r}_M\|}\right) \tag{2.19}$$

Finally, we also define an exclusion cone of 10° with respect to the direction of the solar illumination in the rotating frame ($\hat{\boldsymbol{s}}$). Occlusion in this region is defined by

$$\arccos\left(-\hat{\boldsymbol{s}}\cdot\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}\right) \leq 10^\circ. \tag{2.20}$$

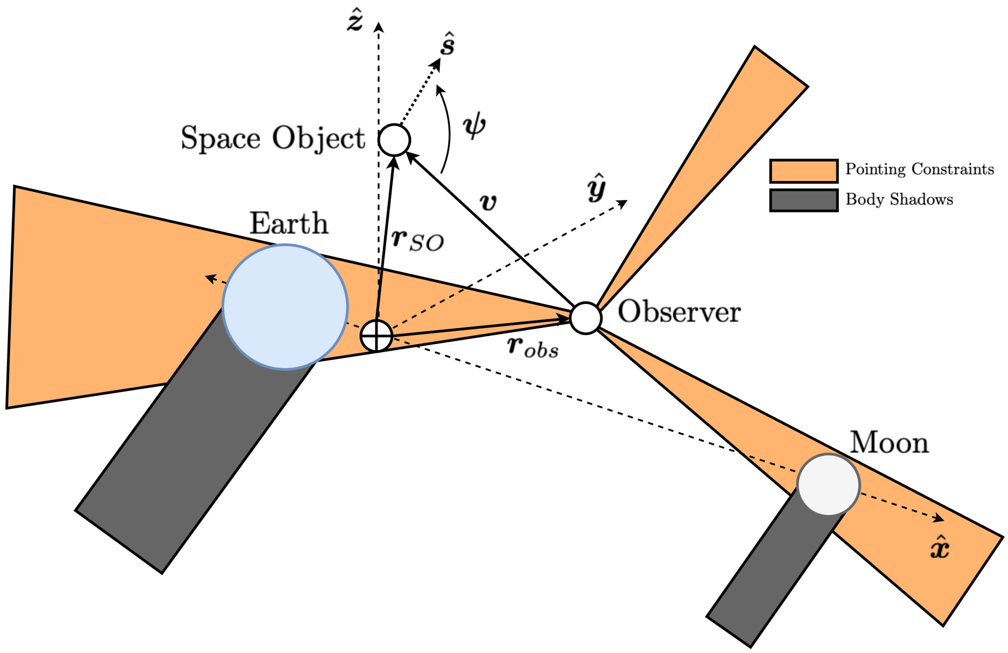


Figure 2.4: Illumination and pointing constraints for a space object in Cislunar space. The SO is illuminated if it is outside the Earth and Moon's shadows, and can be detected by an observer if it is outside of the angular exclusion cones, representing the pointing constraints.

2.1.3 Sensor Modeling

To simulation detections of space objects in Cislunar space, a sensor model is leveraged based off of the work from Coder [25], which allows for simulating a signal-to-noise ratio (SNR) based on observer capabilities. In this thesis, as in the majority of the Cislunar SSA literature, a space object is modeled as a $d = 1m$ diameter Lambertian sphere with specular reflectance $a_{spec} = 0.0$ and diffuse reflectance $a_{diff} = 0.2$ (assumed constant across all wavelengths) as in [94]. First, the solar phase angle ψ, defined as the angle between the observer's line of sight to the SO and the Sun, is calculated as

$$\psi = \arccos\left(\hat{\boldsymbol{s}} \cdot \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}\right) \tag{2.21}$$

the diffuse phase angle function is written as

$$p_{diff}(\psi) = \frac{2}{3\pi}[\sin\psi + (\pi - \psi)\cos\psi] \tag{2.22}$$

Then, given the specular and diffuse components of object reflectance a_{spec} and a_{diff}, the apparent magnitude of the SO can be estimated as

$$m_{SO} = m_S - 2.5\log 10\left(\frac{d^2}{\|\boldsymbol{v}\|^2}\left[\frac{a_{spec}}{4} + a_{diff}p_{diff}(\psi)\right]\right) \tag{2.23}$$

where here $m_S = -26.74$, resulting in the apparent magnitude of an SO as a function of the solar phase angle and relative observer and SO positions.

Using the apparent magnitude, we can obtain the photon flux density Φ_{SO} of the SO as

$$\Phi_{SO} = \Phi_0 \times 10^{-0.4m_{SO}} \tag{2.24}$$

$$\Phi_0 = \frac{M_0\lambda}{hc} \tag{2.25}$$

where $M_0 = 3.67 \times 10^{-9}$ Wm^{-2} is the solar existence of a magnitude 0 object, λ is the wavelength of interest (here chosen to be $\lambda = 625$nm), and h and c are Planck's constant and the speed of light respectively.

Then, the photon flux captured by the optical system in e^{-}/s is given as

$$q_{SO} = \Phi_{SO} \tau_{atm} \tau_{opt} \left(\frac{\pi D^2}{4} \right) \mathrm{QE} \tag{2.26}$$

Here, τ_{atm} and τ_{opt} are the transmittance of the atmosphere and optical components defined in $\tau \in (0, 1]$. The aperture diameter of the telescope used by the observer is given as D, and the quantum efficiency of the camera is given as QE $\in (0, 1]$. Furthermore, by setting $\tau_{atm} = 1$ Equation 2.26 is tailored specifically for a space-based observer, as used in this work. Finally, given an integration time t we can estimate the SNR of an SO in an exposure as

$$\mathrm{SNR} = \frac{q_{SO} t}{\sqrt{q_{SO} t + m(1 + \frac{m}{z})[(q_{p,\mathrm{sky}} + q_{p,\mathrm{dark}})t + \frac{\sigma_r^2}{n^2}]}} \tag{2.27}$$

In this equation, m represents the amount of pixels occupied by the SO, z is the amount of pixels used to estimate the background noise, n is a binning factor (set to 1 in this work), and σ_r^2 is the sensor read noise. Equation 2.27 also considers the photon flux per pixel associated with the background sky irradiance, $q_{p,\mathrm{sky}}$ and the dark current per pixel from the camera sensor, $q_{p,\mathrm{dark}}$. We set $q_{p,\mathrm{sky}} = 0$ since the simulated observers used here are space-based. Finally, the integration time t is set to a constant value of $t = 30$ seconds. Throughout this thesis, the SNR is used as a proxy for the ability of an observer to detect an SO, within an observer's field of regard (FoR). A threshold of SNR ≥ 6 is used to determine if an SO is detectable.

The electro-optical sensors used here are based off of an off-the-shelf Finger Lakes Instruments Kepler KL4040 CMOS camera, with capabilities and relevant SNR calculation parameters shown in Table 2.2.

Table 2.2: Sensor specifications and relevant parameters for calculating SNR. Based on the Finger Lakes Instruments Kepler KL4040 CMOS camera.

Parameter	Symbol	Value
Quantum Efficiency	QE	70% (@625nm)
Sensor Read Noise	σ_r	10 e^-
Dark Current	$q_{p,dark}$	0.2 e^-/s
SO Pixels	m	1
BG Noise Pixels	z	10
Binning Factor	n	1
Integration Time	t	30 s
Atm. Trans.	τ_{atm}	1.0
Opt. Trans.	τ_{opt}	0.75

2.1.4 Distributed Architectures

As previously discussed, constellations consisting of multiple observers distributed throughout Cislunar space are neccessary for effective SSA. Throughout this thesis, various models of distributed architectures are considered for their cost and performance in Cislunar SSA activities. These architectures consist of potentially multiple observers, with each observer defined by a tuple:

$$Obs_i = [Family,\ Index,\ Phasing,\ Telescope\ Diameter] \tag{2.28}$$

Here, $Family$ indicates the type of orbit, $Index$ refers to the specific orbit in the family, $Phasing \in [0, 1]$ is the initial position of the orbit as a fraction of its period, and $Telescope\ Diameter \in [200mm, 300mm, 500mm]$ is the diameter of the observer telescope. Each observer's state can be evaluated at time t as $\boldsymbol{x}_{Obs_i}(t)$. The initial phasing is enforced at an arbitrarily chosen simulation epoch T_{epoch}, and in results throughout this thesis is often reported as an angle $\alpha_{phase} = 360° \times Phasing$. Then, an architecture $\boldsymbol{X}$ can be defined as a set of observers

$$\boldsymbol{X} = \{Obs_1, Obs_2, ...\} \tag{2.29}$$

As discussed throughout the literature [94, 46, 40, 89], periodic orbits with periods approximately resonant with the Earth-Moon synodic period benefit from a predictable repeating geometry with respect to the incoming solar illumination, when phased appropriately in their orbits. An orbit with $m : n$ resonance with the synodic period indicates that the orbit completes m revolutions for every n synodic periods. Thus a $2 : 1$ synodic resonant orbit completes two revolutions for every synodic period. In an architecture with i observers with varying resonances of $m_1 : n_1, ..., m_i : n_i$, the period of the entire configuration (i.e. the time for the relative initial geometries to repeat) is $lcm(n_1, ..., n_i)$ synodic periods, where $lcm(\cdot)$ is the least common multiple. In this thesis, candidate orbits include approximate resonances of $1 : 1$, $1 : 2$, $1 : 3$, $1 : 4$, $2 : 1$, $3 : 1$, $3 : 2$, $4 : 3$, $5 : 2$, $5 : 3$, $5 : 4$, and $6 : 5$, resulting in the maximal possible period of an architecture being 60 synodic periods or about five years. In reality, this assumption of resonance is imperfect and relative geometries will not exactly repeat. This assumption will be addressed in more detail in Chapters 5 and 6.

2.2 Multi-Objective Optimization

Multi-objective optimization (MOO) is the process of optimizing two or more often competing objective functions. Marler et al [70] provides an excellent discussion of MOO theory, terminologies, and algorithms, so only a discussion of the relevant definitions will be provided here. Given a vector of design or *decision* variables $\boldsymbol{x} \in E^n$, where n is the number of independent variables, a MOO problem (MOOP) can be defined as (adopted from [70])

$$\min_{\boldsymbol{x}} \boldsymbol{F}(\boldsymbol{x}) = [F_1(\boldsymbol{x}), F_2(\boldsymbol{x}), ..., F_k(\boldsymbol{x})]^T \tag{2.30}$$

where $\boldsymbol{F}(\boldsymbol{x}) \in E^k$ is a vector of k objective functions where $F_i(\boldsymbol{x}) : E^n \rightarrow E^1$. The *feasible design space* $\boldsymbol{X}$ is the set of decision variables that satisfy potential constraints (often presented as $g(\boldsymbol{x})$ and $h(\boldsymbol{x})$). The *feasible criterion space* is the set $\{\boldsymbol{F}(\boldsymbol{x}) | \boldsymbol{x} \in \boldsymbol{X}\}$ that represents the objective space that is achievable through the feasible design space. Typically, such a problem has no single optimal value $\boldsymbol{x}^*$ that satisfies Eqn. 2.30, and as such it is necessary to determine a set of points that meet some condition of optimality. Most widely used in the field is *Pareto optimality*, defined as follows (adopted from [70])

Definition 1 (Pareto Optimal) A solution $\boldsymbol{x}^* \in \boldsymbol{X}$ is **Pareto optimal** if and only if there does not exist another point $\boldsymbol{x} \in \boldsymbol{X}$, such that $\boldsymbol{F}(\boldsymbol{x}) \leq \boldsymbol{F}(\boldsymbol{x}^*)$, and $F_i(\boldsymbol{x}) < F_i(\boldsymbol{x}^*)$ for at least one $F_i \in \boldsymbol{F}$. The set of Pareto optimal solutions evaluated in $\boldsymbol{F}(\cdot)$ is referred to as the **Pareto set** (P), and $\boldsymbol{x}^*$ is said to be **non-dominated** in P. Furthermore, $\boldsymbol{x}^*$ **dominates** $\boldsymbol{x}$ in P.

Thus, the Pareto set, or *Pareto front*, contains all the non-dominated vectorial objectives evaluated at the non-dominated solutions. For ease of reference, this set can be expressed as a dictionary object $P = \{\boldsymbol{x}_i^* : \boldsymbol{F}(\boldsymbol{x}_i^*)\}$ for all non-dominated $\boldsymbol{x}_i^* \in \boldsymbol{X}$. By abuse of language, $\boldsymbol{F}(\boldsymbol{x}_i^*)$ is also said to be non-dominated in P. Typically, when referring to *points* in the Pareto set, this is referring to the solutions $\boldsymbol{x}^*$ evaluated in objective space $\boldsymbol{F}(\boldsymbol{x}^*)$. It should be noted that a MOOP can have objectives meant to be minimized or maximized, in which case objective functions in Eqn. (2.30) can be multiplied by -1 to convert minimization to maximization.

There are a multitude of methods for solving or approximating the true Pareto set for various problems with performance and convergence highly dependent on problem type and complexity. Marler et al. provide detailed discussions and comparisons between some of the most common methods applied to the engineering domain [70].

2.2.1 Multi-Objective Optimization Algorithms

Two algorithms are used in concert throughout this thesis to solve MOO problems. First, these algorithms are explained in detail, and then the specific implementation used in this thesis are discussed.

2.2.1.1 NSGA-II

A genetic algorithm (GA) is a type of evolutionary algorithm that mimics the process of natural selection to solve optimization problems. In a GA, a population of candidate solutions is evolved over generations through the processes of selection, crossover, and mutation. The fitness of each candidate solution is evaluated based on an objective function, and the best performing candidates are selected to form the next generation. Crossover combines two parent solutions to create offspring solutions, while mutation introduces random changes to individual solutions to maintain genetic diversity. GAs are widely used across various domains due to their ability to explore large state spaces without requiring gradient information, making them suitable for complex and nonlinear multi-objective optimization problems [70].

Nondominated Sorting Genetic Algorithm II (NSGA-II), first introduced in [31], is a popular multi-objective evolutionary algorithm that implements sorting of individuals based on their levels of nondominance. It also utilizes elitism, or the process by which best performing individuals remain unaltered, while others go through the processes of crossover and mutation. NSGA-II has been shown to perform well with highly nonlinear MOOPs. In this work, we utilize the Deap

Python package [37] for the implementation of NSGA-II.

2.2.1.2 Markov Decision Processes

Recent advances in computer processing power and reinforcement learning methods have prompted the exploration of novel engineering problems within the framework of sequential decision processes. A Markov Decision Process (MDP) is a mathematical model that represents a discrete time sequential decision process. An MDP is defined by the tuple (S, A, T, R, γ) where

- S is the state space
- A is the action space
- $T : S \times A \times S \rightarrow [0, 1]$ is the state transition function
- $R : S \times A \times S \rightarrow \mathcal{R}$ is the reward function from selecting an action a at state s and ending up in state s'
- γ is the discount factor used for weighting immediate rewards higher than distant rewards

Relevant to this thesis is that the scalar reward function can be altered to include multiple objectives so that $\boldsymbol{R} : S \times A \times S \rightarrow \mathcal{R}^k$ where $k \geq 1$ is the number of objectives.

A key trait of an MDP is that it adheres to the *Markov assumption*. A fundamental property of MDPs, this assumes that the next state in a sequential process depends only on the current state and action, and not on any previous action or state [66]. This allows for a dramatic reduction in complexity because one does not need to store an entire sequence history in order to reason about future states. It also provides a foundation for proving the convergence of different algorithms.

The solution to an MDP can be represented as a policy function $\pi^*(s) : S \rightarrow A$ that maximizes expected utility $U^{\pi^*}(s)$ for all $s \in S$. *Utility*, also referred to as the *value function*, represents the maximum expected discounted cumulative reward over the available actions at a state

$$U(s) = \max_a \left(R(s,a) + \gamma \sum_{s'} T(s' \ s,a) U(s') \right) \tag{2.31}$$

$$= \max_a Q(s,a) \tag{2.32}$$

Here, $Q(s,a)$ is defined as the *action value function*, which represents the expected discounted cumulative reward of taking action a in state s. Note that in the case of a vector reward function ($k \geq 1$), the value and action value functions are similarly sized vectors. From the action value function, the policy can be extracted as

$$\pi(s) = \arg\max_a Q(s,a) \tag{2.33}$$

In simple MDPs, solving for the optimal policy $\pi^*(s)$ can be accomplished through value or policy iteration, which recursively estimates the value or action value functions, respectively, and is guaranteed to converge in finite time given sufficient computing resources. However, in problems with larger state and action spaces, the curse of dimensionality appears and makes finding exact solutions intractable. Thus, methods are needed to generate approximations of the optimal policy. For further discussion on MDPs, the reader is referred to [66].

2.2.1.3 Monte Carlo Tree Search

Monte Carlo Tree Search (MCTS) is a tree search algorithm that balances exploration and exploitation of the state-action space of an MDP to efficiently estimate the action value function of a root node [23]. The bandit-based Upper Confidence Bounds Applied to Trees (UCT) as proposed by [67] is nearly synonymous with MCTS, as it has become the dominant mechanism for balancing exploration and exploitation across the field. While only a brief overview of the steps of MCTS is provided here, there are myriads of articles and reviews on variations and applications of MCTS [92, 85, 83, 84, 7, 19, 55, 72].

There are a few fundamental concepts to understand that underpin the MCTS methodology:

(1) **Tree Structure** - *Nodes* in the tree represent states in the MDP. The tree is initialized with a *root node*. A node in the tree can have any number of child nodes (*children*), which represent the resultant state from taking an action at the *parent node*. A node with no children is called a *leaf node*.

(2) **Selection (UCT)** - Starting from the root node, traverse to children by selecting an action using the UCT equation [66, 67]

$$a^* = \arg\max_a \left(Q(s,a) + c\sqrt{\frac{\log N(s)}{N(s,a)}} \right) \tag{2.34}$$

where $N(s)$ is the number of times the node corresponding to state s has been visited, and $N(s,a)$ is the number of times action a has been selected at state s. This equation is critical to the success of MCTS because it balances the exploration of parts of the tree that have not been visited much with the exploitation of promising sections that have been visited previously.

For an MDP with a vector reward function (and value and action value functions), this equation must be modified. Wang, et al. [98] propose a modified UCT algorithm that uses the hypervolume indicator (HVI) [82] to guide selection. Given the Pareto set P as defined in Def. 1, the multi-objective UCT equation selects an action by

$$a^* = \arg\max_a \left(HV(\boldsymbol{Q}(s,a) \cup P) - d(\boldsymbol{Q}(s,a)) + c\sqrt{\frac{\log N(s)}{N(s,a)}} \right) \tag{2.35}$$

$$d(\boldsymbol{Q}(s,a)) = \begin{cases} 0 & \boldsymbol{Q}(s,a) \text{ is non-dominated in } P \\ ||\boldsymbol{Q}(s,a) - \boldsymbol{C}(\boldsymbol{Q}(s,a), P)|| & \text{otherwise} \end{cases} \tag{2.36}$$

where $\boldsymbol{C}(\boldsymbol{Q}(s,a), P)$ is the projection of $\boldsymbol{Q}(s,a)$ onto the convex hull of non-dominated points in the Pareto set. A property of the HVI is that when $\boldsymbol{Q}(s,a)$ is *dominated* in P, $HV(\boldsymbol{Q}(s,a) \cup P) = HV(P)$. Conversely, if $\boldsymbol{Q}(s,a)$ is *non-dominated* in P, $HV(\boldsymbol{Q}(s,a) \cup$

$P) > HV(P)$. Thus, the penalty term $d(\cdot)$ is needed to reflect the approximate Euclidean distance from the dominated point to the Pareto front. Therefore, Eqn. 2.35 returns an action a^* that balances exploration and exploitation of previously visited nodes that may increase the HVI of the Pareto front. For rewards defined in $[0, 1]$, the range of the HVI is similarly constrained and the constant c in Eqns. 2.34 and 2.35 is typically chosen to be $\sqrt{2}$.

(3) **Expansion (Progressive Widening)** - Selection continues until a node is selected that has no children (a *leaf node*) or is not fully expanded. At this point, if possible, a new action is selected heuristically or stochastically, the action is simulated in the environment, and the resultant state s_i at tree depth i is added to the parent as a child and leaf node. A node is considered fully expanded if all of its possible children have been expanded. In MDPs with large action spaces, or large *branching factors*, requiring a node to be fully expanded before continuing selection down the tree can quickly limit the ability of MCTS to explore states over longer time horizons. Thus, progressive widening (PW) [26] can be used to limit the number of children required to consider the parent fully expanded. This limit k_{PW} increases as a function of the number of times a node has been visited by

$$k_{PW} = \lfloor CN(s)^{\alpha} \rfloor \tag{2.37}$$

with constants $C > 0$ and $\alpha \in [0, 1]$.

(4) **Simulation** - At the newly expanded leaf node, a rollout is performed to generate an estimate of the value function at the new state. A rollout heuristic, or simulation, generates a set of actions to take from the leaf node until a maximum depth or terminating criteria in the MDP is reached. At the end of the rollout, a discounted cumulative reward is returned to the leaf node. The rollout heuristic or simulation can be fully random or use domain knowledge to return a reward.

(5) **Backpropagation** - After a discounted cumulative reward $\boldsymbol{r}_i$ is returned from simulation at the leaf node (s_i), the action value function and node visit counts are updated for every action taken and node visited during the traversal. Starting from the parent of the leaf node and running until the root node is reached at depth $i = 0$

$$\boldsymbol{Q}(s_j, a_j) = \boldsymbol{Q}(s_j, a_j) + \frac{\boldsymbol{r}_{j+1} - \boldsymbol{Q}(s_j, a_j)}{N(s_j, a_j)} \tag{2.38}$$

$$N(s_j, a_j) = N(s_j, a_j) + 1 \tag{2.39}$$

$$N(s_j) = N(s_j) + 1 \tag{2.40}$$

for $j = \{i-1, i-2, ..., 0\}$ where a_j is the action taken at state s_j to get state s_{j+1} at tree depth j.

In MDPs with vectorial rewards, MCTS is referred to as multi-objective MCTS (MO-MCTS). The implementation of MO-MCTS in this thesis is adapted from [98].

It is worth mentioning that MO-MCTS can be applied to MDPs where the sequence of actions $\{a_1, a_2, ..., a_j\}$ taken from an initial state s_1 does not affect the final state s_j resulting from those actions. Though extra logic must be included to ensure that a previously visited state is not added twice as a node, MO-MCTS combined with UCT used in MDPs with fully deterministic state transitions is a powerful tool for state space exploration. MO-MCTS has been implemented in numerous multi-objective optimization problems in the engineering domain posed as single player games [84] due to its versatility and ability to effectively balance exploration with exploitation in highly dimensional state spaces [7, 83, 58]. In this thesis, MO-MCTS is used to intelligently explore the state space of MOO problems and return a set of Pareto optimal solutions instead of an optimal policy. However, within the implementation of MO-MCTS there is an implication that the problem can be altered to enforce sequential actions, which can be used to simulate an optimal sequence of observer placement in the Cislunar SSA architecture design problem. This such use case is not addressed in this thesis, but is a potential direction for continuing work in the field.

2.2.1.4 Hybrid Approach

Within this thesis, a hybrid approach is implemented that combines the strengths of NSGA-II and MO-MCTS to more efficiently solve MOOPs. While MO-MCTS is typically used in sequential decision making processes, since we are not interested in the specific sequence of actions taken, the underlying MOOPs can be solved using a variety of methods across the fields of multi-objective optimizaiton and machine learning/reinforcement learning (ML/RL). In this thesis, MO-MCTS is used to explore the initial state space of the MOOP, at which point NSGA-II is used to refine the solutions found by MO-MCTS. This is effective in that MO-MCTS with UCT and progressive widening is able to quickly identify promisin areas of the state space, while NSGA-II with initial population filled with the solutions found by MO-MCTS is able to refine the solutions and converge on the Pareto front. This hybrid approach is used in Chapters 3 and 4 to solve the Cislunar SSA architecture design problem, and is shown to be effective in finding a diverse set of Pareto optimal solutions.

2.3 Graph Theory

Here, a brief summary of the basic concepts of graph theory is presented as it relates to this thesis. For an exhaustive discussion, the reader is referred to [12]. A graph G is a an ordered pair of sets $G = (\mathcal{N}, \mathcal{E})$ such that $\mathcal{E} \subset \mathcal{N}^{(2)}$, where $\mathcal{N}$ is the set of *nodes* or *vertices* and $\mathcal{E}$ is the set of *edges.* An edge $(\eta_i, \eta_j) \in \mathcal{E}(G)$ connects nodes $\eta_i, \eta_j \in \mathcal{N}(G)$, and η_i and η_j are said to be adjacent or neighbors. For brevity, this edge can be written as $\eta_i\eta_j$ or ij. Note that where applicable in this thesis, edges are *implied* through the existence of neighbors.

The set of nodes that are adjacent to a node $\eta_i \in G$ is called the *neighbourhood* of η_i, denoted by $\Gamma(\eta_i)$. The *degree* of a node $d(\eta_i) = |\Gamma(\eta)|$ is the number of neighbors of η_i. If the edges in G are *ordered* pairs of nodes, then G is said to be a *directed graph* where ij and ji are unique edges. If only one such edge ij or ji is allowed, then the graph G is said to be an *oriented graph.* In these graphs, a node thus has an *indegree*, $d^{-}(\eta)$ and *outdegree* $d^{+}(\eta)$, representing the amount of edges going into and out of the node, respectively.

A *path* is a graph $P = (\mathcal{N}_P, \mathcal{E}_P)$ such that $P \subset G$ and

$$\mathcal{N}_P = \{\eta_0, \eta_1, ..., \eta_l\} \qquad \mathcal{E}_P = \{\eta_0\eta_1, \eta_1\eta_2, ..., \eta_{l-1}\eta_l\} \tag{2.41}$$

where $l = e_P$ is the length of P and each η_i is distinct. Two paths are considered independent if the only shared nodes between them are the start and end node. Here, since edges are implied through node neighbors, a path is simply represented by a sequence of nodes. In this thesis, the exploration of *acyclic oriented graphs* will be conducted, oriented graphs in which all directed edges are oriented in the same direction, in this case forward in time.

2.3.1 A* Search

The A* algorithm is a well-established pathfinding method that combines the principles of Dijkstra's algorithm and greedy best-first search. It is designed to find the shortest path between two nodes in a graph by minimizing the total estimated cost from the start node to the goal node. This is achieved by utilizing a heuristic function $h(\eta)$, which estimates the cost to reach the goal from a given node η, in conjunction with the actual cost $g(\eta)$ incurred to reach that node from the start [53].

A* operates by exploring paths that minimize the function $f(\eta) = g(\eta) + h(\eta)$, where $g(\eta)$ is the cumulative cost of traversal from the start node to η, and $h(\eta)$ (the heuristic) provides an estimate of the cost from η to the goal. The algorithm prioritizes nodes in the search based on this combined cost, allowing it to effectively balance the exploration of the search space with the goal of finding the optimal path efficiently [53].

In practice, the A* algorithm maintains a priority queue of nodes to be explored, starting from the initial node. At each step, the node with the lowest $f(\eta)$ value is expanded, and its neighbors are added to the queue with updated $g(\eta)$ and $f(\eta)$ values. The search continues until the goal node is reached, ensuring that the path found is the one with the lowest total cost, assuming an admissible heuristic is used. For this work, we use a weighted heuristic based on the current node's distance and time to a predetermined goal. The cost of traversal between nodes, represented by $g(\eta)$, can then be determined through a custom distance function, depending on the specific context.

2.4 Reachable Set Theory

Reachability theory involves solving for all possible states that a spacecraft can reach given a set of feasible control, over some time horizon t_f. This is typically formulated as a set of optimal control problems, where the objective is to maximize the reachable set while adhering to the spacecraft's dynamics and control constraints. The discussion here is largely taken from [57, 18], which both provide methods for approximating reachable sets.

The optimal control problem can be stated as

$$\begin{aligned} \max_{\boldsymbol{u}\in\mathcal{U}} & \left[\int_{t_0}^{t_f} \mathcal{L}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau) d\tau + V(\boldsymbol{x}_f, t_f)\right] \\ \text{s.t.} \qquad & \dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}, t) \\ & g(\boldsymbol{x}_0, t_0, \boldsymbol{x}_f, t_f) \leq 0 \end{aligned} \tag{2.42}$$

where $\boldsymbol{x}$ and $\boldsymbol{u}$ are the state and control, respectively, $t \in [t_0, t_f]$ is time, $\mathcal{L}$ is the system Lagrangian, $V(\cdot)$ is the terminal performal function, $\boldsymbol{f}(\cdot)$ is the system dynamics (here the planar CR3BP), $\boldsymbol{g}(\cdot)$ is a boundary condition, and $\mathcal{U}$ is the set of all feasible controls. In a reachability analysis, a minimum-time formulation is used such that $\mathcal{L} = 0$, which leads to maximizing the final value function $V(\cdot)$. This results in the Hamilton-Jacobi-Bellman (HJB) partial differential equation, which can be solved to obtain the minimum-time optimal control, maximizing the rechable set.

$$\frac{\partial V}{\partial t} + \max_{\boldsymbol{u}\in\mathcal{U}} \left[\frac{\partial V}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t)\right] = 0 \tag{2.43}$$

Here, the second term is the Hamiltonian, which represents the maximum rate of change of the value function with respect to the state.

$$\mathcal{H}^* \;=\; \max_{\boldsymbol{u}\in\mathcal{U}}\Big[\frac{\partial V}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u},t)\Big] \tag{2.44}$$

Often tmes, the HJB is solved using level-set methods for the value function V, the zero-level set solution of which marks the boundary of the exact reachable set. Such methods suffer from the curse of dimensionality. Most often, the reachable set is approximated by solving multiple optimal control problems, sufficiently sampling the state and control space to obtain an overapproximation of the reachable set at time t_f [18, 57]. The forward reachable set over t_f can then be defined as

$$\mathcal{R}(t_f;\mathcal{U},\boldsymbol{f},\boldsymbol{g},t_0) = \{\boldsymbol{x}_f \in \mathbb{R}^n : \forall \boldsymbol{u} \in \mathcal{U}, \boldsymbol{g}(\boldsymbol{x}_0) \leq 0, \boldsymbol{x}_f = \phi_{\boldsymbol{f}}(t_f;\boldsymbol{x}_0,\boldsymbol{\lambda}_0,t_0)\} \tag{2.45}$$

where $\phi_{\boldsymbol{f}}$ is the dynamics flow function, $\boldsymbol{x}_0$ is the initial state, and $\boldsymbol{\lambda}_0$ is the initial costate. For notation's sake, the reachable set and intermediate states eminating from $\boldsymbol{x}_0$ over time horizon $T = [t_0, ..., t_f]$ is denoted as $\mathcal{R}(\boldsymbol{x}_0, T)$.

Chapter 3

Optimal Cislunar Architecture Design Using Monte Carlo Tree Search

This chapter presents a novel multi-objective Monte Carlos Tree Search (MO-MCTS) algorithm to uncover near-optimal solutions to the Cislunar architecture design problem, addressing a gap in the SSA literature at the point of publication. First, the simulation framework for evaluating architecture is described, including discussion on the solar illumination. Then, the multi-objective architecture design problem is described and formulated as a Markov Decision Process (MDP). The MDP is then solved using the MO-MCTS algorithm, which is designed to efficiently explore the architecture design space and identify Pareto optimal solutions. Finally, the results of the MO-MCTS algorithm are presented within both a trajectory- and volume-oriented SSA framework, demonstrating its effectiveness in finding high-quality architectures for Cislunar SSA.

3.1 Multi-Objective Architecture Design

3.1.1 Simulation Framework

To effectively evaluate the performance of Cislunar architectures, a simulation framework is developed, incorporating solar illumination data from the Python wrapper for the Spice toolkit [4] within the CR3BP as a function of time. The framework allows construction of architectures at specified simulation epochs, at which initial phasings of observers are assigned. Furthermore, because of the influence of solar illumination on the visibility of SOs in the region, we evaluate architecture over multiple simulation epochs, each with a different initial solar illumination, in order to effectively capture the time-dependent nature of Cislunar SSA. Specifically, as discussed in

Section 2.1.4, the maximum possible architecture period (time for all relative geometries to repeat) is about 60 synodic periods. As such, each architecture is evaluated over 60 synodic periods, or about five years. Within this total time period, there are 30 simulation epochs, with the solar illumination at each epoch being shifted by 24 hours. This allows for a more comprehensive evaluation of each architecture's unique and complex geometry. This epoch variation is described in Figure 3.1.

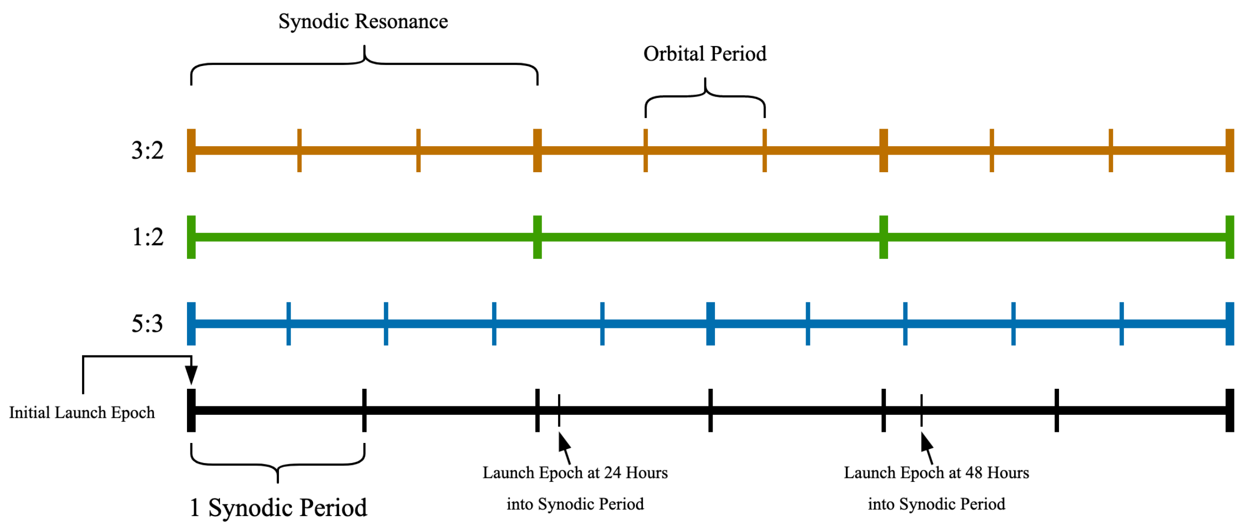


Figure 3.1: Resonance of a Cislunar architecture configuration consisting of three observers with 3:2, 1:2, and 5:3 resonance with the Earth-Moon synodic period. Every other synodic period the architecture configuration is evaluated at a different epoch, each separated by 24 hours relative to the synodic period. The pattern repeats for 60 synodic periods.

3.1.2 Average Coverage Metric

To quantify performance, we use a simple metric that captures the ability of an architecture to maintain or obtain coverage of a trajectory within Cislunar space. For an SO in Cislunar space at $\boldsymbol{r}_{SO}(t_i)$, we use the equations from Sections 2.1.2 and 2.1.3 to determine wether it is detectable by any observer (within its FoR) in a distributed architecture $\boldsymbol{X} = \{Obs_1, Obs_2, ..., Obs_n\}$. For a SO $\boldsymbol{r}_{SO}$ at simulation time t_i, a general operator $\boldsymbol{\Lambda}(t_i, \boldsymbol{r}_{SO}, \boldsymbol{X})$ calculating SNR for all observers can be expressed for convenience as

$$\mathbf{\Lambda}(t_i, \boldsymbol{r}_{SO}, \boldsymbol{X}) = \begin{bmatrix} \Lambda(t_i, \boldsymbol{r}_{SO}, Obs_1) \\ \Lambda(t_i, \boldsymbol{r}_{SO}, Obs_2) \\ \vdots \\ \Lambda(t_i, \boldsymbol{r}_{SO}, Obs_{n_{obs}}) \end{bmatrix} \in \Re^{n_{obs}} \tag{3.1}$$

$$\Lambda(t_i, \boldsymbol{r}_{SO}, Obs_j) = \begin{cases} \text{SNR}(t_i, \boldsymbol{r}_{SO}, Obs_j) & \text{if visible by } Obs_j \\ \emptyset & \text{otherwise} \end{cases} \tag{3.2}$$

where $\text{SNR}(\boldsymbol{r}_{SO}, \boldsymbol{r}_{Obs_j}(t_i))$ is the SNR of the SO at time t_i as seen by observer Obs_j, and $\emptyset$ indicates that the SO is not visible by observer Obs_j at time t_i due to either illumination or pointing constraints as detailed in Section 2.1.2. Note that the SNR is calculated using the equations from Section 2.1.3, and for simplicty it is implied that all relevant parameters (e.g., telescope diameter, exposure time, etc.) are included in the SNR calculation, and that the simulation environment is updated to account for the changing solar illumination and observer positions over time. Then, a detection indicator $D(t_i, \boldsymbol{r}_{SO}, \boldsymbol{X})$ is defined as

$$D(t_i, \boldsymbol{r}_{SO}, \boldsymbol{X}) = \begin{cases} 1 & \text{if any } \mathbf{\Lambda}(t_i, \boldsymbol{r}_{SO}, \boldsymbol{X}) > \text{SNR}_{min} \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

where SNR_{min} is the minimum SNR required for detection. Finally, a detection mask $\boldsymbol{D}(\boldsymbol{T}_j, \boldsymbol{r}_{SO}, \boldsymbol{X})$ can be compiled for an architecture over a simulation epoch $\boldsymbol{T}_j = [t_0, t_1, ..., t_{n_{steps}}]$ for n_{steps} time steps per simulation epoch as

$$\boldsymbol{D}(\boldsymbol{T}_j, \boldsymbol{r}_{SO}, \boldsymbol{X}) = [D_{det}(t_i, \boldsymbol{r}_{SO}, \boldsymbol{X}) \mid t_i \in \boldsymbol{T}_j] \tag{3.4}$$

To reiterate, it is implied that all relevant simulation parameters including the solar illumination, observer positions, and SNR calculation parameters are updated at each time step t_i in the simulation epoch $\boldsymbol{T}_j$. This also includes if the SO is traversing along a precomputed trajectory over time. The detection mask $\boldsymbol{D}(\boldsymbol{T}_j, \boldsymbol{r}_{SO}, \boldsymbol{X})$ is then used to calculate the average coverage metric for an architecture $\boldsymbol{X}$ over a trajectory $\boldsymbol{r}_{SO}$ as follows. So, for the j^{th} simulation epoch over time $\boldsymbol{T}_j = [t_0, t_1, ..., t_{n_{steps}}]$, the average coverage metric $\eta_j(\boldsymbol{X})$ for a SO is defined as

$$\eta_j(\boldsymbol{X}) = \frac{\boldsymbol{D}(\boldsymbol{T}_j, \boldsymbol{r}_{SO}, \boldsymbol{X}) \cdot \boldsymbol{I}}{n_{steps}} \tag{3.5}$$

The total average coverage of architecture $\boldsymbol{X}$ over all simulation epochs is then

$$\eta(\boldsymbol{X}) = \text{avg}(\{\eta_1(\boldsymbol{X}), \eta_2(\boldsymbol{X}), ..., \eta_{n_e}(\boldsymbol{X})\}) \tag{3.6}$$

for n_e simulation epochs.

3.1.3 Architecture Cost Metric

The cost metric assigns a cost to an architecture based on the number of observers and the diameter of the telescope it uses. It is designed to represent the nonlinear price increase associated with going from smaller to larger diameter telescopes. Using the equation $(\frac{C}{200mm})^2$ for $C = 200mm, 300mm, 500mm$, the cost of an architecture can be defined as a function of the number of observers with varying telescope diameters:

$$C_{total} = C_{200mm}n_{200mm} + C_{300mm}n_{300mm} + C_{500mm}n_{500mm} \tag{3.7}$$

$$C_{200mm} = 1$$

$$C_{300mm} = 2.25$$

$$C_{500mm} = 6.25$$

where n_{200mm}, n_{300mm}, and n_{500mm} are the number of observers with telescope apertures of 200mm, 300mm, and 500mm, respectively. For use in the MO-MCTS algorithm such that it scales well with the coverage objective defined in $[0, 1]$, the cost is normalized to be in $[0, 1]$ and to be maximized

$$\hat{C} = 1 - \frac{C_{total}}{C_{max}} \tag{3.8}$$

where C_{max} is the maximum possible cost of an architecture. In this case, with a maximum of five observers and a maximum telescope diameter of 500mm, $C_{max} = 6.25 \cdot 5 = 31.25$. While many other studies use the linear stablity index [89, 97, 29, 24] which is representative of the cost of stationkeeping, the cost metric used here seeks to capture the costs associated with the observers' actual hardware which is coupled with its capabilities. This cost also encapsulates the potential cost of launching an observer into orbit, which also increases nonlinearly with the size of the telescope.

3.1.4 MOO-MDP Formulation

The architecture design problem can now be expressed as a multi-objective optimization problem, where the objectives are to maximize the average coverage η and minimize the cost $\hat{C}$ of an architecture. The problem can be expressed as follows:

$$\max_{\boldsymbol{X}} \left(\hat{C}, \eta\right) \tag{3.9}$$

where solutions to this problem lie along the Pareto-front of the two objectives. Here, the Pareto-front represents the trade-off between the cost of an architecture and its performance in terms of average coverage. The state space consists of all possible architecture layouts consisting of observers in various orbits with varying phasings and telescope diameters. There are a total of 156 orbits for observers to occupy and with a discretization of the phasing into 10 evenly spaced points, if the total number of observers in an architecture is constrained to five then the state space of this problem is $\sim 10^{18}$.

To fit within the MO-MCTS framework, the MOOP can be reformulated as a MDP with deterministic transitions, and discount factor $\gamma = 1$. Actions in this contribution are defined as the addition of a new observer to an architecture layout. Since there are varying amounts of orbits per family, each orbit family is equally weighted when sampling actions, after which a random orbit is selected from the family. The sampling of initial phasing and telescope diameter is similarly uniformly sampled from the discretized set of phasings and diameters. Within MO-MCTS, the maximum search depth is set to five, as an architecture with five observers is considered a terminal state.

3.2 Trajectory-Oriented Architecture Design

First, we optimize Cislunar architectures for coverage of a basic Earth-Moon transfer trajectory. The trajectory departs from a GEO orbit, takes around three days to reach the Moon, and is similar to a typical trans-lunar injection (TLI) trajectory. The trajectory is shown in Figure 3.2.

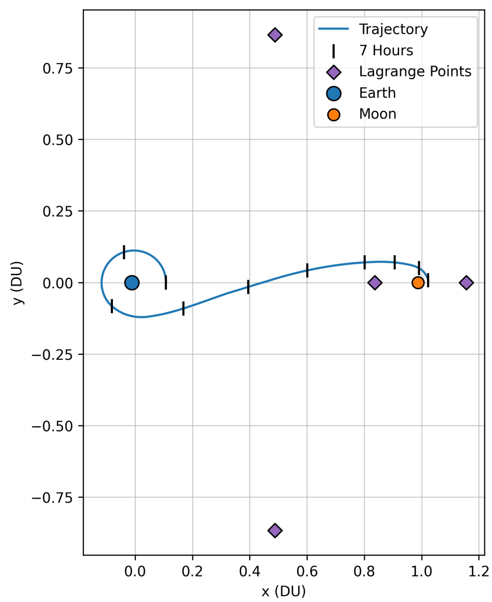
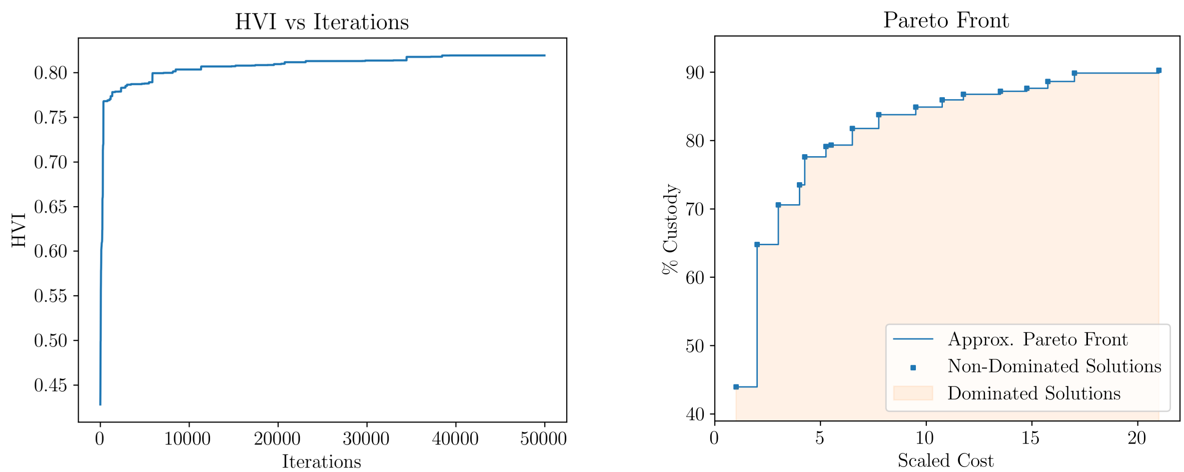


Figure 3.2: Basic Earth-Moon transfer trajectory used for the trajectory-oriented architecture design problem. The trajectory is similar to a typical trans-lunar injection (TLI) trajectory, departing from a GEO orbit and taking around three days to reach the Moon.

As described in Figure 3.1, a wide range of relative geometries are evaluated for each architecture iteration during optimization. Each architecture is evaluated over 30 simulation epochs within the total 60 synodic periods of the simulation. At each iteration, the average coverage metric is calculated for the trajectory and combined with the cost metric, the architecture is compared to the current Pareto front.

We run MO-MCTS with the trajectory-oriented architecture design formulation for 50,000 iterations, updating the Pareto set every 10 iterations. The algorithm takes approximately 14 hours to complete running on a 2021 M1 Macbook Pro. Figure 3.3a shows the change in hyper-volume indicator over iterations, showing that after about 10,000 iterations, solutions found are only slightly improving with respect to the objective functions, a case of diminishing returns. Figure 3.3b shows a fairly crowded approximate Pareto front, with many solutions past scaled cost of 5 showing only slight improvement in coverage maintained over the simulation epochs.



(a) MO-MCTS HVI vs iterations for the trajectory-oriented architecture problem.

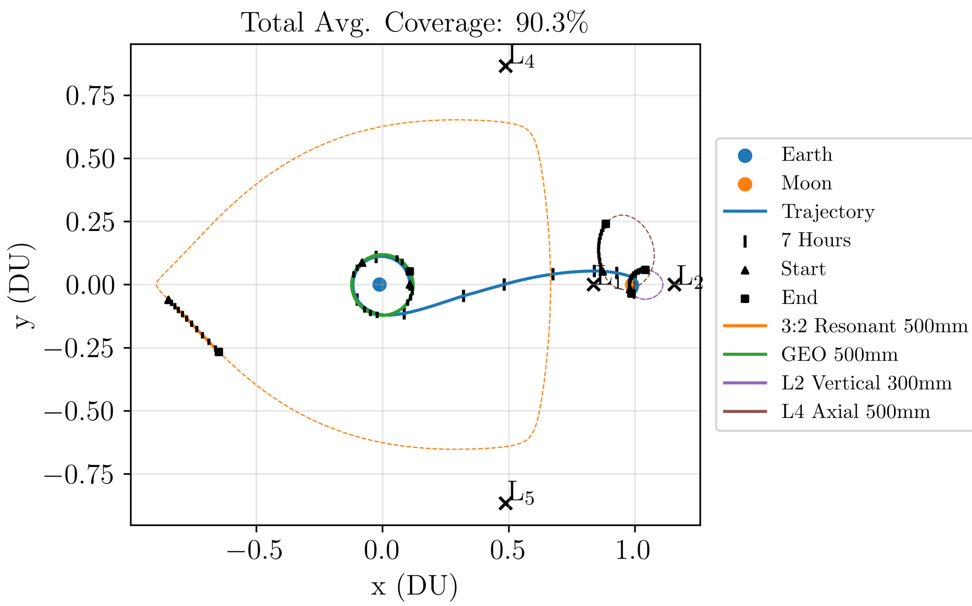
(b) Resultant approximate re-scaled Pareto front for the trajectory-oriented architecture problem.

Figure 3.3: MO-MCTS results for the trajectory-oriented Cislunar architecture problem.

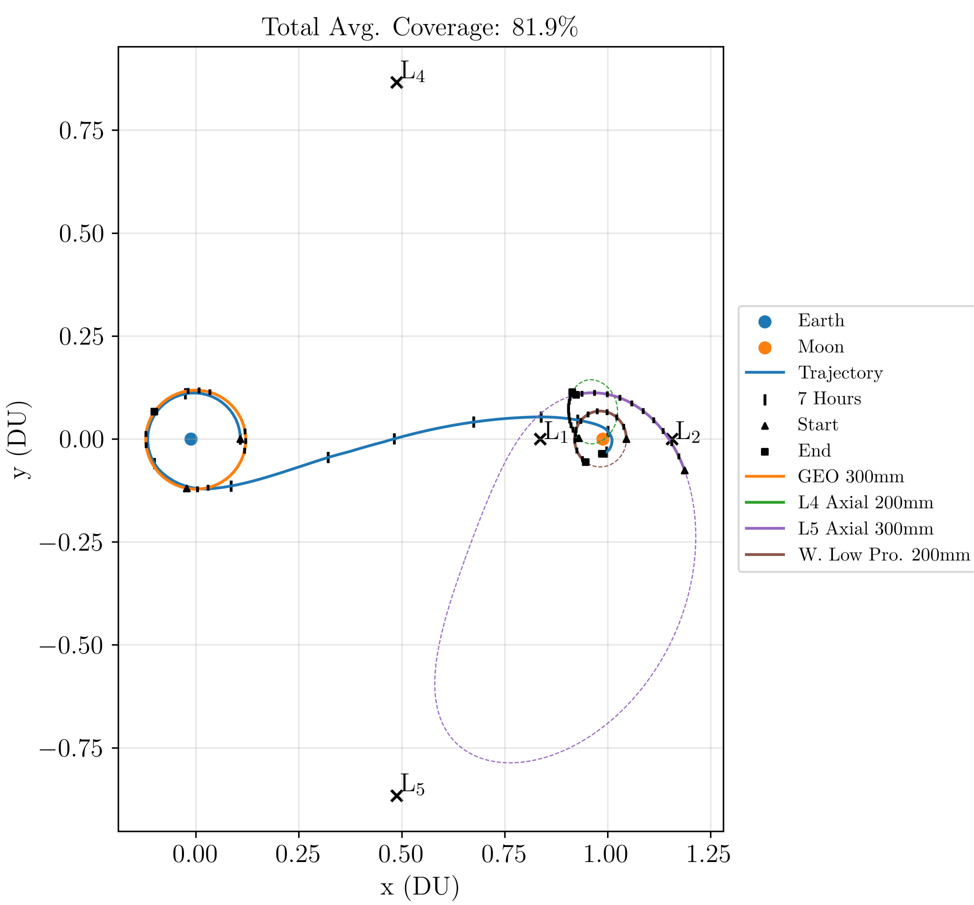
3.2.1 Results and Discussion

A selection of architectures from the Pareto front are discussed here, representing different trade-offs between cost and coverage.

Figure 3.4 shows a sample of the Pareto-optimal solutions found from the MO-MCTS algorithm. These solutions are further detailed in Table 3.1, including the epochs at which minimum and maximum coverage of the SO in the reference trajectory is maintained. Of initial interest is the overwhelming tendency of the algorithm to pick multiple GEO observers in all solutions providing coverage for the initial portion of the reference trajectory. This result is as expected, since observers in Lagrange point periodic orbits will have difficulty observing a space object as it starts its trajectory in a near-Earth orbit. Figures 3.5 and 3.6, which display the SNR history of

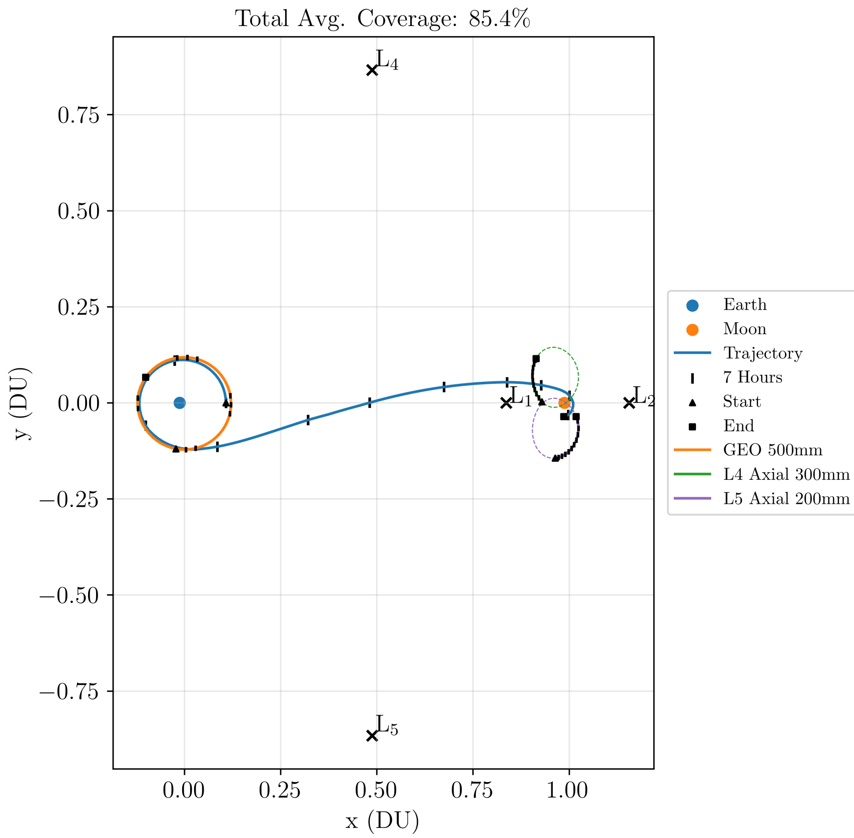


(a)

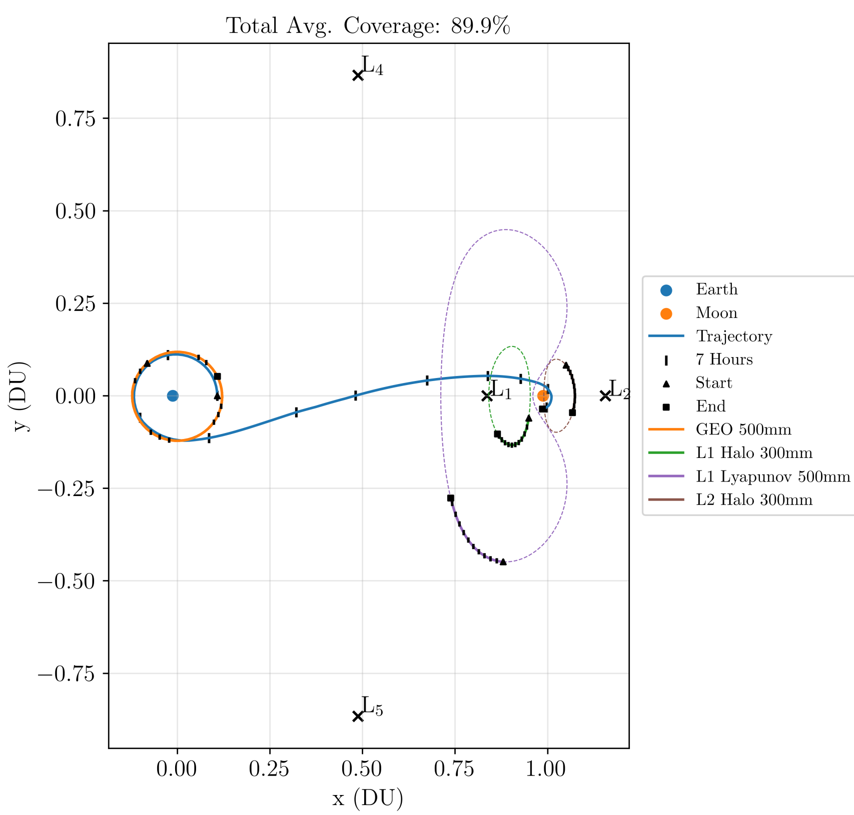


(b)

Figure 3.4: Four resultant Pareto-optimal Cislunar architecture layouts with trajectory starting at the initial simulation epoch (continued on next page).



(c)



(d)

Figure 3.4: Four resultant Pareto-optimal Cislunar architecture layouts with trajectory starting at the initial simulation epoch.

Table 3.1: Configuration statistics from Pareto-optimal outputs in Figure 3.4.

(a) Configuration statistics from Figure 3.4a.

Total Coverage (%)	**Total Cost**	**Min. Coverage (%)**	**Max. Coverage (%)**
90.297	21.0	77.707	96.815
	Epoch	2026 JUN 09 11:00:00	2024 AUG 18 06:30:00

(b) Configuration statistics from Figure 3.4b.

Total Coverage (%)	**Total Cost**	**Min. Coverage (%)**	**Max. Coverage (%)**
81.783	6.5	61.146	94.904
	Epoch	2026 AUG 08 11:30:00	2024 OCT 17 07:00:00

(c) Configuration statistics from Figure 3.4c.

Total Coverage (%)	**Total Cost**	**Min. Coverage (%)**	**Max. Coverage (%)**
84.904	9.5	62.420	98.089
	Epoch	2026 AUG 08 11:30:00	2024 JUN 19 06:00:00

(d) Configuration statistics from Figure 3.4d.

Total Coverage (%)	**Total Cost**	**Min. Coverage (%)**	**Max. Coverage (%)**
89.873	17.0	74.522	98.089
	Epoch	2022 MAY 01 00:30:00	2024 AUG 18 06:30:00

the reference trajectory as seen by each observer in the architecture, further display this behavior and demonstrate the ability of the MO-MCTS algorithm to piece together observers in different orbits to maintain maximal coverage of the space object in the reference trajectory. These plots also reinforce the neccessity to evaluate performance over long periods of time as the the relative geometries of the observers and solar illumination change significantly over time. As such, observers play various roles over time in providing coverage of the SO in its trajectory.

Figure 3.7 presents a histogram of the types of orbits found optimal to be used in the Pareto-optimal configurations. Again, here we see the algorithm's tendency in using GEO observers, as well as its reliance on L1 Halo and Lyapunov orbits and orbits such as the Southern Butterfly which tend to hover around the general location of the Moon. An expected result is that larger orbits, such as Resonant orbits, are selected only a couple of times due to their distance from the reference trajectory. When a Resonant orbit is selected for use in the architecture, such as the 3:2 Resonant orbit in Figure 3.4a, it is chosen with a 500mm telescope in order to view the SO at a reasonable SNR, thus increasing the cost of the observer. In general, many of the Pareto-optimal solutions returned from MO-MCTS place observers in the same orbital families or in similar orbital families while making slight variations to the initial phasing and telescope aperture.

3.3 Volume-Oriented Architecture Design

Next, we optimize Cislunar SSA architectures for coverage of two volumes of interest, one including the GEO region and the Earth-Moon corridor, referred to as the "GEO Gateway" volume, and the other a larger volume, including the L3 Lagrange point, referred to as the "Large" volume. The volumes are shown in Figure 3.8 and extend along the z axis in the CR3BP rotating frame. In this more "generic" formulation, we might expect to find architecture layouts with observers more equally spaced in decision variable space such that a larger volume of Cislunar space may be accessible by the architecture at any given epoch.

In this formulation, a volume point in Cislunar space bounded by a predefined ellipsoid is considered accessible if it is illuminated and not occluded with respect to an observer as defined in Section 2.1.2. The set of volume points consists of n uniformly sampled 3D Cartesian points across a predefined ellipsoid. Starting from each epoch during the simulation, rather than performing a check for if the SO is accessible at a given time step along its trajectory, we check if an SO at each volume point in the specified region is detectable by the architecture. We perform these checks for every volume point at each epoch over multiple time steps in order to obtain an accurate representation of the viewing geometry of the architecture at each epoch. As before, there are 30 epochs over the 60 synodic period time horizon at which each architecture is evaluated, and at each epoch the coverage is calculated over a 30 day period at a 12 hour time step. Given the amount of calculations performed at epoch time step throughout the simulation, this volume-oriented approach is significantly more computationally intensive than the trajectory-oriented approach. Because of this, we perform 1,000 iterations of MO-MCTS for this section and use multiprocessing methods from `Python`'s `multiprocessing` library to spread out the cost of computing the volume point accessibility checks across multiple CPU cores.

3.3.1 Results and Discussion

For both the GEO Gateway and Large volumes, we run MO-MCTS for 1,000 iterations, updating the Pareto front every 10 iterations. At each epoch, observer positions are propagated for 24 hours along their respective orbits. A longer observation period was considered at each epoch, but again, due to the computational intensity of these calculations, we use the truncated observation period of 24 hours as a further discretization of the underlying MI-NLP problem. The resultant HVI histories and Pareto fronts for the GEO Gateway and Large volumes are shown in Figures 3.9a-3.9b and Figures 3.10a-3.10b, respectively. Intuitively, the GEO Gateway volume is covered by architectures at a lower cost than the Large volume. MO-MCTS finds architectures that reach approximately 90% coverage at a cost of about 10 for the GEO Gateway volume, while at the same cost it only finds architectures providing around 70% coverage to the Large volume. This is an intuitive result, and underscores the difficulty in conducting effective SSA in increasingly larger volumes of Cislunar space.

Figures 3.11 and 3.12 show a selected sample of layouts from the resultant Pareto-optimal states from MO-MCTS with the GEO Gateway and Large volumes, respectively. Notably, the GEO Gateway volume relies heavily on a single GEO observer to cover the 3.4xGEO volume, and one or many observers in orbits near the Moon to cover the Cislunar corridor. On the other hand, the Large volume produces architectures with observers "spread out" around the volume of interest occupying orbits that traverse much of the volume. Furthermore, there is a tendency in the selected layouts to use the 500mm diameter telescope to extend the reach of the observers in these larger orbits. This is in slight opposition to the size of telescopes used in the GEO Gateway volume results, another intuitive and expected result.

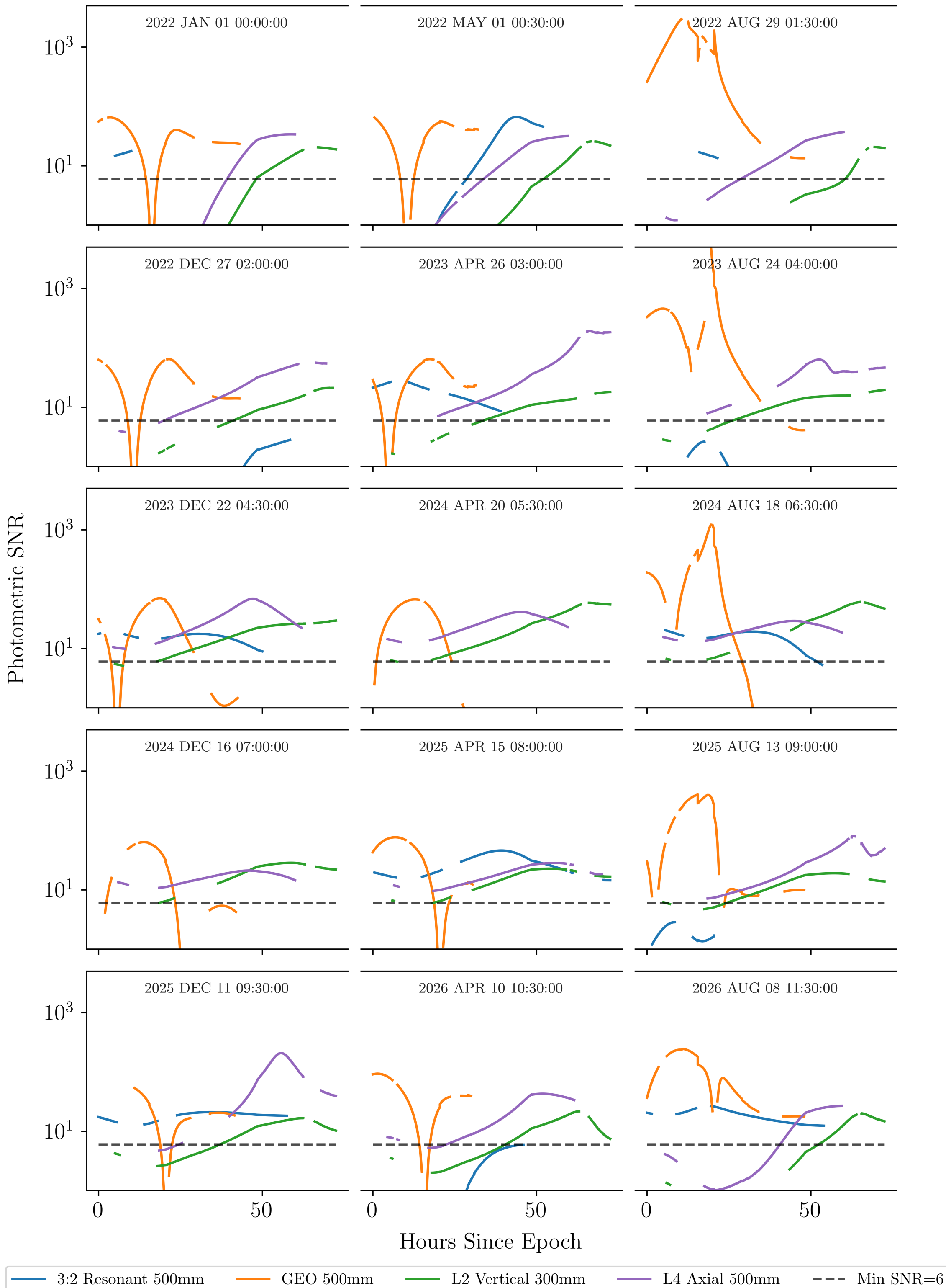


Figure 3.5: SNR history at every other simulation epoch for each observer in Figure 3.4a.

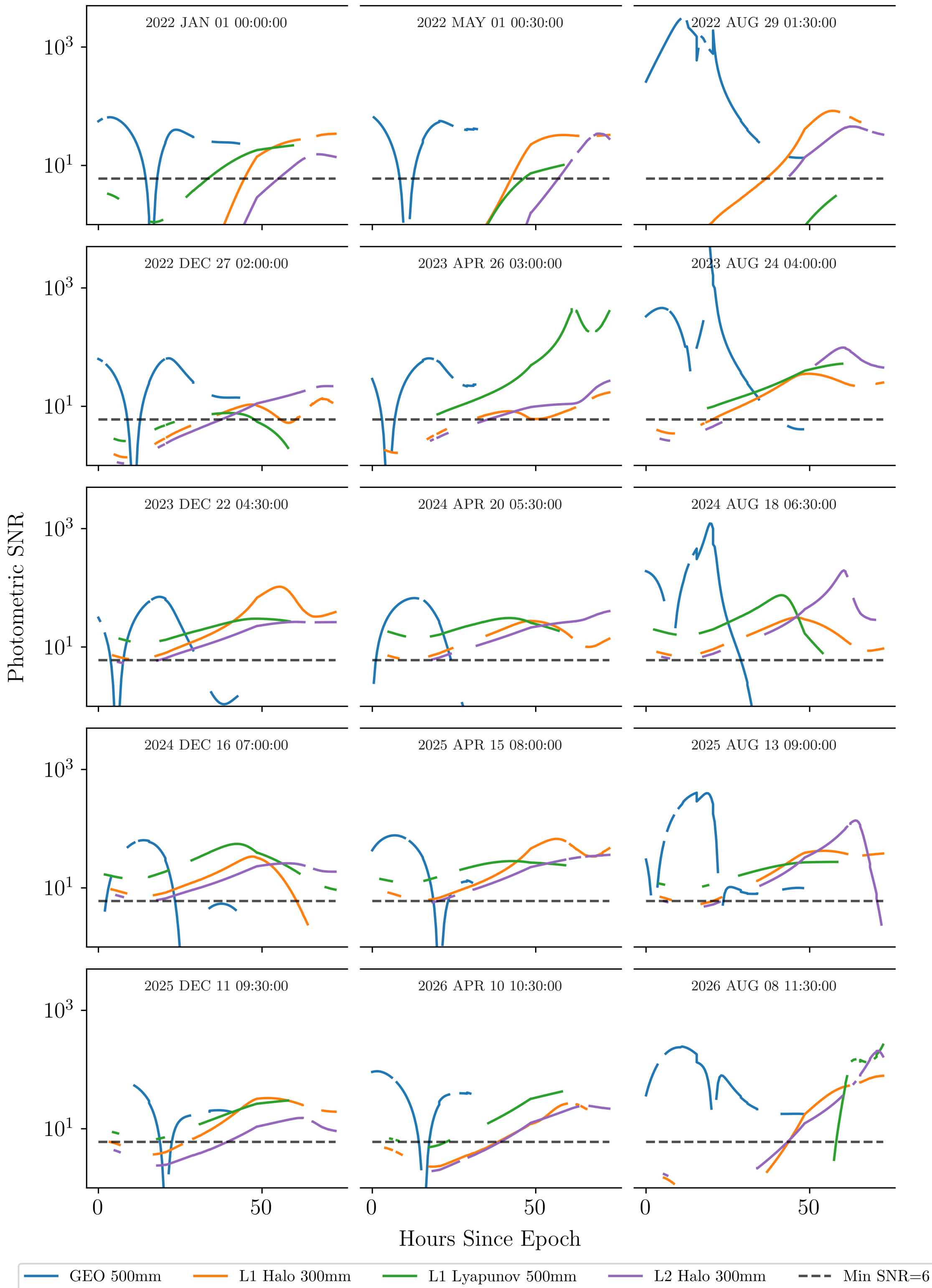


Figure 3.6: SNR history at every other simulation epoch for each observer in Figure 3.4d.

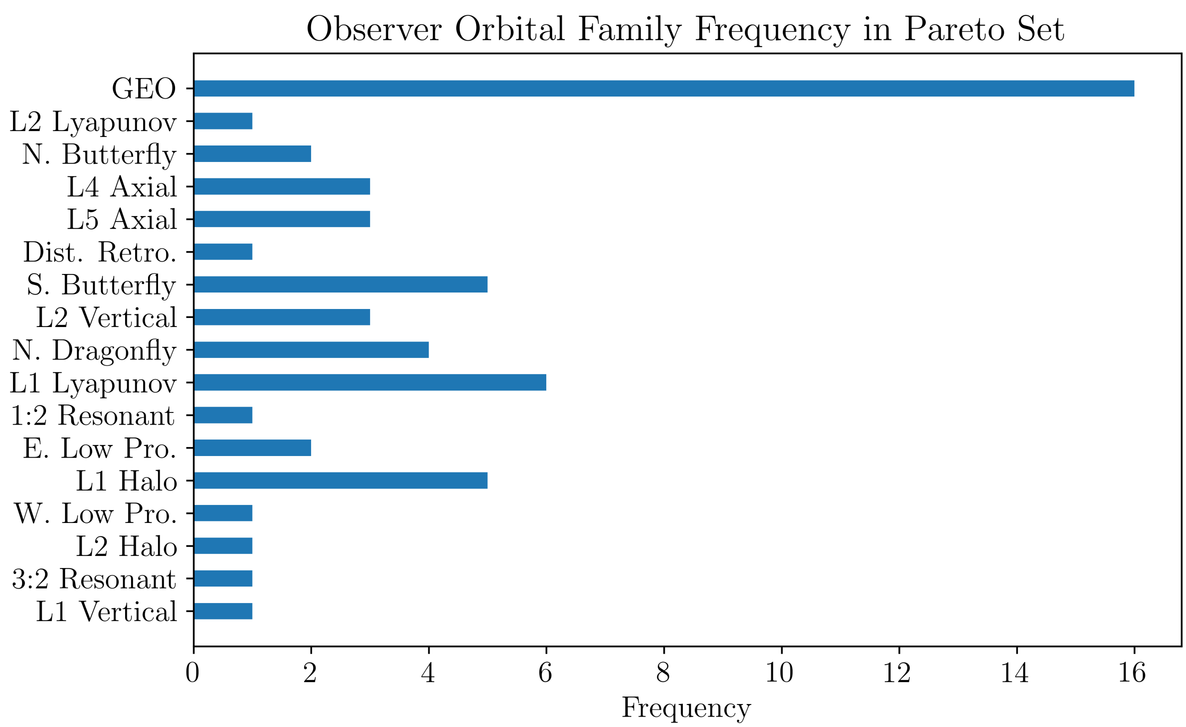
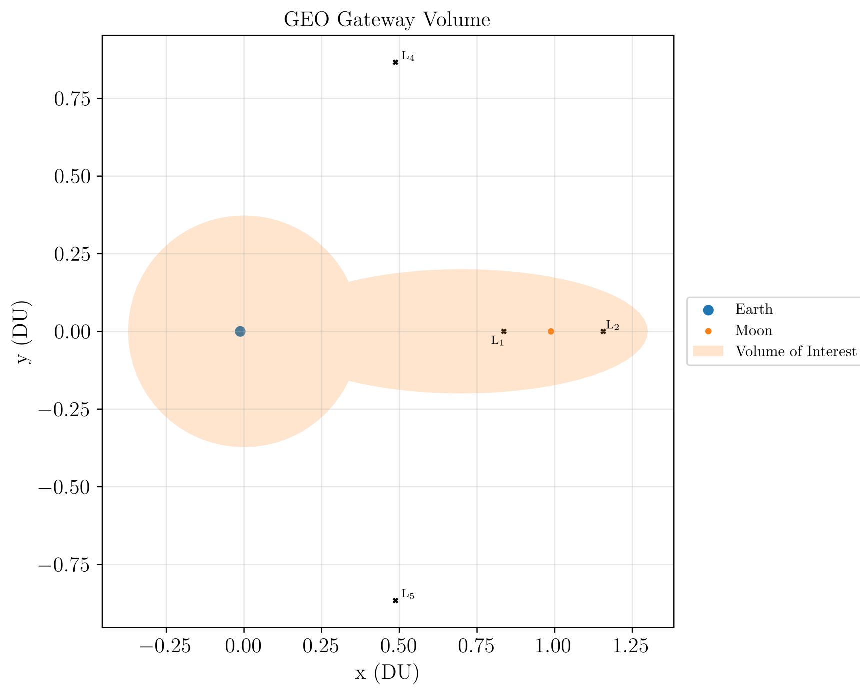
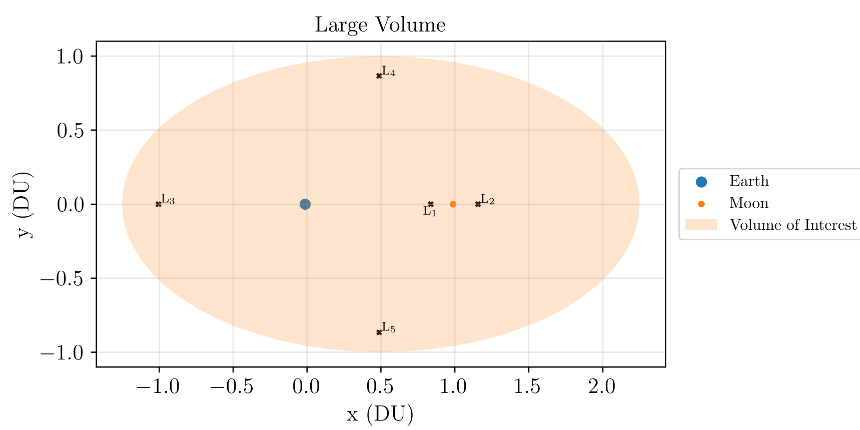


Figure 3.7: Frequency of orbit families used in solutions in the Pareto set returned by MO-MCTS.

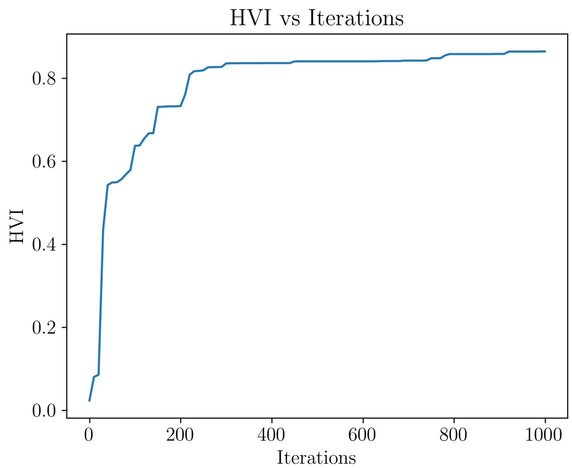


(a) "GEO Gateway" volume of interest for volume-oriented Cislunar architecture design, including a sphere centered on the Earth of radius 3.4xGEO.

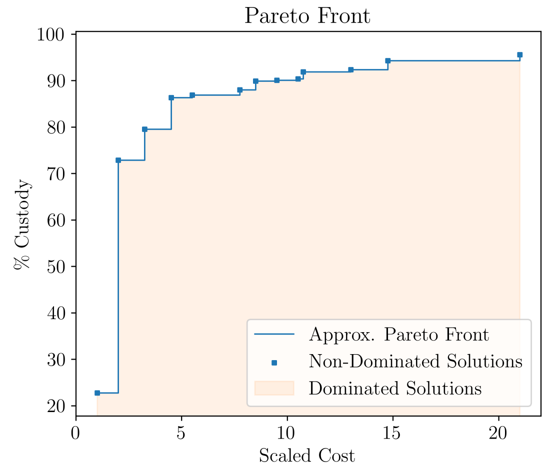


(b) "Large" volume of interest for volume-oriented Cislunar architecture design, including regions near L3 and well past L2.

Figure 3.8: Planar visualization of volumes used for volume-oriented Cislunar architecture design.

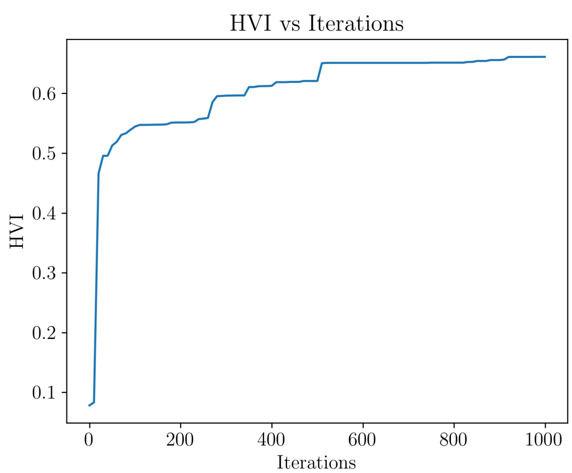


(a) MO-MCTS HVI vs iterations for the volume-oriented Cislunar architecture problem using the GEO Gateway volume.

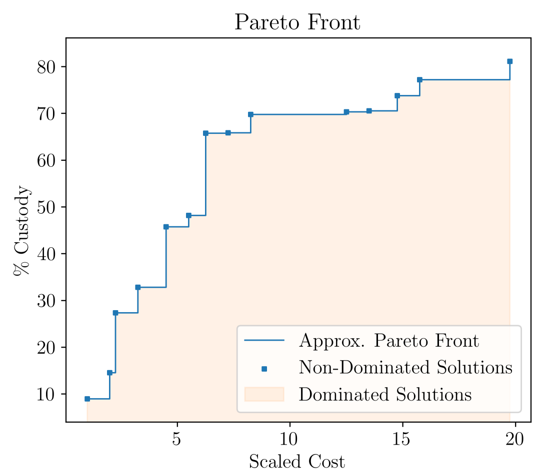


(b) Resultant approximate re-scaled Pareto front for the volume-oriented Cislunar architecture problem using the GEO Gateway volume.

Figure 3.9: MO-MCTS results for the architecture-oriented Cislunar architecture problem using the GEO Gateway volume.

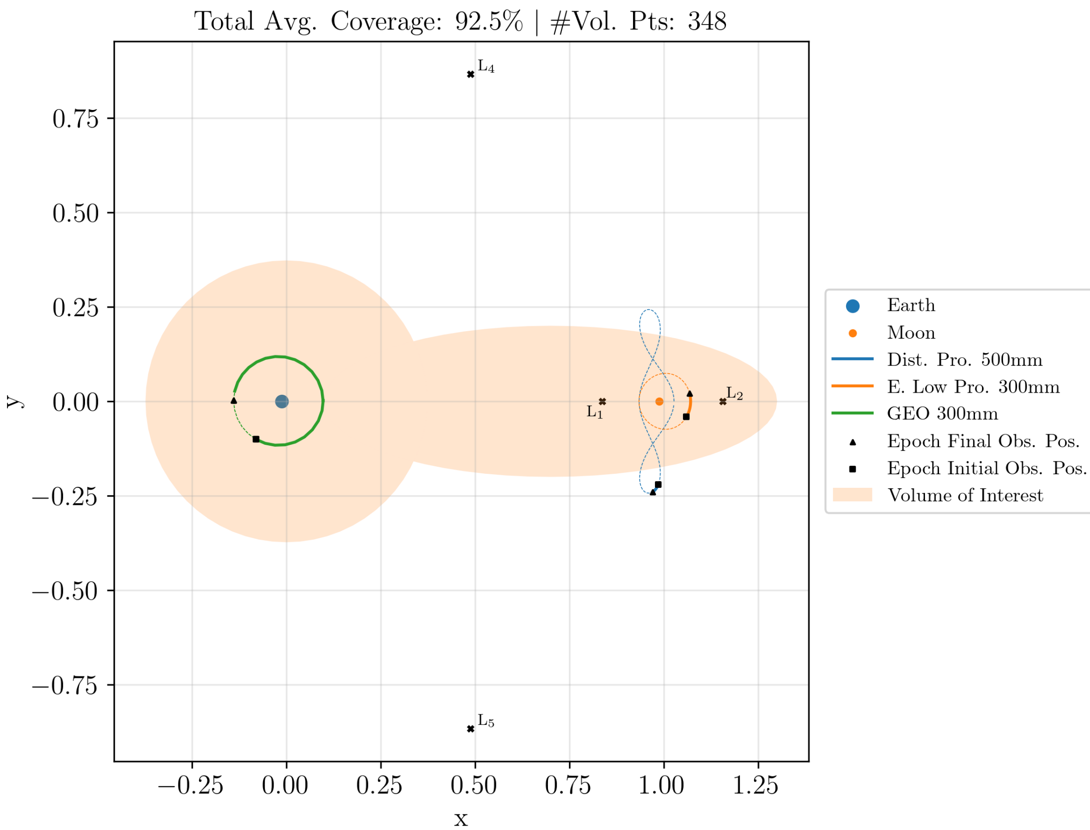


(a) MO-MCTS HVI vs iterations for the volume-oriented Cislunar architecture problem using the Large volume.

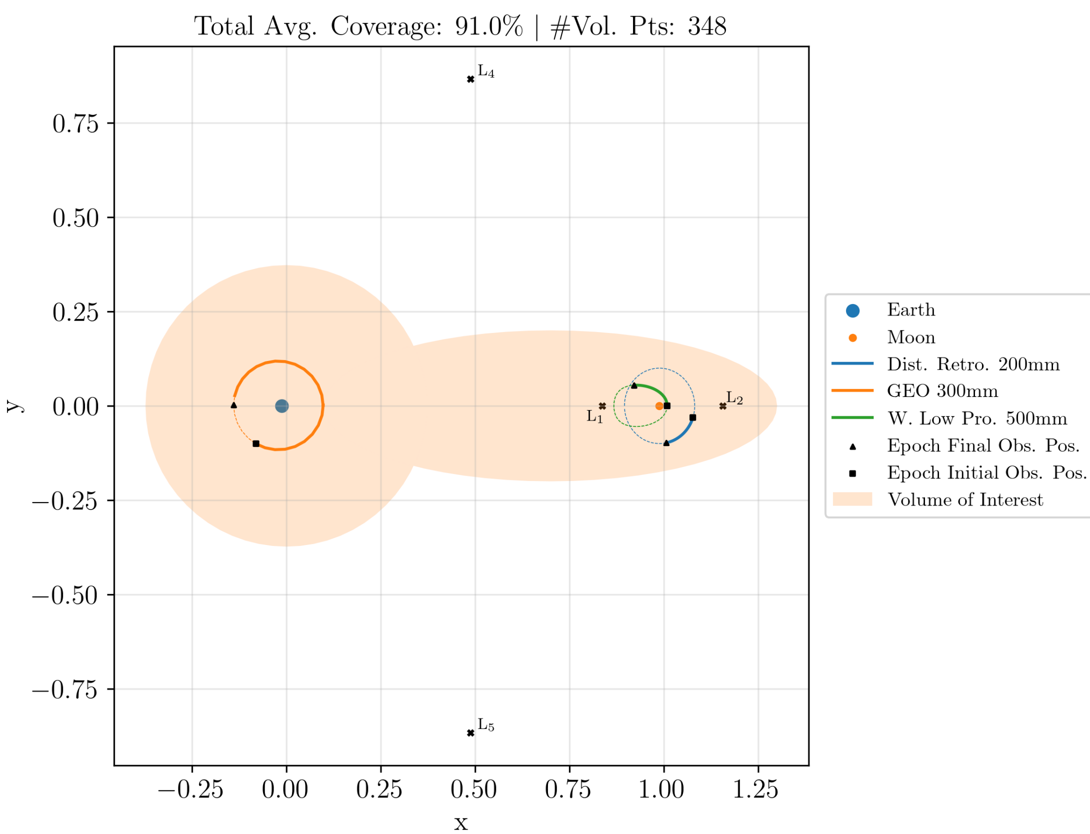


(b) Resultant approximate re-scaled Pareto front for the volume-oriented Cislunar architecture problem using the Large volume.

Figure 3.10: MO-MCTS results for the architecture-oriented Cislunar architecture problem using the Large volume.

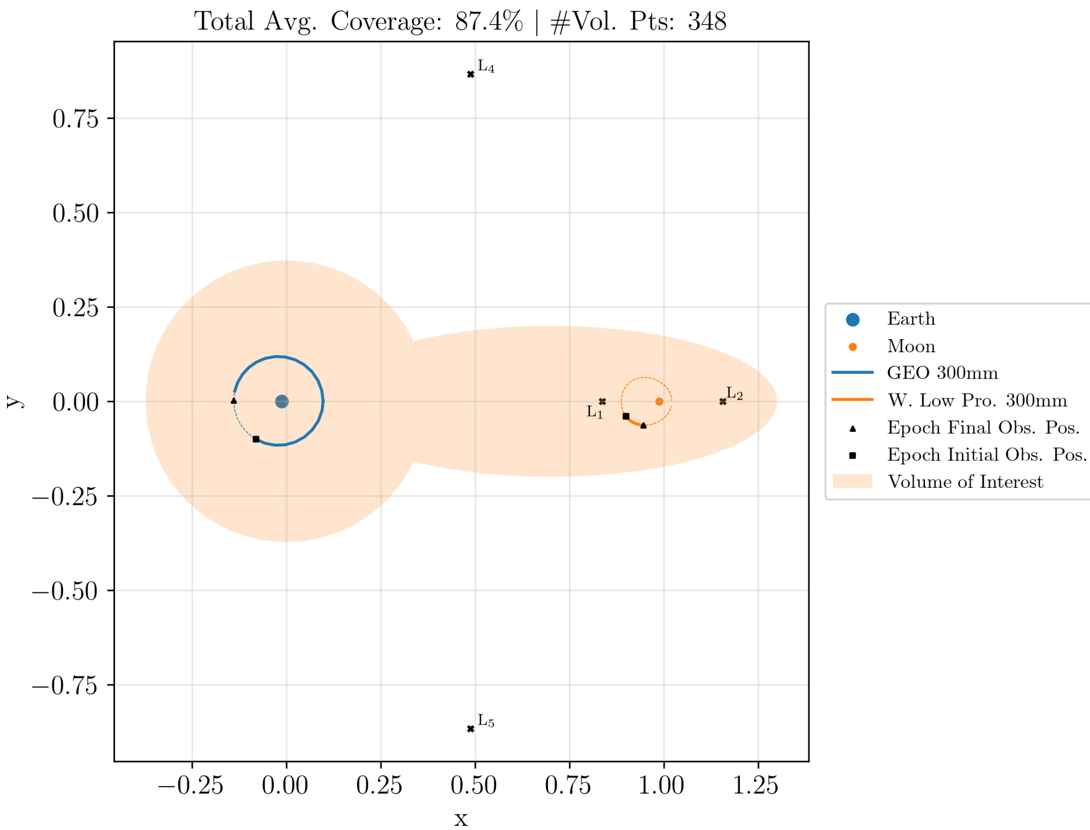


(a)

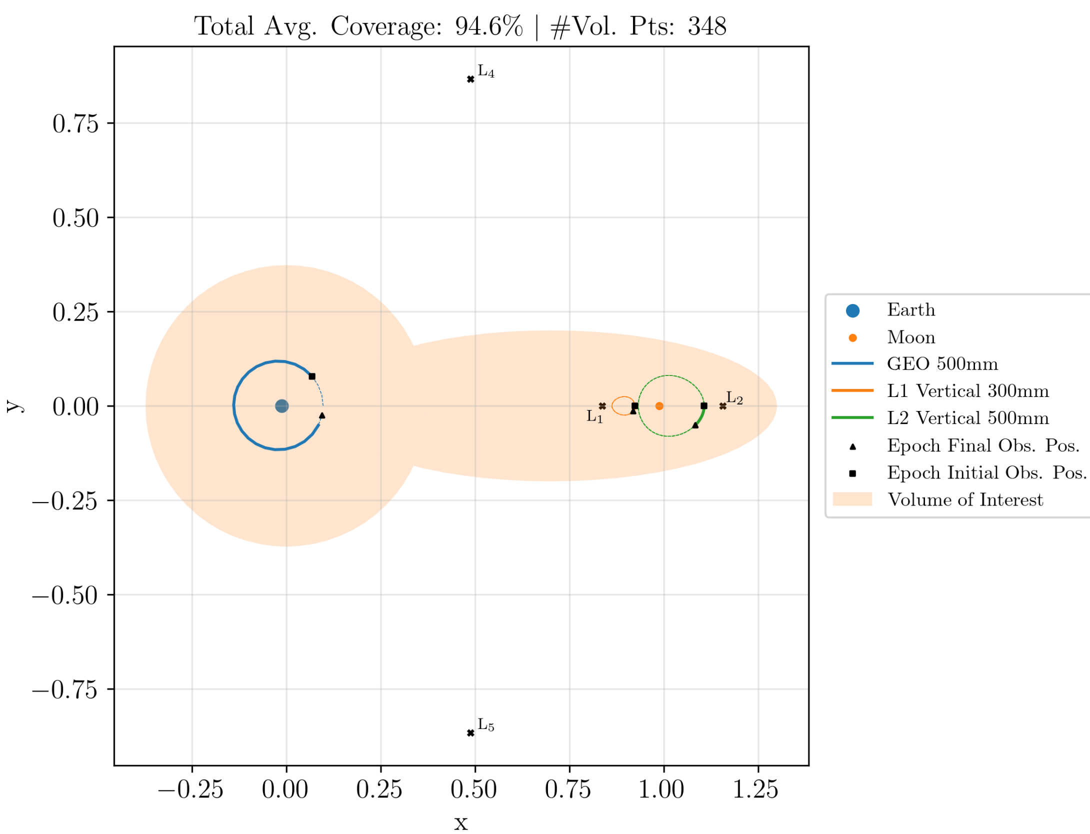


(b)

Figure 3.11: Four resultant Pareto-optimal Cislunar architecture layouts using the GEO Gateway volume with trajectory starting at the initial simulation epoch (continued on next page).

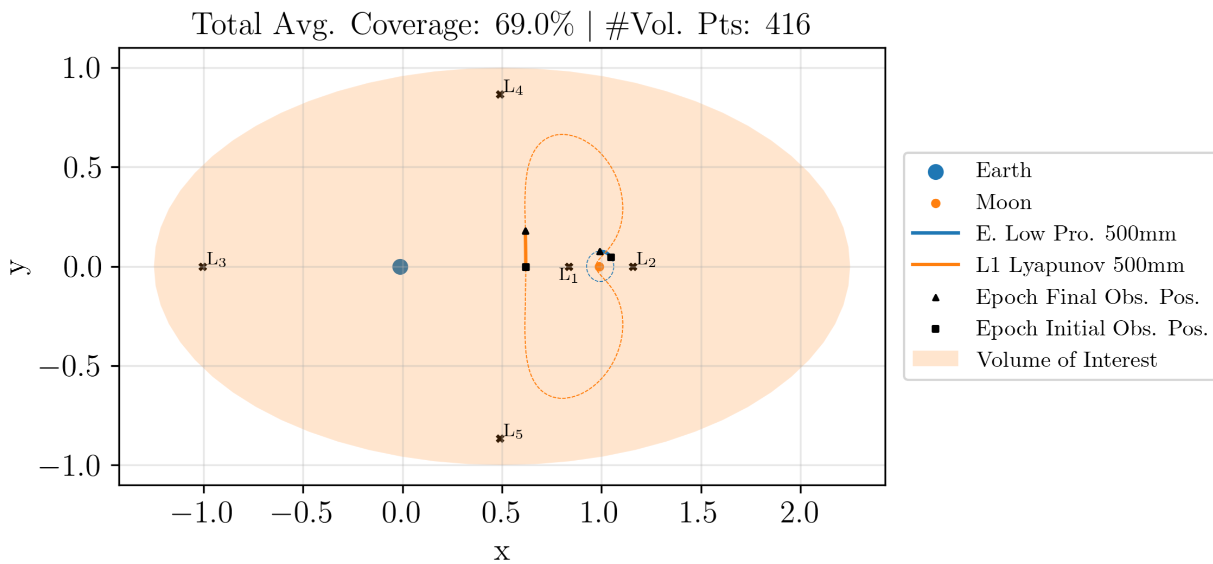


(c)

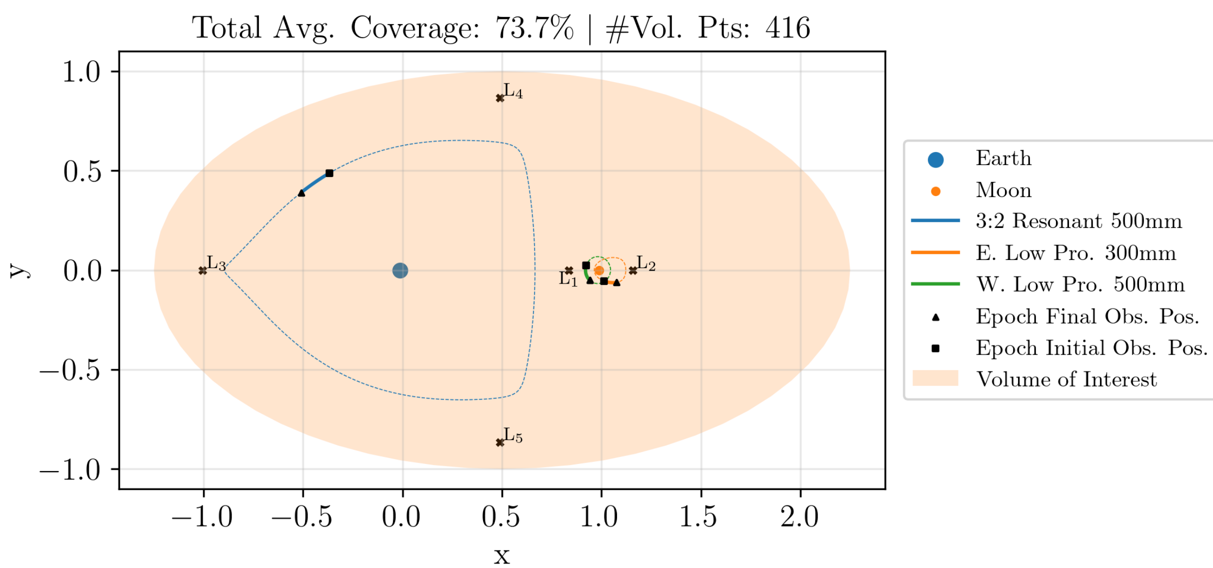


(d)

Figure 3.11: Four resultant Pareto-optimal Cislunar architecture layouts using the GEO Gateway volume with trajectory starting at the initial simulation epoch.

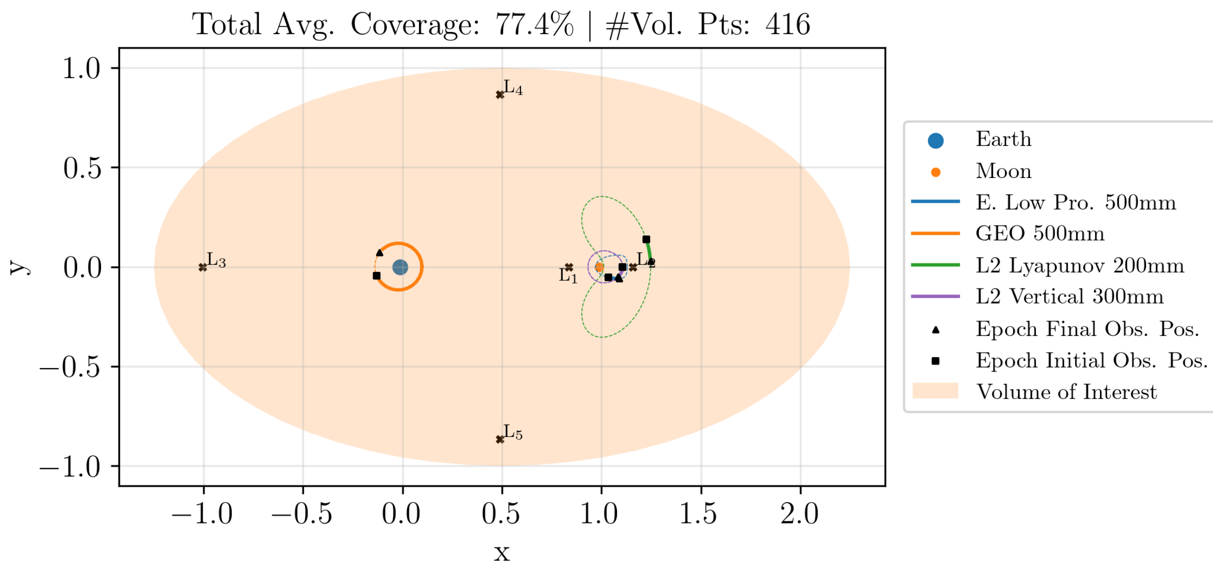


(a)

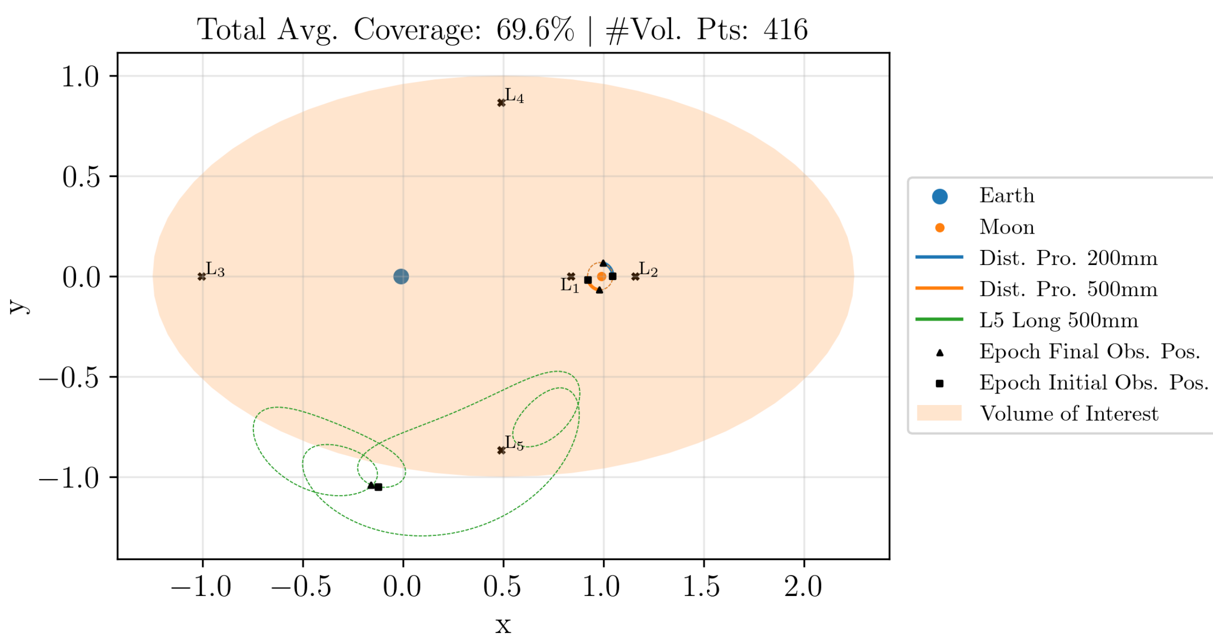


(b)

Figure 3.12: Four resultant Pareto-optimal Cislunar architecture layouts using the Large volume with trajectory starting at the initial simulation epoch (continued on next page).



(c)



(d)

Figure 3.12: Four resultant Pareto-optimal Cislunar architecture layouts using the Large volume with trajectory starting at the initial simulation epoch.

Chapter 4

Cislunar Architecture Design for Cooperative Agents

4.1 Cooperative Agents Interacting with Architectures

In this contribution, we augment our simple performance and cost objectives with a third: minimum delta-v required by a cooperative agent using the architecture for its own goals. At a high level, this objective aims to encapsulate the goals of an architecture user by ensuring that a Cislunar SSA architecture is constructed such that it is realistically cost effective for said user with respect to its mission objectives. Obviously, each Cislunar mission has varying objectives, whether those be fuel usage, transfer time, or target requirements. With that said, it can be reasonably assumed that all these objectives are critical, perhaps with varying weights, and total delta-v can be considered to sufficiently capture an overall mission cost. Furthermore, a CA model acting optimally and in concert with an architecture acts as a useful tool for analysis, as will be discussed later.

The first step in creating these minimum delta-v trajectories is to develop a CA model that is able to interact with a Cislunar architecture within a simulation, and define its parameters and intentions. We assume that the CA model has full knowledge of a given architecture layout (state of each observers and their capabilities). The goal of the cooperative agent is perform a transfer from GEO to the L1 Lagrange point or an L2 Lyapunov insertion point in the Earth-Moon CR3BP. The CA actuates an optimal control thrust profile to minimize delta-v usage, while also minimizing its cumulative apparent magnitude with respect to each observer in the architecture.

4.1.1 Cooperative Agent Model

Using indirect methods from the `Python` package `beluga` [5], a unique, optimal agent transfer trajectory from GEO to L1/L2 is generated for every architecture configuration visited during optimization. To facilitate rapid generation of a large number of optimal transfer trajectories, the CA is restricted to planar motion in the Earth-Moon plane. However, the architecture observers are computed from full 3-DOF motion. The optimal control objective for the agent is stated in integral form as

$$J = \int_{t_0}^{t_f} \|\boldsymbol{u}(t)\|^2 + \chi \sum_{i=1}^{n_{obs}} \text{apmag}(\boldsymbol{x}(t), \boldsymbol{x}_{Obs,i}(t), t)\,\mathrm{d}t \tag{4.1}$$

where $\boldsymbol{x}$ is the agent state, and $\boldsymbol{u}$ is the thrust acceleration input. The first term corresponds to an L^2-norm optimal control law, which reduces the overall control effort (and therefore delta-v) across the trajectory. The apmag$(\cdot)$ function computes the apparent magnitude of the agent relative to the i-th observer state $\boldsymbol{x}_{Obs,i}$ at time t, as described in Sections 2.1.2 2.1.3. The integrated total apparent magnitude over the trajectory is minimized in order to increase the likelihood of detection as seen by each observer. Note that this problem is non-autonomous, as the apparent magnitude is a direct function of time as it depends on the relative solar phase angle at each instant. Finally, a constant $\chi \in [0, 1]$ is multiplied into the objective for dimensional consistency. The value of this constant is changed during the continuation process to place varying levels of emphasis on the control or apparent magnitude objectives.

The objective in Eqn. (4.1) is minimized subject to the dynamical constraints of the CR3BP system, which can be stated generally as

$$\dot{\boldsymbol{x}} - \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t) = \boldsymbol{0} \tag{4.2}$$

The transfer trajectory must also satisfy boundary conditions at the initial and terminal points.

$$\boldsymbol{c}(\boldsymbol{x}(t_0), \boldsymbol{x}(t_f), t_0, t_f) = \boldsymbol{0} \tag{4.3}$$

Boundary conditions for both the L1 and L2 Lyapunov transfers are given in Table 4.1.

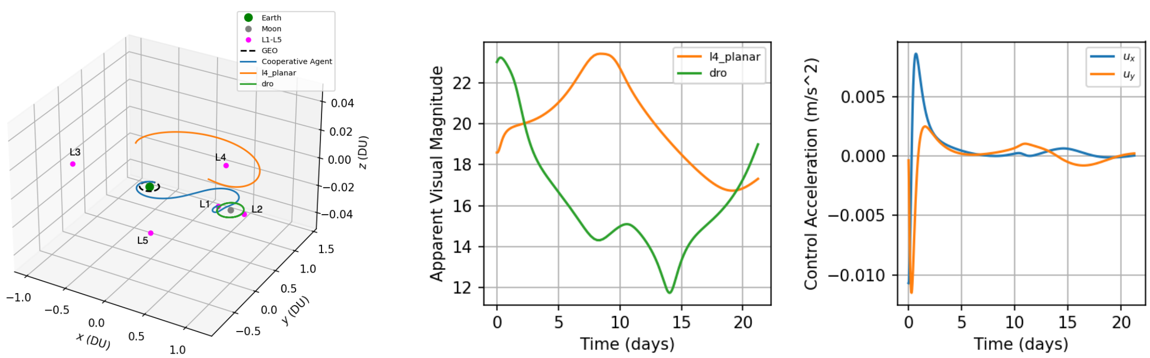
Table 4.1: Summary of cooperative agent boundary conditions for generated optimal transfer trajectories (assumes planar x-y motion).

Boundary	x	y	$\dot{x}$	$\dot{y}$
Initial	-μ	r_{GEO}	$-v_{\text{GEO}}$	0
Terminal (L1)	L1_x	L1_y	0	0
Terminal (L2 Ly.)	1.0399	0	0	0.6265

A sample trajectory for the optimal Cislunar agent is illustrated in Figure 4.1. The transfer trajectory begins from GEO departure and ends with insertion (zero relative velocities) at L1 or insertion into the L2 Lyapunov orbit. Both the apparent magnitude and control acceleration are also shown. Although the agent trajectory is assumed planar, incorporating full 3-DOF motion is unlikely to significantly affect apparent visual magnitude, which is assumed coincident with the x-y plane in the CR3BP system. These results illustrate a single potential architecture; however, the MOO approach must generate a new trajectory for every simulation epoch for each potential architecture visited during optimization.

Witin the simulation framework, the CA's trajectory generated for each simulation epoch is evaluated for detectability by the architecture using SNR as previously defined in Section 2.1.3. At the same time, the total control effort of the CA is recorded by integrating the control acceleration over the transfer time.

As the CA generates new trajectories, it also serves to provide us with a useful analysis tool, quantifying the existence of corridors within which varying levels of persistent detection is possible for an architecture. We expect that at any given simulation time, a Cislunar architecture can provide coverage to specific volumes of the space. A *persistent detection corridor* is represented by the existence of a trajectory that can realistically reside in these covered regions for the duration of its transit. Figure 4.2 displays this behavior with an arbitrary architecture. Along with a corresponding optimal trajectory, this analysis tool describes a time-varying volume of Cislunar space specific to an architecture layout and time. An architecture with many such corridors over time that are accessible within some range of delta-v's is one that may prove to be highly advan-



(a) Trajectory (b) Visual Magnitude (c) Control

Figure 4.1: Sample optimal trajectory to L1 of cooperative agent for two observers (L4 Planar and DRO).

tageous for Cislunar SSA. A point of nuance with these corridors is that there is likely to exist many discontinuities along such corridors with respect to coverage. To account for this, we may wish to further categorize corridors by their levels of persistent detection throughout a representative transfer along said corridor. Furthermore, corridors as described here are generated via a binary indicator distinguishing between "detectable" or "undetectable" regions. The CA in this work optimizes for the continuous value of apparent magnitude as an indirect method to represent detectability. As such, the CA model will only end up returning approximations of these corridors, while adhering to the CR3BP dynamics with control.

4.1.2 Coverage Metrics for Cooperative Agents

4.1.2.1 Cooperative Agent Control Effort

The control effort of the CA at simulation trajectory/epoch i is defined as the integral of the control acceleration over the transfer time, and is given by

$$\Delta \hat{v}_i = \int_{t_0}^{t_f} \|\boldsymbol{u}(t)\|^2 \, \mathrm{d}t \tag{4.4}$$

As in the previous contribution, we wish to maximize all objectives in the optimization schema, so we define the normalized control effort metric at the i^{th} simulation trajectory as

$$\Delta \hat{v}_i \quad = \max\left(1 - \tfrac{\Delta v_{CA,i}}{\Delta v_{max}}, 0\right) \tag{4.5}$$

$$\tag{4.6}$$

where Δv_{max} is a pre-defined maximum theoretical delta-v allowed for the cooperative agent. Given that for a single architecture layout we consider multiple epochs at which to generate an optimal trajectory for the CA, we also take the average of all the normalized delta-v's for use in our optimization, $\Delta \hat{V}$, where

$$\Delta \hat{V} = \mathrm{avg}(\{\Delta \hat{v}_1, ..., \Delta \hat{v}_j\}) \tag{4.7}$$

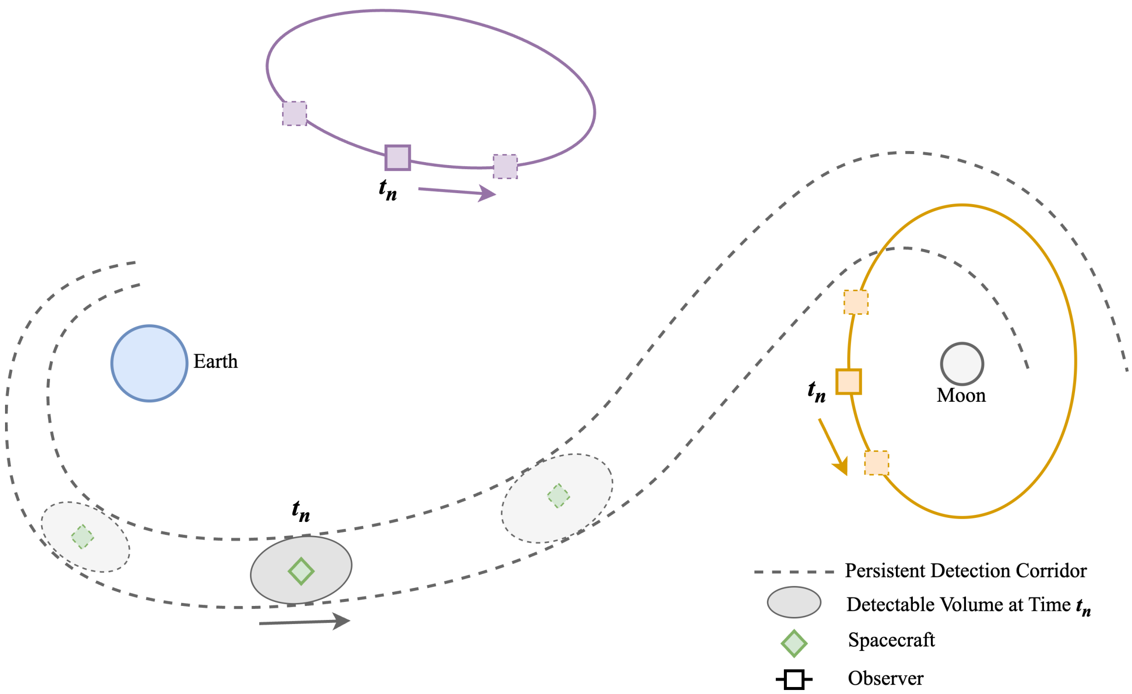


Figure 4.2: A single theoretical persistent detection corridor for an arbitrary architecture. The detectable volume of space by the architecture varies with time along the spacecraft's trajectory. In reality, there may be multiple corridors each corresponding to various levels of delta-v required to remain within the corridor.

for j simulation epochs.

4.1.2.2 Control Weighted Coverage Metric

Here, we use the average coverage metric as defined in Eqn (3.6), as well as introduce a modified version of this metric that encourages detection of the CA during large maneuevers, reflecting the concept of numerical information gain associated with tracking a maneuvering spacecraft. Mission operators benefit from making observations of an SO during large maneuvers, where missing an observation during these moments can lead to a significant loss of information, jeopardizing mission success. Given an array containing the history of control magnitudes from the SO, $\boldsymbol{U}_i$, and the detection mask $\boldsymbol{D}_i$ as defined in Eqn. 3.4 both at the i^{th} simulation, we define the control weighted coverage metric at the i^{th} simulation trajectory as

$$\eta_{cw,i} = \frac{\boldsymbol{U}_i \cdot \boldsymbol{D}_i}{\sum \boldsymbol{U}_i}. \tag{4.8}$$

The total weighted average coverage metric for architecture $\boldsymbol{X}$ is then

$$\eta_{cw}(\boldsymbol{X}) = \text{avg}(\{\eta_{cw,1}(\boldsymbol{X}), \eta_{cw,2}(\boldsymbol{X}), ..., \eta_{cw,j}(\boldsymbol{X})\}) \tag{4.9}$$

for j simulation trajectories.

4.2 Cooperative Agent Architecture Design Problem Formulation

Now, the optimization problem can be written as

$$\max_{\boldsymbol{X}} \left(\hat{C}, \Delta\hat{V}, \eta\right) \tag{4.10}$$

where $\hat{C}$ is the normalized architecture cost, $\Delta\hat{V}$ is the normalized control effort of the CA, and η is the average coverage metric (or average control weighted coverage metric) as defined in Eqn. 3.6.

4.2.1 MDP Formulation

As in the previous contribution, we can formulate the optimal Cislunar architecture design problem as a sequential decision problem in the form of an MDP. First, we define the state in our MDP, which is the same as in Eqn. 2.29. Next are the actions. In this formulation, the actions within our MDP guide the construction of an iteration of a Cislunar SSA architecture. We define two distinct types of actions, namely, "add" actions and "change" actions. "Add" actions take an architecture and add a new randomly generated observer Obs_i to the layout. In the previous contribution, this type of action was quite rigid as once an observer was added to an architecture layout it could not be altered. Thus, here we include "change" actions, which act on existing observers in an architecture to alter $Phasing$ or $Telescope\ Diameter$. The inclusion of these actions allow MO-MCTS to further refine the geometric qualities of an existing observer. During MO-MCTS, when sampling an action during the expansion step we weight the selection of "add" and "change" actions to be equal, but include logic to ensure that a single decision variable for an observer can only be changed once per architecture. This ensures that MO-MCTS does not waste time cyclically changing a single decision variable for an observer. Finally, the reward for our MDP is defined as the three dimensional vector $\boldsymbol{r} = [C_{normalized}, \Delta V_{total}, \eta]$ whose values are obtained for each architecture iteration from the core simulation.

4.2.2 Genetic Formulation

To formulate our MOOP for use within NSGA-II we introduce two distinct methods for navigating the state space, namely, crossover and mutation. Crossover involves taking two states within a population and randomly or heuristically combining them to form a new state containing decision variables from both parents. In this paper, we perform crossover on two states $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ as in Eqn. (2.29) by uniformly sampling a number of individual observers from both $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ such that the total number of observers in this set is less than our predefined maximum. The sampled observers then form a new architecture $\boldsymbol{X}_{child}$ whose parents are $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$. Mutation works on a single state by uniformly sampling a number of decision variables to alter in an observer, $Phasing$ and/or $TelescopeDiameter$ as in Eqn. (2.28). The usage of crossover and mutation are dictated by both elitism within NSGA-II as well as user defined probabilities. We introduce another slight alteration wherein during mutation, we allow with some probability a completely new random architecture to be added to the NSGA-II population. In brief tests, this has been shown to encourage NSGA-II to find even more diverse solutions within our specific MOOP.

4.3 Results and Discussion

4.3.1 Transfers to L1

To start, we generate architectures with the cooperative agent transferring to the L1 Lagrange point. We run our architecture optimization within the hybrid algorithm for a total of roughly 1,000 state evaluations, or architecture iterations. We choose to generate CA trajectories at two epochs over the same 60 synodic period time frame. Due to the complexity of the CA's trajectory optimization, a single iteration can take upwards of several minutes to complete. Successful trajectory generation is relatively frequent and provides us with a diverse set of results. In the optimization with transfers to L1, we use the average coverage metric from Eqn. (3.6) rather than the control weighted coverage metric from Eqn. 4.9, for the sake of comparison between the metrics.

Figure 4.3 shows the hyper-volume history per state evaluation for this run. Of note here is the fact that the HVI seems to quickly plateau once NSGA-II begins its optimization around iteration 400 (when NSGA-II takes over). This is likely due to the relatively small population size chosen for the hand-off from MO-MCTS to NSGA-II of only 28 individuals. There is inherently a trade space within NSGA-II between population size and number of generations. Larger population sizes may allow for more Pareto optimal solutions per generation, providing slightly larger increases to the HVI, but at the same time inherently limits the amount of mutation and crossover that can be performed on the population members. Our selection of 28 individuals was deemed to be an okay

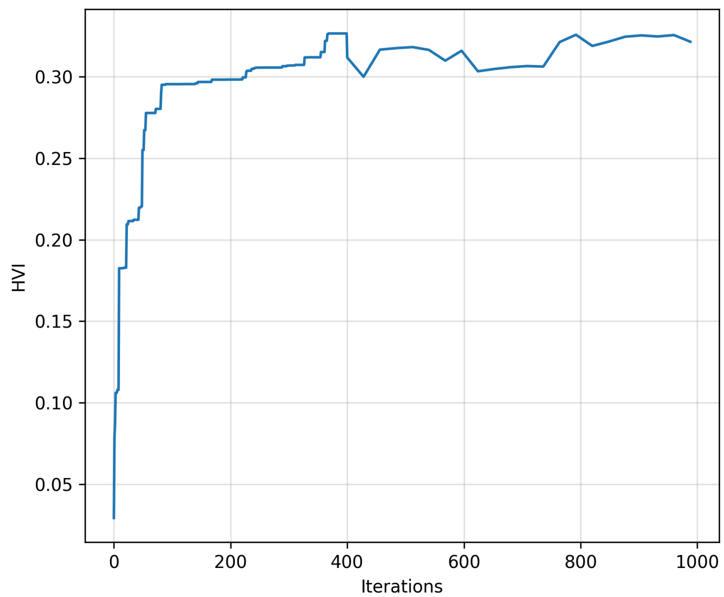


Figure 4.3: Hyper-volume indicator per state evaluation/iteration for L1 transfer.

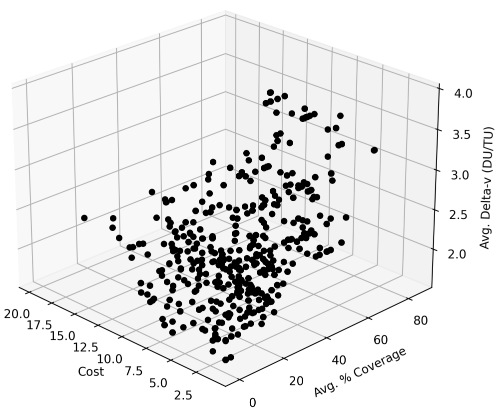
choice to balance these hyperparameters given the computational complexity of the simulation.

Figure 4.4 shows the diversity of points in objective space explored by our optimization run. As seen in the "Coverage vs Cost" projection plot in Figure 4.4b, there are a number of architectures that provide 40-80% coverage from costs of 1-10, but in no case does a single architecture of four observers provide 100% coverage. This indicates that the limit of four observers in our problem formulation may be limiting in coverage, and that the cooperative agent may be expending more delta-v to make up for the architectures' shortcomings. This behavior also helps to confirm that our CA model is acting as expected by maximizing detection by the given architecture.

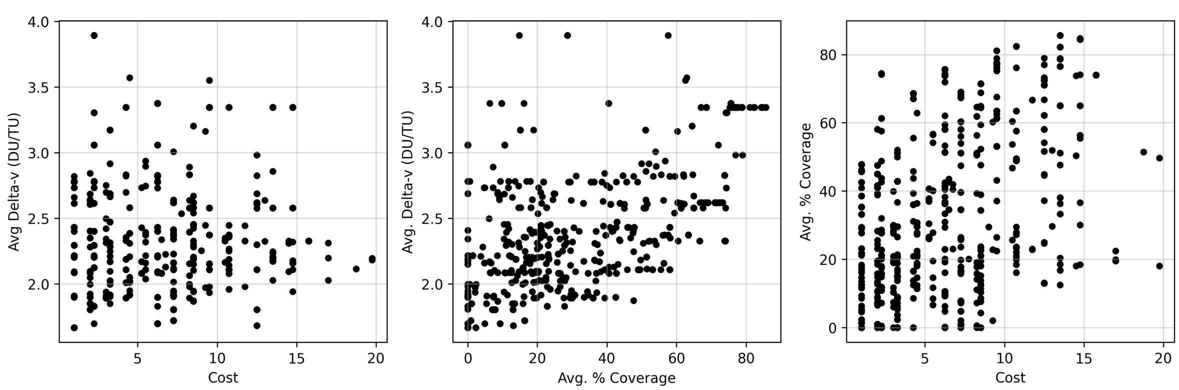
The Pareto front is shown in Figure 4.5a and represents the trade space of solutions, with behavior we generally expect from this problem. This Pareto front represents all the non-dominated solutions uncovered during the optimization process. We can dissect the trade spaces by looking at the projections of the Pareto front onto their respective axes as in Figure 4.5b. For example we can look at the "Coverage vs Cost" plane to see the behavior indicating that, in some cases, we may not have to incur a relatively large cost constructing an architecture to maintain near perfect coverage of a cooperative agent. However, as seen in the "Delta-V vs Cost" and "Delta-V vs Coverage" planes, such a low cost architecture while maintaining high coverage tends to require a significantly greater delta-v expenditure by the cooperative agent. On the other hand, if we wanted to reduce the total effort required by the cooperative agent while maintaining sufficient coverage we would have to construct a significantly more costly architecture. Even so, there are few architectures here that provide coverage over 75%. By increasing the maximum allowed number of observers per iteration up from four we may see even more improvements in CA delta-v usage, at the expense of a more costly architectures. The results represented in this tradespace are intuitive, and emphasize how the cost of conducting effective Cislunar SSA may be distributed amongst numerous potential agents. Inevitably, the layout and capabilities of a Cislunar SSA architecture will directly impact aspects of mission design within the regime.

Figure 4.7 shows a number of the resultant architectures from the Pareto front as well as the optimal trajectory generated by the cooperative agent at a single simulation epoch. As in [63], there seems to be a shared quality of these architectures where each observer is spread out in space. Most notably, however, are the diversities of transfer geometries created by the CA in its attempt to minimize it cost function in Eqn. (4.1). Most trajectories follow the same general "S" curve in their transfer to L1 in part due to the constraints on the initial position being in a GEO orbit. By varying this constraint in future work we may see a broader diversity in transfer geometry. Regardless, there are distinct differences among the trajectories recovered, and we can see that the cooperative agent expends significant effort in some cases to ensure detection by the observers.

While in [**?**] we saw frequent use of L1 and L2 Lyapunov orbits in the solution space, here we see frequent usage of L4 and L5 axial orbits as shown in Figure 4.6. These L4 and L5 orbits traverse a large volume of Cislunar space, allowing for the cooperative agent more freedom to traverse the region while uncovering unique persistent detection corridors. Furthermore, an observer not being restricted to the (x, y) plane while providing coverage to a planar trajectory likely reduces the impact of Earth, Moon, and Sun pointing constraints. It may also be that at the two simulation epochs at which trajectory optimization was performed, the solar phase angle was oriented in such a way that the viewing geometry from near the L4 and L5 Lagrange points was more conducive to

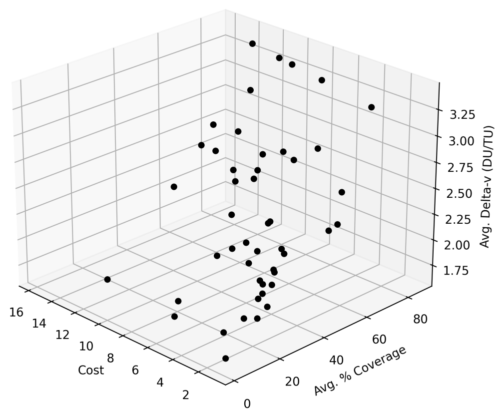


(a) Objective history for all visited states in which a transfer trajectory was successfully generated in the L1 transfer.

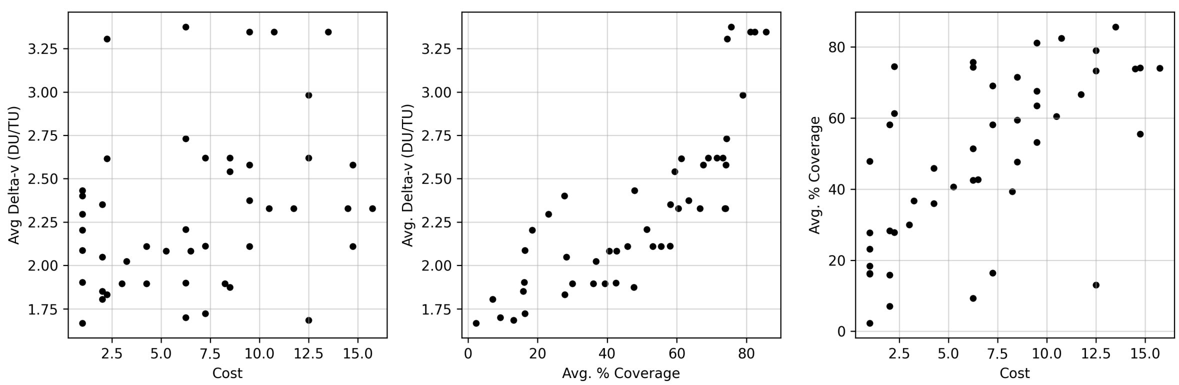


(b) Objective history projections in the L1 transfer.

Figure 4.4



(a) Final Pareto-optimal solutions in objective space in the L1 transfer.



(b) Final Pareto front solutions in objective space with their projections onto each axis in the L1 transfer.

Figure 4.5

detection of the cooperative agent. In this sense, evaluating architectures over only two epochs is limiting in that we are not able to fully evaluate the relative geometry between the observers and the cooperative agent over a long period of time.

The collection of trajectories generated by the CA across all iterations is shown in Figure 4.8. Of interest is the fact that there are some transfers corresponding to Pareto-optimal architectures

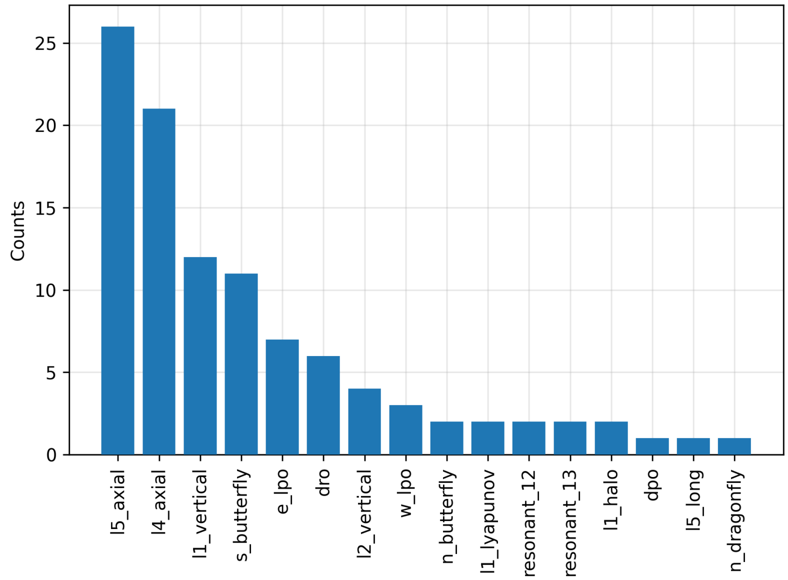


Figure 4.6: Counts of orbit families used by observers within the Pareto-optimal architectures in the L1 transfer.

that seem to expend a great deal of delta-v to remain detectable near L4. This may be slightly surprising, as such a transfer may be undesirable from a mission design perspective. However, we know from the tradespace in Figure 4.5b that many other optimal architectures exist at similar coverage levels, just with increased architecture cost to lower the CA delta-v. This process allows designers to better plan for future Cislunar missions, as some transfers produced by the CA during optimization may result in favorable coverage and architecture cost, but be completely infeasible with respect to control effort or geometry. We recall that each transfer is unique and locally optimal with respect to a specific architecture and simulation epoch. So, selecting a more feasible transfer with a lower delta-v or more favorable geometry will inevitably require significant changes to an architecture layout to maintain coverage. Similarly, changes to an architecture affecting layout or cost will inevitably require altering the CA's transfer as it may no longer be locally optimal to the altered layout. These interactions must be considered at all phases during the design and construction of a Cislunar SSA architecture.

Finally, Table 4.2 contains a summary of the resultant delta-v's from Pareto-optimal solutions. Based on the delta-v's and transfer times, these trajectories are reminiscent of direct lunar transfers [75] like those used in the Apollo missions and Artemis 1. These types of missions are expensive, and there is growing interest in low-energy transfers using ballistic lunar transfers such as the CAPSTONE mission, with a total duration of about 140 days. Furthermore, the trajectory optimization process performed by the cooperative agent is inherently multi-objective as well, and in this work we do not perform an analysis of the trade space between delta-v, transfer duration, and coverage when selecting a trajectory.

Table 4.2: Summary of cooperative agent transfer delta-v's and durations in the L1 transfer. These correspond to direct, high-energy transfers similar to those used in the Apollo missions and Artemis 1. Magnitudes of delta-v's here are similar to those in the L2 Lyapunov transfer case.

	Δv *km/s (DU/TU)*	Duration *days (TU)*
Min	1.709 (1.668)	5.250 (1.209)
Avg	2.487 (2.427)	12.057 (2.777)
Max	3.587 (3.501)	21.708 (4.999)

4.3.2 Transfers to L2 Lyapunov Orbit

Next, we optimize architectures with the cooperative agent transferring to an L2 Lyapunov insertion point. Similar to before, we run our optimization for 1,000 iterations. Here, though, we evaluate each individual architecture over 5 simulation epochs similarly spaced over the 60 Earth-Moon synodic periods. In this simulation environment, we use the control weighted average coverage metric from Eqn. (4.9). From using the control weighted coverage metric, we prioritize providing coverage to the cooperative agent during large maneuvers, and would expect to see a greater occurrence of GEO observers, since a large amount of delta-v is expended to escape from GEO.

Figure 4.9 shows the resultant HVI history for this run. The magnitude of the HVI is slightly higher than that from the L1 transfer due to the higher values of the weighted average coverage metric. We see decent improvement over iteration in this scenario similar to before.

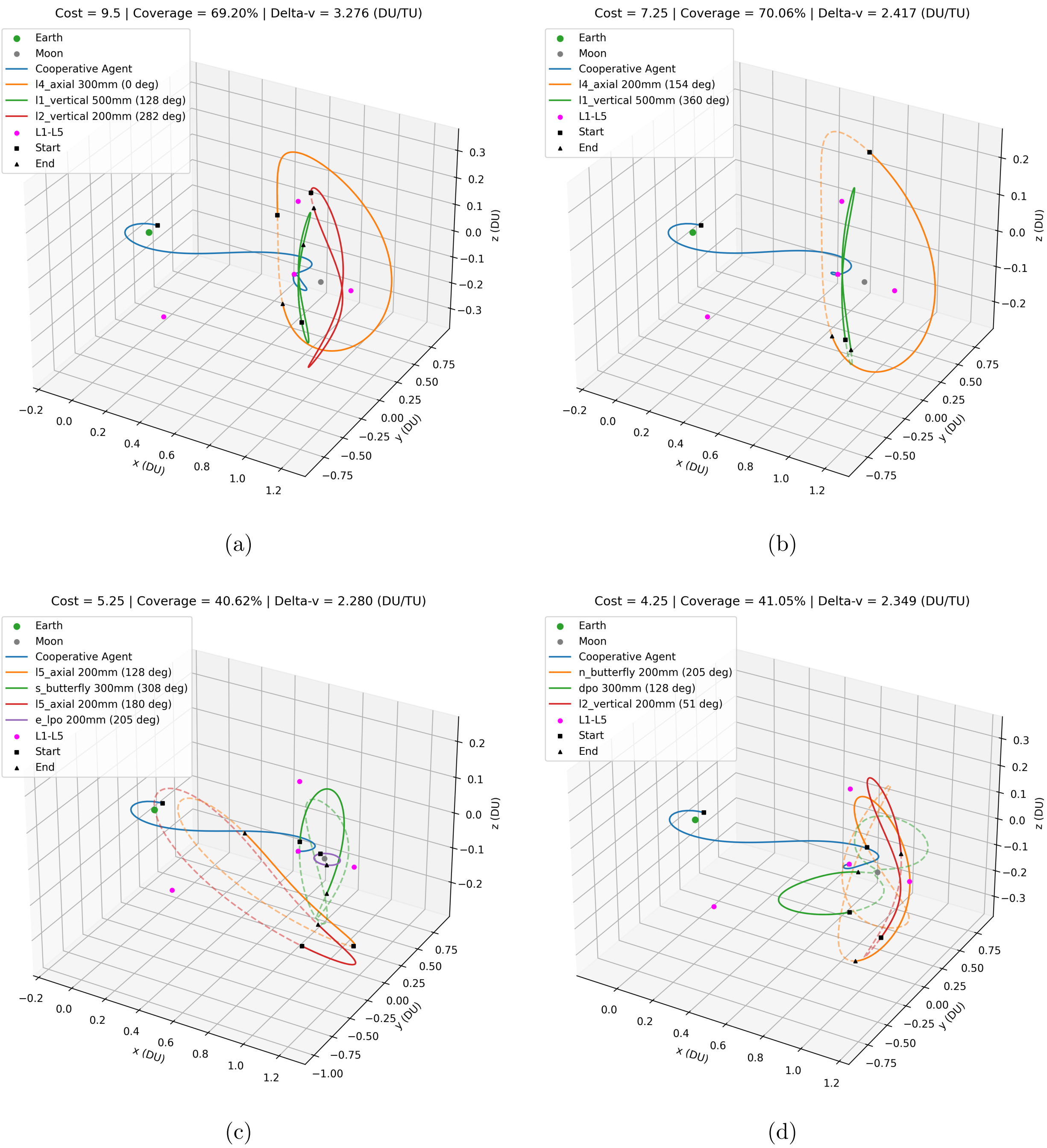


Figure 4.7: Selection of trajectories and architecture layouts at a single epoch from the set of Pareto-optimal solutions in the L1 transfer. Coverage and delta-v values correspond to the individual epoch shown and not the over all epochs considered for optimization. Observer state histories during the transfer are shown in solid, with their full orbits shown in dashes. The relative phasing of observer orbits is given in the legend.

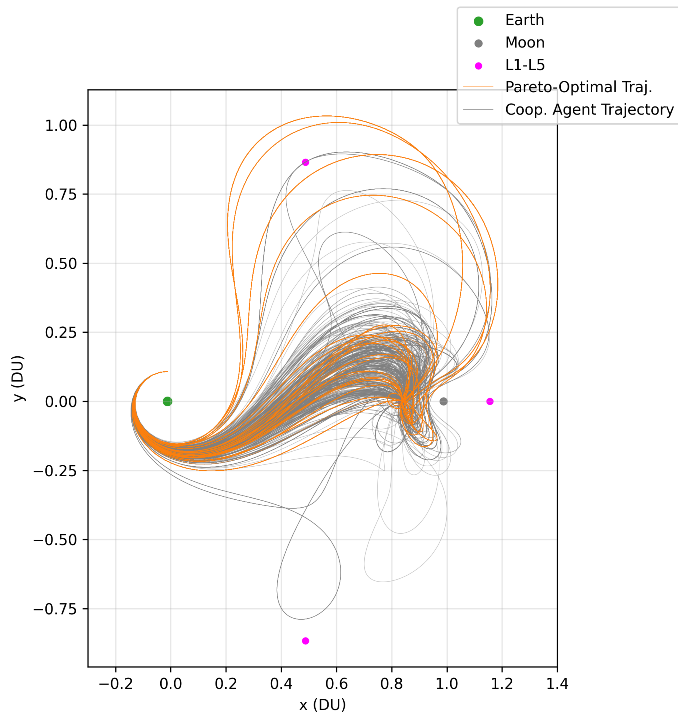


Figure 4.8: All trajectories generated throughout the run with those corresponding to Pareto-optimal solutions highlighted in orange in the L1 transfer.

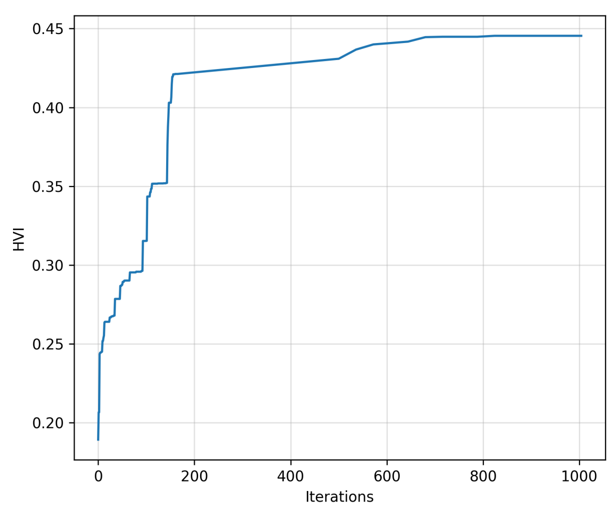


Figure 4.9: Hyper-volume indicator per state evaluation/iteration for L2 transfer.

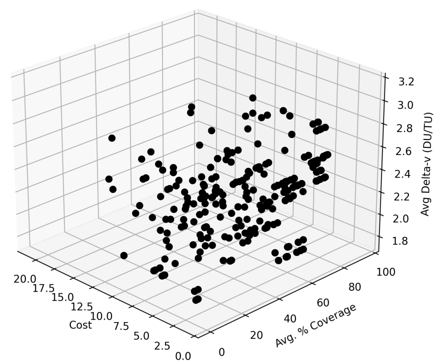
Figure 4.10 once again shows the entire state space searched during optimization, displaying a similar diversity of solutions as seen for the L1 transfer scenario. Notably, however is slightly lower maximum magnitude of delta-v's used by the CA. This is likely due to the zero velocity final condition used for the L1 Lagrange point transfer. Figure 4.11 shows the resultant Pareto front showcasing the tradespace between objectives in this scenario. Notably, we end up with significantly less Pareto-optimal solutions than in the L1 transfer case. This could be due to the increased number of simulation epochs in the L2 transfer case, causing less variation in the evaluation of each architecture. Regardless, we see similar behavior within the tradespace projections shown in Figure 4.11b. Lowering the cost of an architecture generally results in larger delta-v expenditure in order to achieve high weighted coverage. On the other hand, if we want to lower the delta-v expenditure, we either have to greatly increase the cost of an architecture, or settle with subpar coverage.

Figure 4.12 shows the distribution of selected orbits within the Pareto-front. As discussed previously, the use of the average control weighted coverage metric dramatically increases the use of GEO observers due to a large amount of control effort used for the CA to escape from GEO. Similar to the L1 transfer, we also see a high usage of non-planar orbits. The geometry of these orbits allow the architecture to better obtain visual coverage of the CA in combination with the assumed planar solar rays.

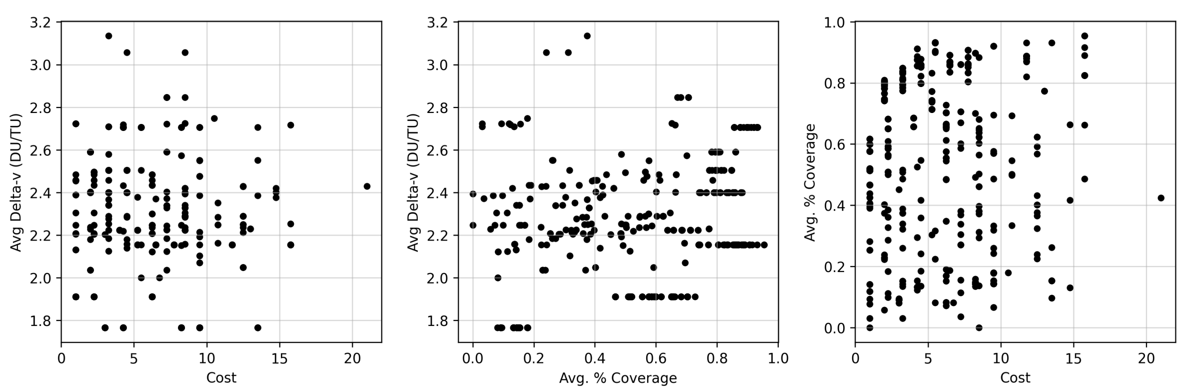
Figure 4.13 shows a selected number of Pareto-optimal architectures and the corresponding CA optimal transfer to the L2 Lyapunov insertion point. The architectures all contain a GEO observer to provide the initial coverage, and rely on out of plane orbits for the remaining transfer duration. Finally, the entire set of optimal transfers to the L2 Lyapunov insertion orbit are shown in Figure 4.14, with those corresponding to the Pareto-optimal architectures highlighted in orange.

4.3.3 Clustering Analysis

As seen in Figures 4.7 and 4.8, through the iterative process of constructing architectures, we uncover a diversity of transfer trajectories with varying levels of delta-v, each uniquely optimized for a specific architecture. We wish to explore methods that utilize these trajectory-architecture pairs and their resultant objectives to better understand the underlying dynamics of Cislunar architecture design. Here, we outline two methods of novel analysis using standard clustering techniques that

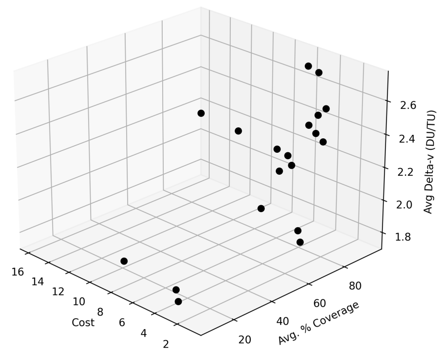


(a) Objective history for all visited states in which a transfer trajectory was successfully generated in the L2 transfer.

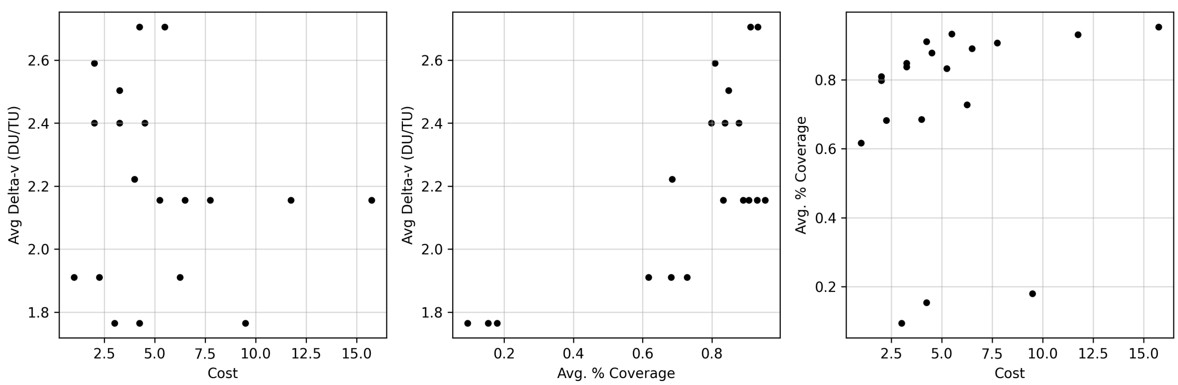


(b) Objective history projections in the L2 transfer.

Figure 4.10



(a) Final Pareto-optimal solutions in objective space in the L2 transfer.



(b) Final Pareto front solutions in objective space with their projections onto each axis in the L2 transfer.

Figure 4.11

aid in the categorization of Cislunar architectures based on observers' orbital families as well as the architectures evaluated in objective space. Through these clusters, we can generate an intuition on what subsets of decision variables often lead to optimal results in our environment.

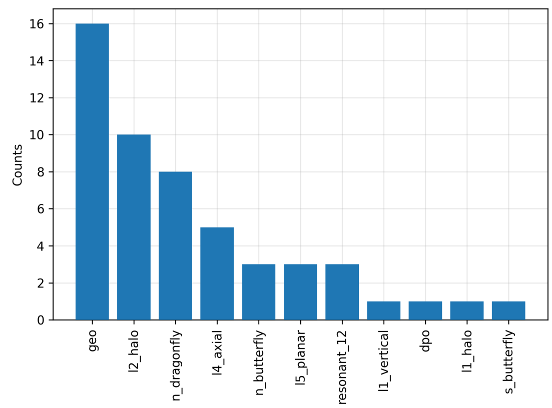


Figure 4.12: Counts of orbit families used by observers within the Pareto-optimal architectures in the L2 transfer.

4.3.3.1 Clustering Observer Orbit Families

First, it is useful to note that a *family* of periodic orbits in the CR3BP is essentially a cluster of orbits that share similar characteristics across different spaces. There is variation within each of these families, like period and Jacobi energy, but in general, each family member resembles the others in an intuitive way. As such, we can begin analysis with clustering architectures by observer orbit family labels. For all clustering analysis that follows, we use the results from the L2 transfer optimization.

To construct the feature vector here, we simply assign integer labels to each orbital family considered in the optimization. As an example, we may have an architecture state $\mathbf{X}_i$ with two observers

$$Obs_1 = [L1\ Halo, 0, 0.4, 200mm]$$

$$Obs_2 = [GEO, 0, 0.2, 500mm]$$

Knowing that no architecture has more observers than the predefined maximum (here chosen to be four), we can construct the corresponding row of the feature vector here to be

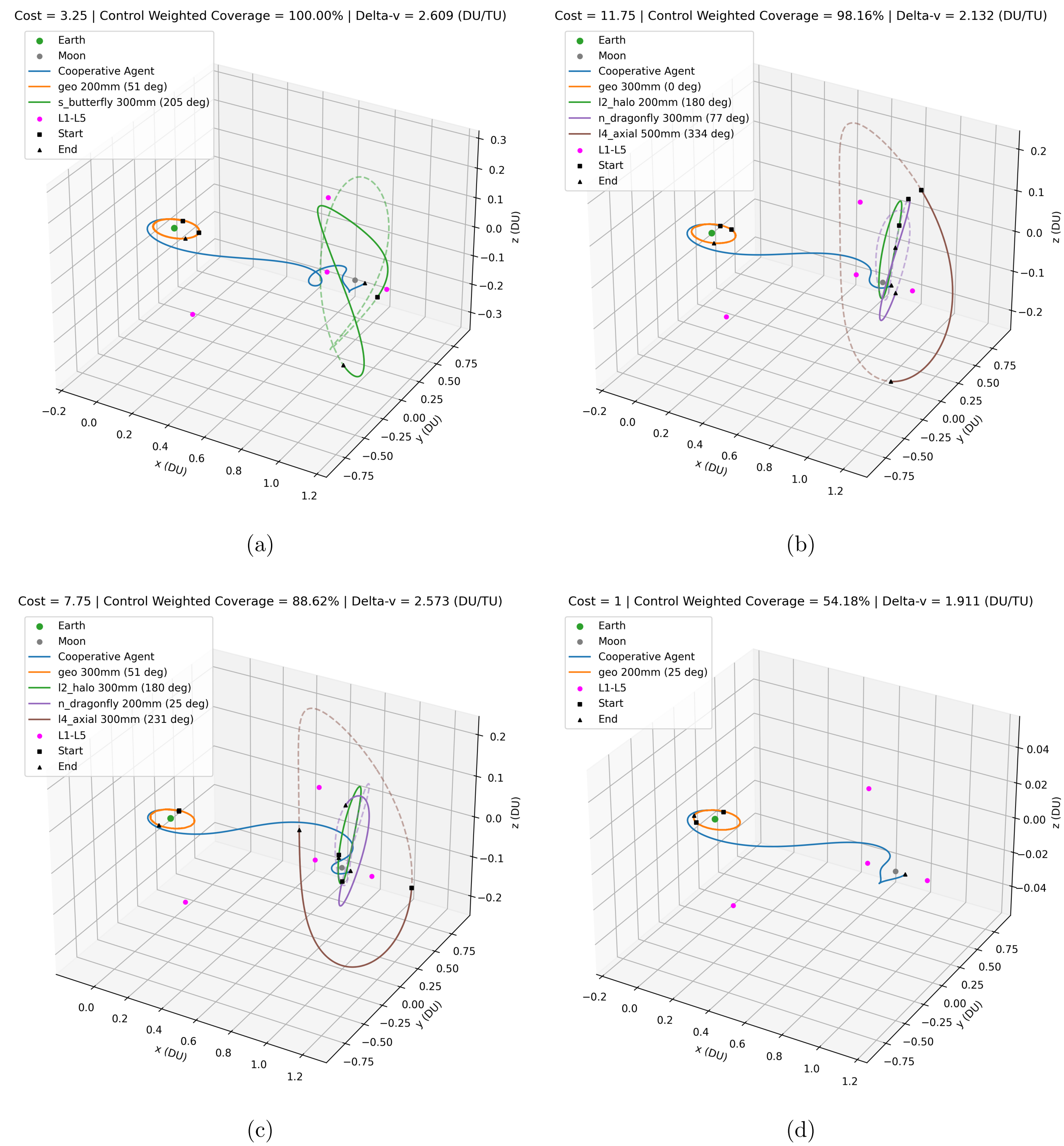


Figure 4.13: Selection of trajectories and architecture layouts at a single epoch from the set of Pareto-optimal solutions in the L2 transfer. Coverage and delta-v values correspond to the individual epoch shown and not the over all epochs considered for optimization. Observer state histories during the transfer are shown in solid, with their full orbits shown in dashes. The relative phasing of observer orbits is given in the legend.

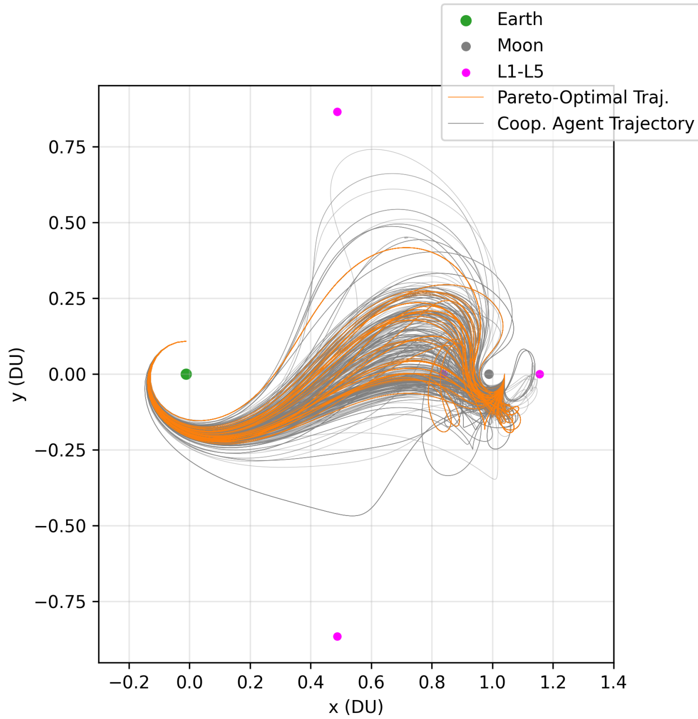


Figure 4.14: All trajectories generated throughout the run with those corresponding to Pareto-optimal solutions highlighted in orange in the L2 transfer.

$$\Gamma_i = [1, 5, -1, -1]$$

where "1" and "5" are the arbitrary integer labels for "L1 Halo" and "GEO" respectively. The label of "-1" indicates no observers in the 3rd and 4th allowed "slots" in the architecture.

Given that this data is categorical, we must use a distance metric that can compare how similar two sets of orbital families are. For this, we uses the Jaccard index, also referred to as the Jaccard similarity index. The Jaccard index provides a measure for how similar two sets of objects are and is used often in clustering applications for comparing groups of categorical data [87]. Since architectures in our environment may have multiple observers within the same family, we introduce a modification to the Jaccard index so that it properly reflects repeated labels within multisets:

$$J(A, B) = 1 - \frac{|A \cap_+ B|}{|A \cup_+ B|} \tag{4.11}$$

where A and B are multisets (may have repeated elements), and $+$ denotes taking the multiset intersection or union. As an example, for two multisets representing rows of our feature vector, $A = \{1, 5, -1, -1\}$ and $B = \{1, 5, 1, 4\}$ the Jaccard index of these (ignoring the "-1" labels) is

$$J(A, B) = 1 - \frac{|\{1, 5\}|}{|\{1, 1, 1, 4, 5, 5\}|} = 1 - \frac{1}{3} = \frac{2}{3}$$

So, we can say that the Jaccard index of these two representative architectures representing the distance between their observer families is about 0.667. By using the modified Jaccard index, we expect to find clusters of architectures with similar orbital families and with similar number of observers. Obviously, by clustering solely on a single decision variable per observer, we may miss structures within the explored state space corresponding to initial orbit phasing and telescope diameter. But by partitioning architectures into groups of similar orbits, we can more closely analyze the distributions of other decision variables and how they affect the resultant architecture performances.

To perform the clustering, we use the K-Medoids algorithm, implemented in `Python` via the `sklearn` library [80, 74]. K-Medoids is a clustering algorithm that separates data into a predefined number of clusters, using the actual data points as cluster centers, or medoids. It works by minimizing the sum of dissimilarities (here the Jaccard indices) between data and their nearest medoids. This method is particularly advantageous for categorical data as it can use more complex distance metrics to determine clusters. Furthermore, while there exist clustering methods that do not require the number of clusters to be known a priori, such as HDBSCAN, using an algorithm requiring a predefined number of clusters can be helpful in simplifying the analysis. In testing different algorithms for this clustering, we found that K-Medoids also resulted in the most distinct groupings compared to others.

4.3.3.2 Orbital Family Results

Once the full feature vector $\mathbf{\Gamma}$ was compiled, a distance matrix was computed using the Jaccard index in Eqn. (4.11), representing the dissimilarity or distance between each set of observer families. Then, K-Medoids was run on the distance matrix with a predefined number of clusters of `n_clusters=10`. The results of this clustering are shown in Table 4.3, where only the most distinctive families per cluster are shown, selected heuristically. Most clusters returned contain more than four distinct families, but the results are summarized for simplicity. Of note in these results is the use of GEO observers in many architectures due to the use of the control weighted coverage metric. This is reflected in comparing the counts of orbit families used in the Pareto fronts between the L1 and L2 transfer cases in Figures 4.6 and 4.12, respectively, wherein the former has no GEO observers and the latter relies heavily on them.

Also shown in Figure 4.12, the set of Pareto-optimal architectures also contains many L2 Halo observers. Cluster 9 contains mostly GEOs and L2 Halos, so we can begin by focusing on the member architectures of this cluster. To begin, we can take note of cluster members that represent Pareto-optimal architectures, shown in Table 4.4. Within this cluster there are multiple instances of architectures with GEO and L2 Halo observers with varying phases as shown in Figure 4.15. However, only very specific combinations of phasings and telescope diameters lead to optimal results.

As shown in Table 4.3, cluster 0 also contains a number of architectures with GEO and L2 Halo observers, although they are all combined with Northern Dragonfly and L4 Axial observers as well. We can look at a sample of the Pareto-optimal architectures represented in cluster 0, and note how adding new observer families to an existing architecture can change how each observer must interact with the others to retain its Pareto-optimality. Table 4.5 shows two selected Pareto-optimal architectures from cluster 0. It is important to note that the L2 Halo orbits here have a slightly longer period than those in cluster 0, but regardless we note how adding two observers in different orbital families requires rather substantial changes to the GEO and L2 Halo orbits' initial phasings

Table 4.3: Most distinct orbit families per cluster from K-Medoids. ***# Mem.*** refers to the number of member architectures per cluster. Note that in some clusters there may be more than one observer in an orbit family per architecture leading to a higher count of orbit members, as in cluster 9.

Cluster: # Mem.	**Family Name**	**Count**
Cluster 0: 94	GEO	93
	L2 Halo	93
	N. Dragonfly	93
	L4 Axial	93
Cluster 1: 172	GEO	142
Cluster 2: 94	1:2 Res.	94
	L5 Planar	89
	N. Butterfly	89
Cluster 3: 34	DPO	34
Cluster 4: 70	S. Butterfly	74
	GEO	46
Cluster 5: 36	L1 Halo	36
	DPO	36
	GEO	35
	L1 Vertical	35
Cluster 6: 77	N. Dragonfly	77
	GEO	69
Cluster 7: 18	L2 Vertical	20
	1:3 Res.	4
Cluster 8: 27	L5 Planar	20
Cluster 9: 84	L2 Halo	127
	GEO	76

Table 4.4: Pareto-optimal architectures represented in cluster 9. ***Orbit Idx.*** refers to the unique orbit label within its relevant family.

Architecture ID	**Orbit**	**Orbit Idx.**	**Initial Phasing**	**Tel. Diameter (mm)**
Architecture 1	GEO	0	0.357	300
	L2 Halo	2	0.143	200
	L2 Halo	2	0.429	300
Architecture 2	GEO	0	0.571	200
	L2 Halo	2	0.571	200
	L2 Halo	2	0.429	300
Architecture 3	GEO	0	0.429	200
	L2 Halo	2	0.429	200

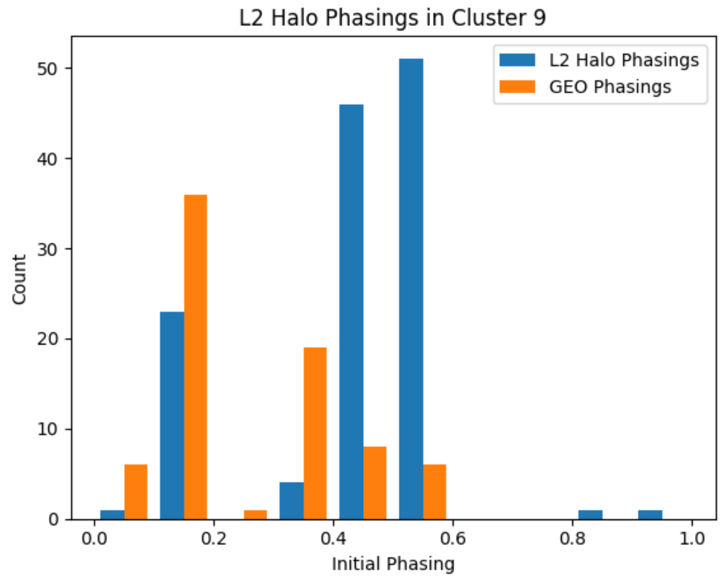


Figure 4.15: Counts of GEO and L2 Halo initial phasings in cluster 9.

and telescope diameters to remain Pareto-optimal. When designing SDA architectures in Cislunar space, care must be taken when adding new observers to an existing layout so that performance gains can be maximized with the inclusion of new observers. This may inevitably require alterations to existing observers' phasings in order to achieve an optimal increase in performance.

4.3.3.3 Clustering in Objective Space

Another method of clustering that can be helpful when determining how different architecture layouts affect performance is to group architectures by their objectives. We may expect to find that certain regions of objective space tend to be inhabited by architectures with observers in similar families or telescope diameters. From the histogram in Figure 4.12, we would expect that a cluster may exist that contains a large amount of GEO and L2 Halo orbits.

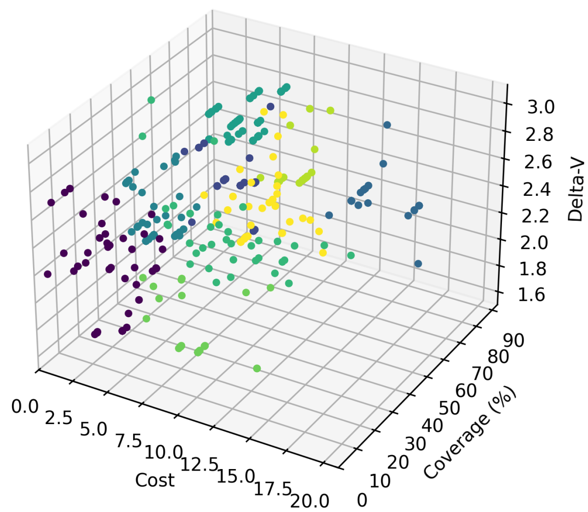
To start, we simply create a feature vector $\mathbf{\Gamma}$ the rows of which contain the normalized objectives $[\hat{C}, \Delta\hat{V}, \eta]$ for each architecture visited during the L2 transfer optimization case. Then, we use K-Means clustering with an L2 Euclidean norm distance metric from `sklearn` [80] with a predefined number of cluster `n_clusters=10` to make comparison to the previous clusters intuitive.

Table 4.5: Two Pareto-optimal architectures represented in cluster 0. ***Orbit Idx.*** refers to the unique orbit label within its relevant family.

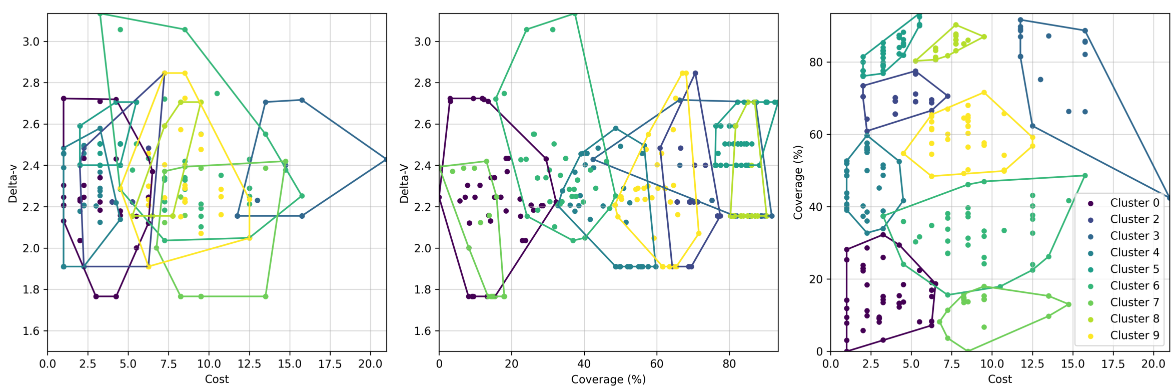
Architecture ID	Orbit	Orbit Idx.	Initial Phasing	Tel. Diameter (mm)
Architecture 1	GEO	0	0.143	500
	L2 Halo	1	0.071	300
	N. Dragonfly	1	0.071	200
	L4 Axial	12	0.642	300
Architecture 2	GEO	0	0.642	200
	L2 Halo	1	0.429	200
	N. Dragonfly	1	0.071	200
	L4 Axial	12	0.643	300

K-Means is used here rather than other methods such as HDBSCAN in that it often resulted in clusters that were more intuitive, and we do not necessarily need complex methods for dealing with phenomena such as noise in our data set. The result of clustering in the objective space is shown in Figure 4.16, with convex hulls added to aid in the visualization.

As seen in Figure 4.16, clusters seem to most clearly defined in the coverage-cost plane, with more overlap in other two. This is likely due to the fact that the domains of cost and coverage objectives are of larger magnitude than that of the delta-v metric. Regardless, we can then look at the distribution of observer families within each cluster to better understand what observer combinations tend to result in certain CA delta-v's and coverage metrics. For instance, we can compare the orbits within cluster 3, which correspond to high architecture cost, high coverage, and medium delta-v's, to those found in cluster 0, which contain low cost, low coverage, and medium delta-v's. Figure 4.17 shows a histogram of observer orbit families found in cluster 3 and 0. As we might expect, there are clear differences in the types of observer orbits utilized between these two clusters. While there are some overlaps between them, cluster 3 is dominated by architectures with GEO, L2 Halo, N. Dragonfly, and L4 Axial observer orbits. This is similar to those architectures found in cluster 0 from Table 4.3, many of which were found to be Pareto-optimal. On the other hand, in cluster 0, we see in Figure 4.17 that it tends to contain architectures with L5 Planar, N. Butterfly, and 1:2 Res. observer orbits, again similar to cluster 2 in Table 4.3.



(a) Plot of objective space with points colored by cluster from K-Means. Note that the direction of the cost axis is flipped from Fig. 4.10a to aid in cluster visualization.



(b) Projections of objective space with points colored by cluster. Convex hulls outline distinct clusters.

Figure 4.16

These two clusters represent distinct regions within the tradespace of objectives, indicating that architectures with GEO, L2 Halo, N. Dragonfly, and L4 Axial observers may have more desirable characteristics for optimizing our control weighted coverage, while maintaining a reasonable delta-v for the CA. However, we must also consider that the differences in coverage between these two clusters could be heavily influenced by the increased cost of architectures in cluster 3 as compared to cluster 0. But, clusters 5 and 8 (top left in coverage-cost projection in Figure 4.16b), are

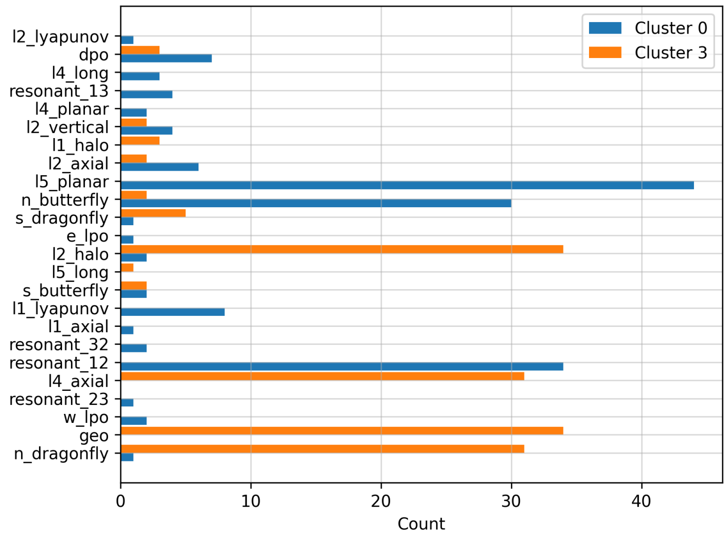


Figure 4.17: Counts observer orbit families in clusters 0 and 3 from K-Means.

also overwhelmingly defined by similar architectures (GEO, L2 Halo, N. Dragonfly, and L4 Axial observers) with lower cost telescope diameters.

If we are concerned primarily with a coverage minimum of, say, 80% in our Cislunar architectures, we may solely select architectures from clusters 3, 5, and 8, again most often consisting of observers in GEO, L2 Halo, N. Dragonfly, and L4 Axial orbits. In these clusters, the CA is able to find many persistent detection corridors over the 5 simulation epochs defined by a minimum detection of 80%, with cost distributed between CA delta-v and architecture cost. There are innumerable potential reasons for the increased performance of these architectures. First, in any combination of these four observer orbits, at least one observer exhibits significant motion along the z-axis (assuming there are not only two GEO observers). In the case of a planar CA transfer trajectory, this allows for more potential observations at moments when planar orbits may not be able to view certain regions due to solar pointing constraints. This phenomenon can also be seen via the lack of planar observers in architectures in the Pareto front, as shown in Figure 4.12 (except for GEO observers). Second, looking again at cluster 0 in 4.16b, the very long periods of 1:2 Resonant orbits make it so that observations from an observer repeat with lower frequency, meaning that other observers would be needed to fill in the gaps, increasing architecture cost or

CA delta-v to increase coverage.

4.3.4 Discussion

The results from both clustering runs indicate a couple of important things. First, certain combinations of observer orbits may perform better with other specific orbits - no architectures were found to be Pareto-optimal with more than two observers in the same orbital family, although by no means was the state space fully exhausted during optimization. Second, the initial phasing of observers in an architecture must be carefully selected to optimize performance, and merely adding a new observer in an existing optimal architecture does not guarantee continued optimality without potentially adjusting the phasings of all observers. Third, architectures with multiple observers that have out of plane motion exhibit exceptional performance with respect to coverage, and GEO observers are crucial for maximizing control weighted coverage for CAs departing from GEO. Finally, we have shown that clustering methods can be utilized as a robust tool for analyzing the design and objective spaces of the Cislunar SDA architecture design problem. Through grouping architectures together by subsets of these spaces, we can develop an intuitive understanding of the relationships between observer states and capabilities within architectures, and why certain architectures may perform better with respect to our novel problem objectives.

Chapter 5

Rapid Uncovering of Cislunar Persistent Detection Corridors for Crewed Missions

5.1 Persistent Detection Corridors

Ensuring that a critical asset, such as a crewed spacecraft, is detectable by an SSA architecture during its entire traversal through Cislunar space is paramount to the success and safety of such missions and to the Cislunar regime as a whole. While good track maintenance can be sufficient in cases of detection gaps for un-crewed or lower value assets, an unexpected emergency during flight can put human lives in danger and jeopardize mission success. As such, ensuring that detection exists of said assets within the field of regard of an architecture must be considered as a crucial aspect of both mission and architecture design.

In the previous contribution, the CA model generated transfers that maximized detectability by the architecture, and as such attempted to uncover so-called *persistent detection corridors*, volumes of Cislunar space defined over time along feasible trajectories corresponding to near continuous detection by said architectures, visualized in Fig. 4.2. However, this method relied on a higher level representation of detection through apparent magnitude rather than a boolean value of detectability based on a minimum photometric SNR value. As such, optimal trajectories returned by the CA were unable to directly interact with the distribution of boolean detectable regions resulting from the architecture layouts. In this work, we aim to create a discrete representation of these regions within which mission designers can explore the state space of feasible trajectories, knowing that minimum levels of persistent detection of their assets can be guaranteed.

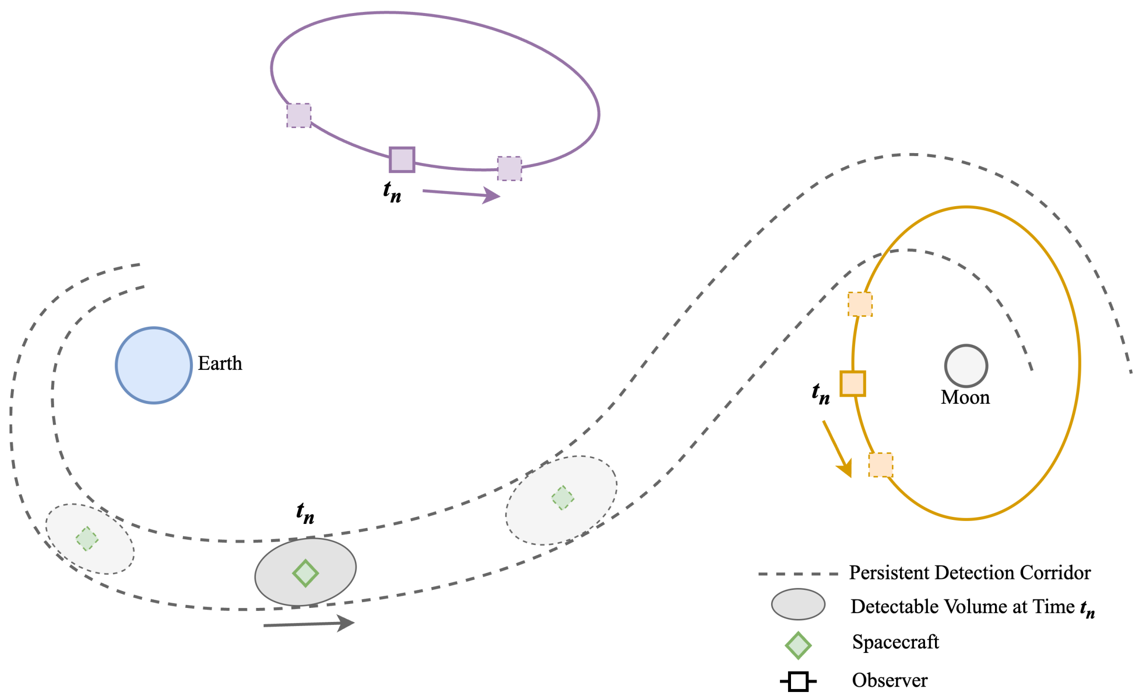


Figure 5.1: Visualization of a persistent detection corridor for an arbitrary Cislunar SSA architecture.

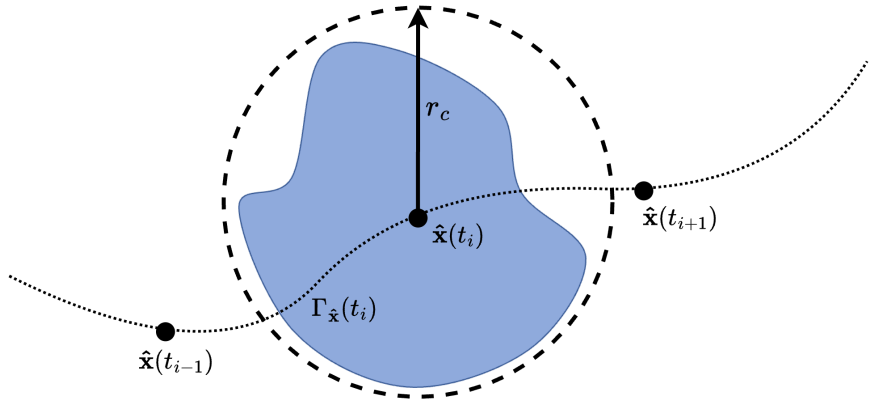


Figure 5.2: Diagram of a local neighborhood of detectability t_i. The shaded region represents the approximate region of continuous detectability about the reference trajectory's position.

These detection corridors exist as a function of observer positions, capabilities, and the incoming solar illumination in the Earth-Moon synodic frame. A corridor may initially be defined along a feasible reference trajectory within the relevant dynamical system (here, the CR3BP) that traverses primarily through the detectable regions of position space as determined by the individual observers constituting the architecture. For ease of discussion and visualization, we will be restricting these corridors to be defined in two-dimensional position space using the planar CR3BP. At each discrete time step t_i along the reference trajectory, there may exist a region of near continuous detectable space provided by the architecture. This region of continuous detectable space represents the local neighborhood of detectability (LNoDe) about the reference trajectory state $\hat{\mathbf{x}}(t)$ at time t_i. Deviation in position space from the reference trajectory that lies within the LNoDe remain detectable, with the assumption that small course corrections are capable of realigning with the reference trajectory.

We denote the LNoDe at time t_i as $\Gamma_{\hat{\mathbf{x}}}(t_i)$, representing the region of position space within a corridor radius r_c about the reference bounded by maximum local detectability. Fig. **FIGURE** provides a visualization of an LNoDe. A *detection corridor* can then be defined as the set of LNoDes evaluated along the discrete time steps of the reference trajectory:

$$\mathbf{C}(\hat{\mathbf{x}}) = \{\Gamma_{\hat{\mathbf{x}}}(t_i) : \ \ t_i \in \{t_0, ..., t_f\}\} \tag{5.1}$$

The idea of *persistence* can be defined in two ways. First, we may consider persistence as a measure of the total area or volume (in the planar or non-planar CR3BP, respectively) enclosed by an LNoDe divided by the area or volume of the circle or sphere defined by the corridor radius r_c. We may refer to this concept as the LNoDe persistence, describing the general coverage of the region near the reference trajectory at a given time step. Second, we can consider both the total average LNoDe persistence over the entire PDC, and the occurence and duration of significant loss in LNoDe persistence, encapsulated by taking the standard deviation of the LNoDe persistence metric. We refer to these quantities as the *corridor coverage* and *corridor variation*, respectively. Given user defined minimums for LNoDe persistence at specified time steps and corridor coverage and variation, detection corridors that meet such requirements are referred to as persistent detection corridors (PDCs). Levels of persistence will inevitably vary across different mission types.

Unique persistent detection corridors exist per-architecture, and may vary significantly in nature depending on architecture attributes. But the enumeration of PDCs for potential Cislunar SSA architectures will allow for mission designers to rapidly generate transfers based on the geometries and SSA qualities of said corridors. Critically, in missions where detectability of assets is crucial at all times, PDCs provide a foundation for mission design that is centered around human safety.

5.2 Sample-Based Trajectory Planning

Sample-based trajectory planning, within the broader field of kinodynamic planning, refers to the process of sampling states and control inputs across a region through which to recover a trajectory, while adhering to dynamical and/or environmental constraints [33, 68]. Often referred to in the context of robotic motion planning, methods from this field are used in many applications related to astrodynamics, often for initial mission design for transfers in multi-body systems [91, 20, 30]. These methods typically involve sampling states within a region of interest such that the underlying system dynamics can be reasonably approximated. The resultant states are represented within a graph, with nodes representing states and edges representing feasible motion between states.

Bruchko & Bosanac [21, 22] introduced a foundation for rapid trajectory design within multi-body systems such as the CR3BP. In their work, the authors outline a method of environment approximation where full states, referred to as centroids, in the CR3BP are sampled with velocities distributed according to a given Jacobi energy. Centroids are propagated forwards and backwards in time for a given arclength or propagation time. The centroids and the final states of propagation are added as nodes to a graph, with edges connecting the nodes' states, capturing natural motion in the system about each centroid. Afterwards, edges are constructed between nodes across centroids such that the distance between node positions and the angle between node velocity vectors are minimized. A shooting algorithm with control is then employed to ensure continuity in position space between nodes, obtaining a delta-v that is added to the edge as a weight. A path finding algorithm is then used to recover paths through the graph that minimize total delta-v of the transfer. Finally, the nodes along recovered paths are used as an initial guess for a shooting algorithm that recovers a transfer within the system.

It should be noted that most of the work in kinodynamic motion planning is intended for the autonomous and optimal motion of robotic systems, where collision avoidance of environmental obstacles is often a priority, and dynamical constraints are typically linear or linearized nonlinear [81, 33, 100]. In highly nonlinear systems, common methods for kinodynamic planning such as rapidly-exploring random trees (RRT) and their variants [69, 100] suffer from solution degradation and instability due to linearization. Within the Cislunar SSA environment, the combination of the CR3BP, complex orbital geometry, and the varying solar illumination lead to a highly nonlinear dynamical system with no analytical solutions. As such, methods based on motion primatives such as Bruchko & Bosanac's and many others [73] can be described as a "search then optimize" approach, where the search is performed in a descritized state space, and discontinuities in the dynamical systems are smoothed out through trajectory optimization. Furthermore, in our application, we are not necessarily concerned with collision avoidance. Rather, we are solely interested in recovering trajectories that traverse through detectable regions as defined by the architecture.

In this work, we implement the graph-based method from Bruchko & Bosanac with modifications to state sampling, cost of traversal, and trajectory recovery that are informed by Cislunar SSA domain knowledge. Rather than enforcing the nonlinear constraint of detectability during the path search, we instead encode detectability directly in the graph structure while simulateously approximating adherence to the CR3BP through motion primatives. This allows us to recover near-feasible paths through the graph that can then be used as an initial guess for a trajectory optimization problem, recovering a feasible trajectory satisfying the CR3BP and the detectability constraints.

5.3 Graph Representation of Detectable Regions Over Time

5.3.1 Centroids and Local Neighborhoods

We begin graph construction by sampling the relevant state space in our problem, that is, the regions of detectability by an arbitrary architecture. Within an architecture simulation, we can identify a time span starting from an epoch of interest for analysis and iterate through time steps, identifying detectable points. Inspired by Bruchko & Bosanac [22, 21], for each time step we select a number of detectable points as centroids. It is of interest that these initial centroids are selected so that they sufficiently represent the distribution of detectable regions in the state space. To ensure accurate spatial representation, we use `sklearn`'s `KMeans` module [80] to cluster detectable points into n_c groups, then selecting the cluster centers as centroids to add to the graph for each time step in the simulation.

A centroid selected from the grid in our problem is defined in position space only, so velocities must be sampled intelligently so that the dynamics of the local region about that centroid are sufficiently captured. First, at each centroid, n_v unit vectors are evenly distributed about the position

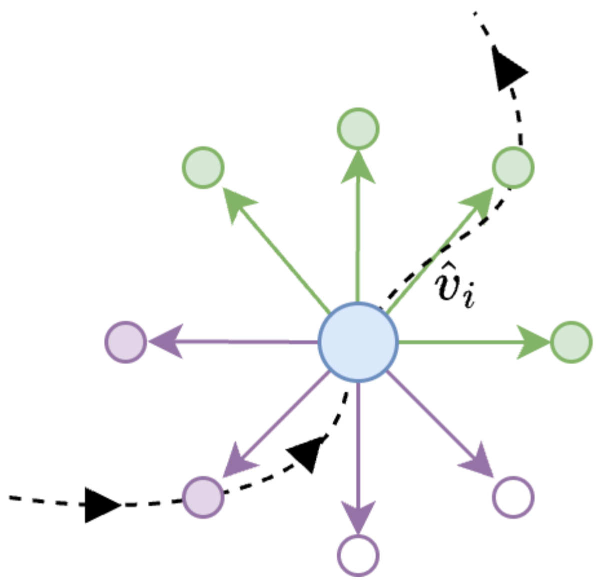


Figure 5.3: A single centroid (blue) with velocity directions propagated forward (green) and backward (purple). Shaded circles indicate detectability. Figure adopted from [22].

representing velocity directions. Then, n_v unique velocity magnitudes are uniformly sampled from a desired range of Jacobi energy at the centroid position. This range of Jacobi energy may be selected according to desirable type of motion when considering the zero velocity surfaces of the corresponding energy levels.

After velocities are sampled, we end up with n_v states represented per identified centroid position. Then, each unique state (containing velocities) is propagated forwards and backwards for a given time or arclength in position space, and the final states of propagation are checked for detectability at their respective times. Forward and backwards states that are detectable by the architecture and their corresponding centroid full state are finally added as nodes to the graph. Nodes η are defined as planar states with time of detection appended: $\eta = [x, y, \dot{x}, \dot{y}, t_i] \in \mathbb{R}^5$. Then, centroid nodes are added as (forward) neighbors to their respective backward nodes, and forward nodes are similarly added as neighbors to their respective centroid nodes. Since forward and backward nodes are the result of propagation, traversal from backward to centroid nodes and centroid to forward nodes represents natural motion in the system, thus the default cost to traverse between nodes in this manner is 0.

The collection of nodes defined from a centroid position along with the corresponding detectable forward and backward nodes makes up the local neighborhood of the centroid at the centroid time step. A local neighborhood consists of up to $3n_v$ nodes Fig. 5.3 shows an example of a centroid position and its corresponding forward and backward propagated nodes. In this figure, traversal through the neighborhood starting from a backward node to its centroid and finally through the corresponding forward node represents natural motion in the system dynamics.

The identification of centroids and construction of local neighborhoods occurs at each time step during the time span of interest, resulting in up to $N \cdot 3n_v \cdot n_c$ nodes in the graph for N time steps. The next step involves creating a method by which nodes across disjoint local neighborhoods can be connected as neighbors forward in time.

5.3.2 Node Neighbor Querying via KDTrees

Our graph now consists of nodes with neighbors representing natural motion in the local neighborhood of detectable grid points in the simulation. To find paths through our graph, we need a way to be able to connect nodes between local neighborhoods. Rather than pre-computing edges between nodes in different local neighborhoods, we implement a method by which a node's neighbors are computed on the fly during path searching. To accomplish this, we use `SciPy`'s `cKDTree` [96] method, which enables rapid lookup of the nearest neighbor of any point.

To allow for more control over neighbor selection, two KDTrees are instantiated for each local neighborhood, a *centroid* KDTree and a *forward* KDTree. Candidate neighbor nodes for each tree are filtered by maximum displacement and time, as well as the type of node (backward, centroid, or forward). To encourage paths that traverse through the natural motion of local neighborhoods, backward nodes only have their respective centroid node as a neighbor, which is guaranteed to exist. Furthermore, the following rules are enforced:

- Neighbors can only exist forward in time.
- Besides those from initial construction, nodes cannot have neighbors in the same local neighborhood.
- Nodes cannot have neighbors that occupy the same position.
- Maximum position and time displacement of Δp and ΔT between neighbors.
- Centroid and forward nodes both can only have neighbors of backward and centroid nodes.

So, to construct each KDTree, all graph nodes satisfying these rules are used as the data from which to query, normalized by Δp and ΔT. Then, during the path search, the relevant KDTree (centroid or forward) of a node's local neighborhood is used to query the node for k neighbors. If the node is a backward node, then its corresponding centroid node is returned as noted previously. It is important to clarify that while we are not directly constructing and keeping track of edges, they are implicitly defined through this process of neighbor querying. Using this method significantly reduces the computational cost of graph construction, ensuring that neighbors are only added to the graph when they are needed during path finding.

5.3.3 Cost of Neighbor Traversal

During the path finding phase, traversal between nodes in different local neighborhoods will not necessarily be dynamically feasible. As such, we seek to find paths through our graph that are near feasible by encoding a measure of dynamic feasibility in the cost of traversal. Bruchko & Bosanac implement a weighted sum of Euclidean distance and angular difference between node velocity vectors to aid in this process, followed by a shooting algorithm generating an estimate of the cost of traversing between nodes. Here, we implement a novel distance metric inspired by methods in direct collocation [59] by using a weighted sum of trapezoidal transcription error, Euclidean distance, and velocity angular deviation. The trapezoidal transcription error is derived from approximating a system's dynamics using trapezoidal quadrature, a method for approximating a definite integral in numerical integration. Within direct collocation, constraints are enforced on segments between sampled collocation points that enforce adherence to a system's dynamics. The trapezoidal transcription error $\delta_{k,k+1}$ can be expressed as

$$\begin{aligned} \delta_{k,k+1} = (\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) - \frac{1}{2}h_k\left[\boldsymbol{f}(\boldsymbol{x}_{k+1}) + \boldsymbol{f}(\boldsymbol{x}_k)\right] \\ h_k = (t_{k+1} - t_k) \end{aligned} \tag{5.2}$$

where $\boldsymbol{f}(\cdot)$ is the system dynamics and $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k+1}$ are collocation points between the time steps of t_k and t_{k+1}, respectively. Minimizing this transcription error indicates greater adherence to the system dynamics. The total cost of traversal between nodes is then

$$g_{i,j} = \frac{\|\delta_{i,j}\|_2}{\delta_{max}} + \frac{d_{i,j}}{d_{max}} + \frac{\Delta t_{i,j}}{\Delta T} \tag{5.3}$$

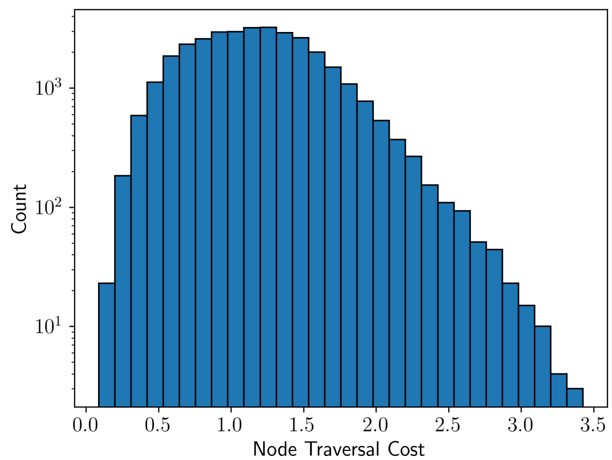


Figure 5.4: Example distribution of traversal costs in detection graphs.

where $d_{i,j}$ is the Euclidean distance between nodes i and j normalized by a the maximum neighbor distance d_{max}, and $\Delta t_{i,j}$ is the time between nodes i and j normalized by the maximum neighbor time ΔT. An example distribution of node traversal costs is shown in Figure 5.4.

5.3.4 Path Finding and Refinement

5.3.4.1 Path Finding and Refinement

With the graph constructed, we can apply a path finding algorithm to recover paths through the graph, minimizing the cost as defined in Eqn. (5.3). Here, a slightly modified version of A* [53] is used where we minimize the function $f(\eta) = \hat{g}(\eta) + h(\eta)$, where $\hat{g}(\eta)$ is the *average* cost of traversing between nodes from the start node to η. Using an average cost encourages A* to potentially return longer paths as long as they are most dynamically consistent. Furthermore, to determine whether to terminate the path search we implement a *vicinity* approach, wherein during each traversal of a new neighbor, we check if the current node is within the vicinity of a predetermined goal position and time. This allows for a greater variety of paths to be returned from the A* search and generally prevents A* from not converging on a path.

Paths recovered from A* within the graph represent approximate feasible trajectories within detectable regions of Cislunar space. Something to note regarding these paths is that depending on the distribution of nodes within the Cislunar volume, the initial and final states of the initial path may not lie close to the reletive states of interest, such as the Earth and Moon for an Earth-Moon transfer. As such, if a path's start or goal node is not within some distance of the Earth or Moon, respectively, then a node is appended to the path to ensure that the resultant path is representative of an Earth-Moon transfer. Because of the discretization of the environment, the resultant paths returned by A* do not exactly represent continuous natural motion within the system. Thus, for the resultant paths to be useful in the context of mission design, we need to find a true feasible trajectory that minimizes the distance to the path nodes. To perform this, we implement methods from collocation theory and perform a non-linear least squares optimization to recover natural a trajectory minimizing the distance to any given initial path.

Typically when performing collocation, a trajectory is sampled at specific collocation points distributed to optimize numerical stability when evaluated interpolated derivatives. These collocation points are often sampled at the roots of a high-order orthogonal polynomial, such as the Legendre polynomial. This allows for exponential convergence rates when the entire trajectory is sampled along these roots, referred to as *pseudospectral collocation* [59]. Fig. 5.5 displays a selection of Legendre polynomial roots, obtained from [42]. Since the nodes in the paths recovered from our graph are not distributed in time according to these roots, we first must interpolate these paths at the roots of the Legendre polynomial. With a properly distributed path, we can formulate the non-linear optimization.

As previously stated, given an initial set of nodes along a path representing points in Cislunar space detectable by an architecture, we wish to recover a natural trajectory within the CR3BP that minimizes the total distance from said nodes. Given an initial path $P = \{\eta_0, \eta_1, ..., \eta_{f-1}, \eta f\}$ represented by a sequence of graph nodes connected via directed edges, we interpolate this path resulting in sequence of collocation points $X_C = \{\boldsymbol{x}(r) : r \in \{r_0, ..., r_n\}\}$ where $\{r_0, ..., r_n\}$ are the roots of a Legendre polynomial. Then, we can construct the optimization problem as

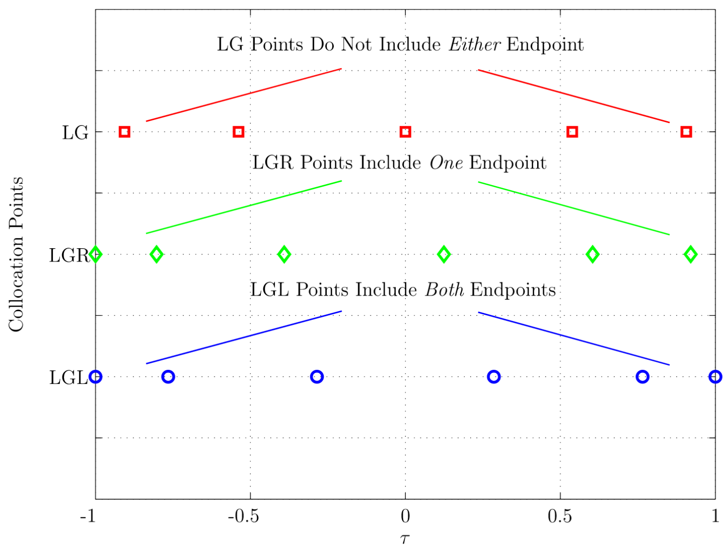


Figure 5.5: Distribution of different types of Legendre-Gauss collocation points. Obtained from [42].

$$\min_{X,U} \left[\alpha(X - X_C)^T(X - X_C) + \lambda U^T U\right] \tag{5.4}$$

$$s.t. \quad DX - \boldsymbol{f}(X, U) = 0$$

where $U \in \mathcal{R}^{n\times 2}$ is a vector containing the continuous time control vector at each collocation point, $\boldsymbol{f}(X, U)$ are the CR3BP dynamics evaluated at the collocation points with control and $DX = \dot{X}$ are the derivatives of the interpolated collocation points. The constraint in Eqn. (5.4) is referred to as the *pseduospectral transcription error* and similar to the trapezoidal transcription error in Eqn. (5.2), it enforces adherence to the system dynamics at the collocation points. As previously stated, with collocation points evaluated at the roots of the Legendre polynomial, this constraint guarantees convergence [59]. The solving of Eqn. (5.4) is performed with `scipy`'s `minimize` function using the SLSQP algorithm [96]. In Eqn. (5.4), the weight α on the position constraint can be set to 0 so that the problem is merely a minimization of control effort, with the bounds in `minimize` on the state vector X instead enforcing proximity to the original path obtained from A*. Once converged, the resultant solution X^* is interpolated at the original simulation path times $t \in t_0, ..., t_f$ and represents a feasible trajectory within the dynamical system.

5.3.5 Detection Corridor Generation

With a feasible reference trajectory recovered from a path traversing through detectable regions of Cislunar space, we then construct local neighborhoods of detectability at each time step along the trajectory. The LNoDes along the trajectory approximate the maximum continuous region of detectability within the local region of the trajectory at each time step. To calculate this, we sample the local region of the reference trajectory at time t_i using polar coordinates centered at $\hat{x}(t_i)$ with $\theta_j \in [0, 2\pi)$ and $r_l \in (0, r_c]$. For each θ_j, the detectability of each point at (r_l, θ_j) is determined, and the boundary at θ_j is determined as the maximum value of r_l such that (r_l, θ_j) is detectable and r_{l+1} is not detectable by the architecture. Fig. 5.6 visualizes this method for determining the boundary points of an LNoDe at time t_i. As discussed previously, the set of

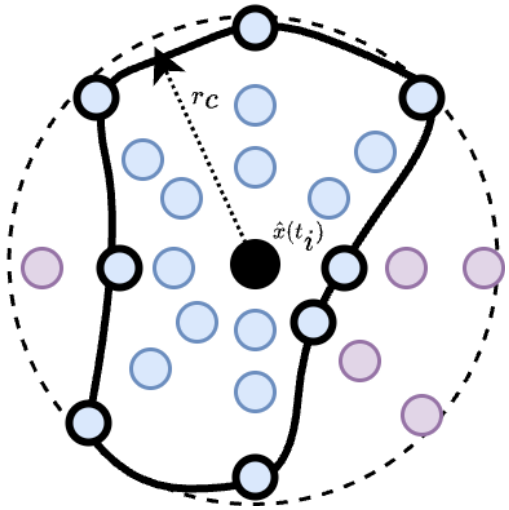


Figure 5.6: Polar coordinate sampling about reference trajectory with LNoDe boundary points estimated. Detectable points colored blue with purple points undetectable. Boundary points are outlined in black.

LNoDes about reference trajectory constitutes its detection corridor, representing the detectability of the trajectory and its local neighborhood in position space over time.

5.3.5.1 Persistence Metrics

Two main intuitive metrics are considered for determining persistence within a given detection corridor, along with the total coverage obtained of the target along the reference trajectory. First, the area enclosed by the boundary points of an LNoDe represents the local region within which an asset could experience perturbations in position space while remaining detectable by an architecture. The corridor width r_c at an LNoDe can be selected heuristically such that it bounds the reachable set in position space resulting from an unexpected perturbation. Thus, the persistence of an LNoDe can be defined as the total area bounded by its boundary points divided by the total area of the corridor with radius r_c:

$$\rho_{\Gamma_{\hat{\mathbf{x}}}(t_i)} = \frac{A_{\Gamma_{\hat{\mathbf{x}}}(t_i)}}{A_{r_c}} \tag{5.5}$$

For this work, as a proof of concept r_c is chosen to be a constant value not necessarily representing the reachable set about the reference trajectory.

Next, the total persistence of an entire detection corridor $\boldsymbol{C}(\hat{\boldsymbol{x}})$ can be evaluated by obtaining statistics of its LNoDes at each time step of the reference trajectory. First, the total corridor coverage can be calculated as

$$\rho_{\boldsymbol{C}(\hat{\boldsymbol{x}})} = \frac{1}{N} \sum_{t_i}^{t_f} \rho_{\Gamma_{\hat{\mathbf{x}}}(t_i)} \tag{5.6}$$

for N time trajectory time steps. This metric represents the average LNoDe persistence over the duration of the reference trajectory. Said differently, this metric captures total detectability about the reference trajectory within the defined corridor radius r_c. If r_c is determined via an approximation of the reachable set about the reference, then this metric represents the coverage of a target if it is perturbed at any time step. This provides insight to mission designers of the expected coverage of an asset if the reference trajectory is not perfectly matched. A more in depth look into detection gaps and variation across LNoDes benefits from a visual analysis of persistence metrics over a trajectory duration.

These metrics can provide initial insight as the to performance of a corridor, and by specifying minimum values of persistence matching relevant mission objectives, mission planners can obtain *persistent* detection corridors that guarantee persistent coverage of their assets. Many detection corridors are returned with our methodology, so filtering by persistence metrics can ensure that only PDCs of interest can be evaluated on a lower level by mission planners.

5.4 Results and Discussion

5.4.1 Cislunar Architecture

As found in the previous contributions, there exists a complex tradespace between objectives in the Cislunar SSA architecture design problem. Notably, increases in coverage provided to volumes of Cislunar space and basic transfer trajectories comes at the expense of either a high cost architecture or a large control effort by a cooperative agent. In this work, we seek to un-

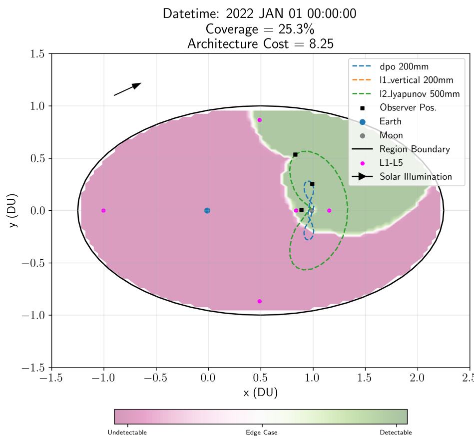
Table 5.1: Selected architectures from [63].

Architecture	Observer Orbit	Initial Phasing	Telescope Diameter	Period (TU)	Jacobi Energy
1	Distant Prograde	0.7	200mm	5.066	3.027
	L1 Vertical	0.7	200mm	5.527	2.772
	L2 Lyapunov	0.8	500mm	6.678	2.926
2	W. Low Prograde	0.1	300mm	5.066	3.186
3	E. Low Prograde	0.6	500mm	5.297	3.169
	GEO	0.6	500mm	4.605	N/A
	L2 Lyapunov	0.4	200mm	5.066	2.977
	L2 Vertical	0.0	300mm	5.066	2.884

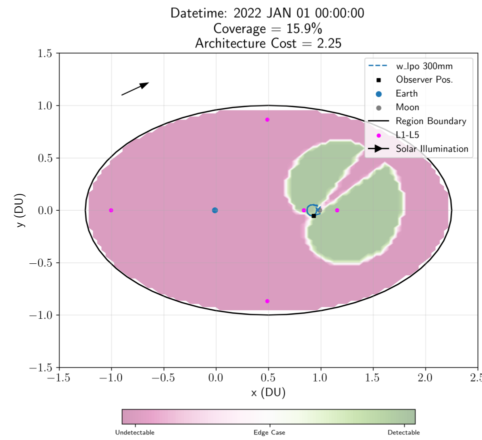
cover persistent detection corridors defined by reference trajectories minimizing control effort while maintaining detection within said corridors. Intuitively, we may expect that architectures with high average coverage of the Cislunar volume would result in more detection corridors with persistent detection. Depending on the distribution of detectable regions we may find decent performance in lower cost, lower volume coverage architectures as well. As such, we examine resultant PDCs across three Pareto-optimal architectures from [63] with varying levels of coverage and cost. These architectures are described in Table 5.1 and shown in Figure 5.7.

5.4.2 Graph Construction

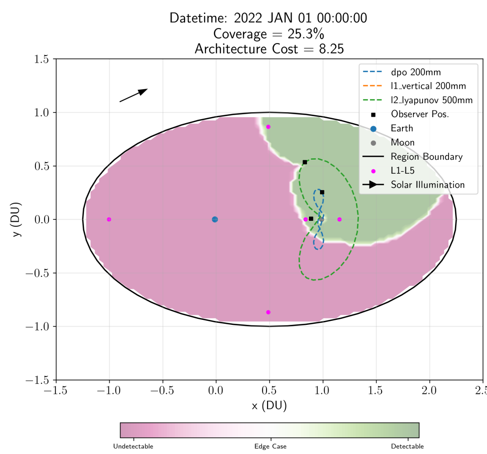
For each architecture, a graph is constructed following the outlined methods above. The initial simulation epoch is set to 01 Jan 2022 00:00:00 GMT, at which point initial observer phasings are set. Graph construction begins by evaluating the detectability of evenly distributed grid points in the $x-y$ plane, over a time period of 60 days with a time step of 24 hours. At each time step, detectable points are sampled via KMeans to become centroids, from which local neighborhoods are generated by sampling velocities within the range of Jacobi energies of $[3.01377, 3.173]$, corresponding to zero velocity curves allowing traversal to the Moon and the L1 and L2 Lagrange points. Then, each unique centroid state is propagated forward and backwards in time for a duration sampled uniformly within the interval $[12, 36]$hrs. This allows for a finer representation of an architecture's detectability between the simulation's time step. The detectability of propagated states by the architecture is evaluated and detectable states are added as nodes. In Fig. 5.8, centroids and



(a) Architecture 1 - DPO with 200mm telescope, L1 Vertical with 200mm telescope, and L2 Lyapunov with 500mm telescope.



(b) Architecture 2 - W. LPO with 300mm telescope.



(c) Architecture 3 - E. LPO with 500mm telescope, GEO with 500mm telescope, L2 Lyapunov with 200mm telescope, and L2 Vertical with 300mm telescope.

Figure 5.7: Variation in detectable regions for three selected architectures. Shown at a single epoch in the simulation history. Artifacts and regions labeled as edge cases are a result of the discretization and interpolation of the sampled grid points.

propagated nodes added to the graph from architecture 1 are shown for three time steps within the simulation. After node construction, KDTrees are created for each centroid neighborhood for the querying of neighbors. The resultant graphs for each architecture are shown in Figure 5.9.

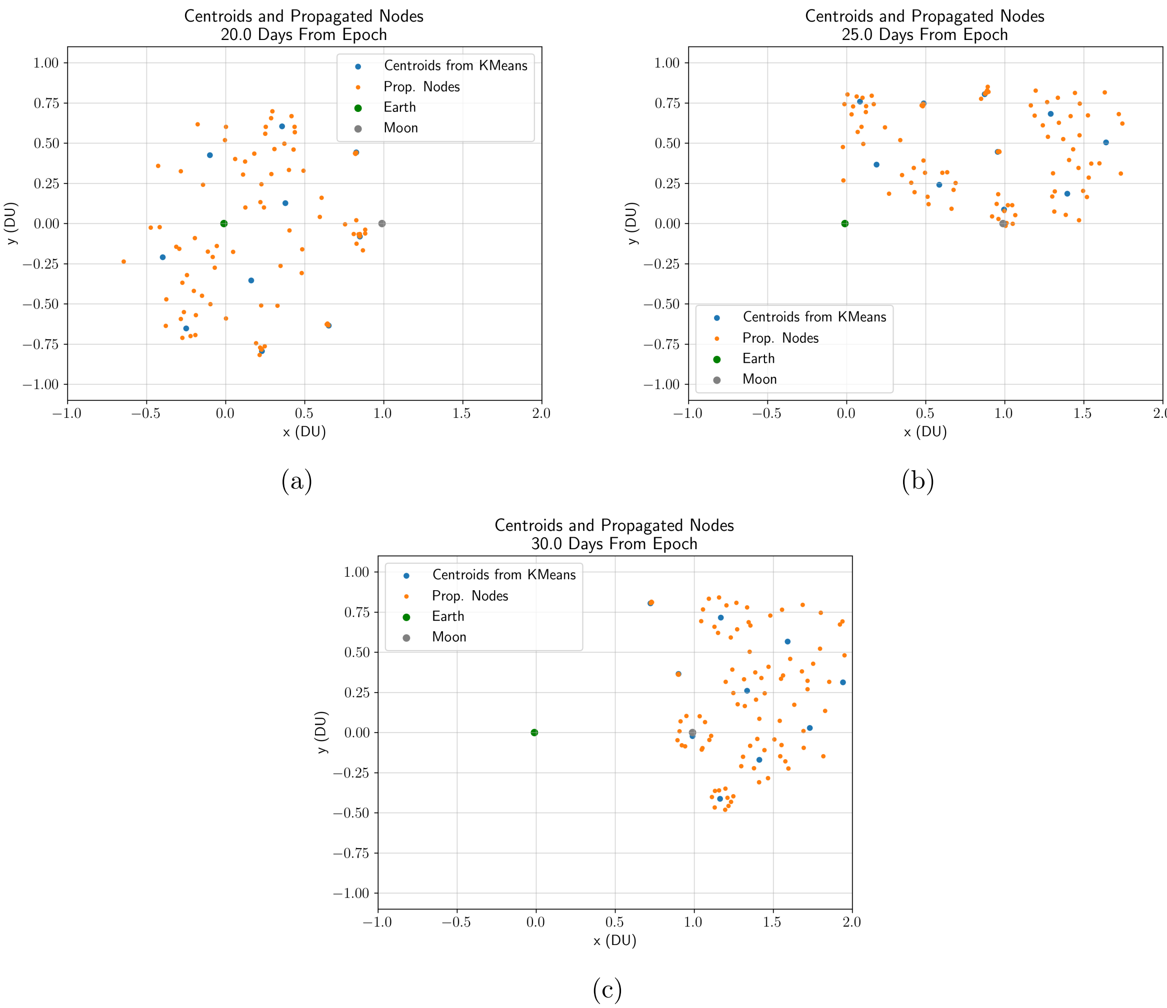


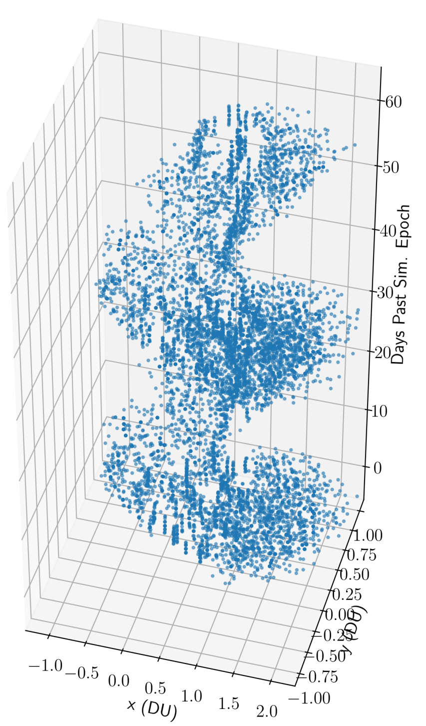
Figure 5.8: Centroids and neighborhood nodes added to the graph at distinct time steps for architecture 1.

The detection graphs in Fig. 5.9 help to elucidate the underlying concept we are presenting in this work. In all graphs, it is clear that the regions of detectable space vary significantly over this 60 day time span between the architectures. Upon initial inspection, one can imagine that a spacecraft could navigate through these continuously detectable regions. The distribution of nodes in these graphs also provide insight into the varying performance of the selected architectures. In all cases, we see clearly the effects of the changing angle of solar illumination, and how it interacts with the moving observers in their respective periodic orbits producing a sort of spiral of detectable points over time. In particular, the graph of architecture 2 in Fig. 5.9 shows this spiral only within the vicinity of the Moon.

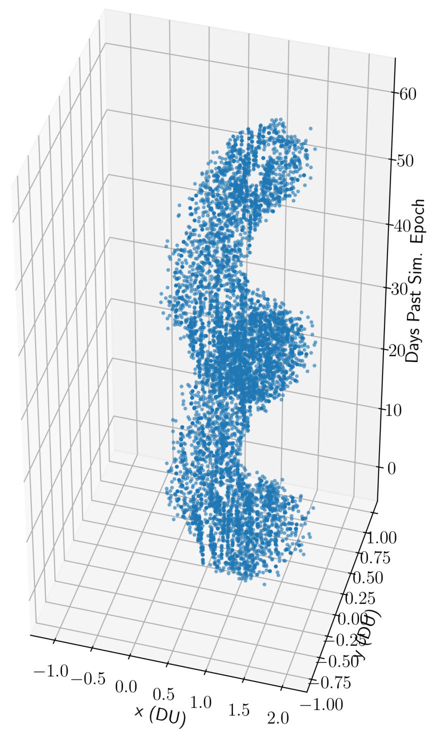
5.4.3 Paths

With our detection graphs constructed shown in Fig. 5.9, the A* path finding algorithm with cost and heuristic functions as previously discussed is applied to uncover near feasible trajectories. To find paths representative of crewed missions from the Earth to the Moon, we identify a number of initial nodes that minimize distance to the Earth respectively at unique time steps throughout the 60 day time span. We iterate over each initial node to find paths that reach the vicinity of the Moon, within a tolerance in distance and time. For each architecture graph, this results in a variety of paths representing transfers of varying durations, control effort, as well as paths with varying departure and arrival dates. The selection of initial and goal nodes can be modified to account for specific mission requirements if so desired.

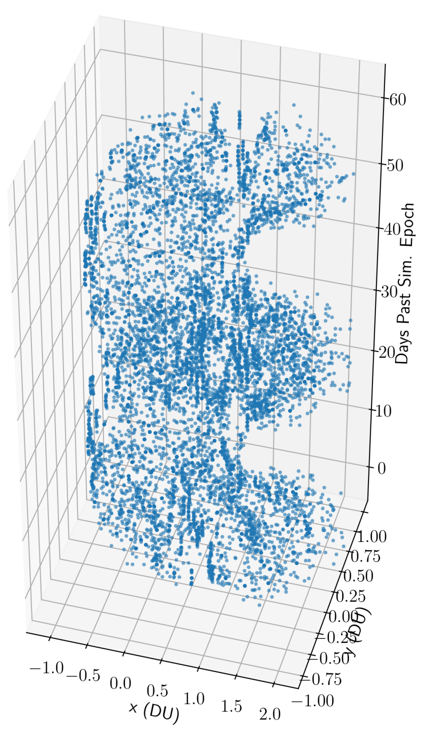
Figure 5.10 displays a number of paths for each architecture found after iterating through 30 unique times in the 60 day time span, at each time identifying three unique candidate start nodes. Recall that these paths traverse through detectable regions of Cislunar space resulting from our graph construction process. An initial indication of persistence can be evaluated by identifying paths with short arcs between bends (representing nodes), representing paths whose traversals through detectable regions is likely near constant. Longer arcs between bends in a path may indicate that in order to reach the goal, the path had to traverse across a larger distance or



(a) Detection graph from architecture 1.

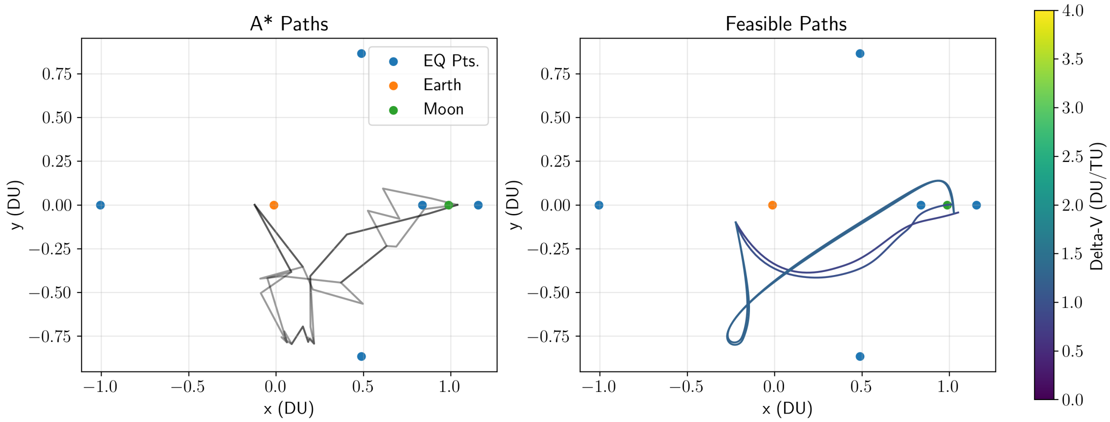


(b) Detection graph from architecture 2.

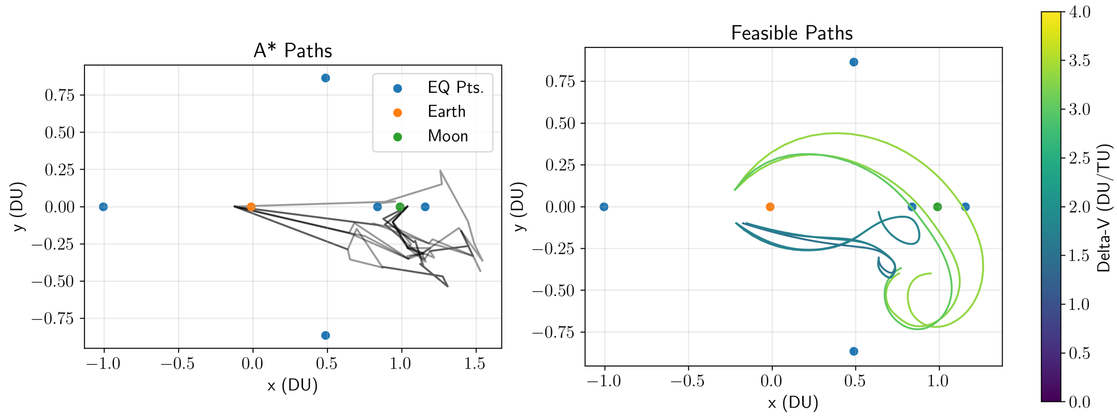


(c) Detection graph from architecture 3.

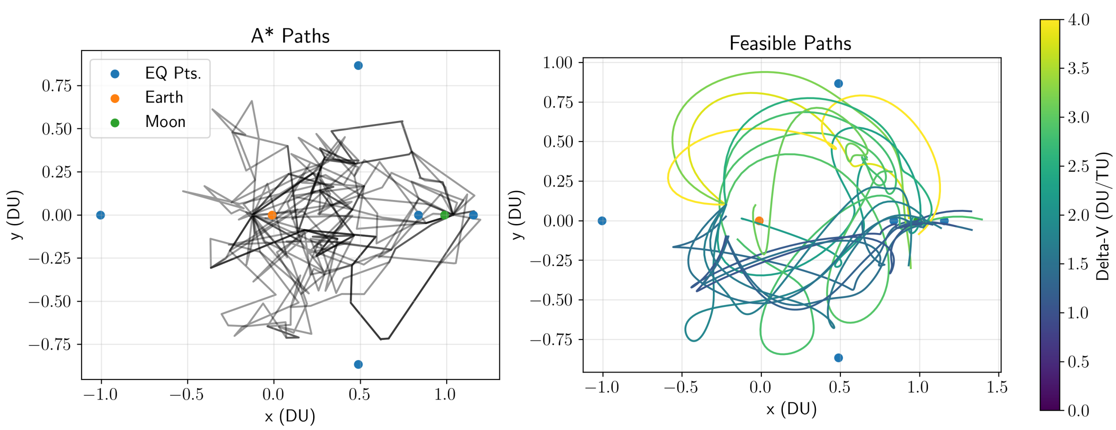
Figure 5.9: Detection graphs from the selected architectures.



(a) Resultant paths and reference trajectories from architecture 1. Only those with corridor persistence above 90% shown.



(b) Resultant paths and reference trajectories from architecture 2. Only those with corridor persistence above 80% shown (none over 90% were returned).



(c) Resultant paths and reference trajectories from architecture 3. Only those with corridor persistence above 90% shown.

Figure 5.10: Paths and feasible trajectories from detection graphs in Fig. 5.9. The feasible reference trajectories are colored by the total control effort used.

period of time, during which detection may be ambiguous. As such, Fig. 5.10 only shows paths that resulted in corridor persistence over some minimum amount, here selected to be 90 %. Architecture 2 in Fig. 5.10b did not have any paths with coverage greater than 90%, so those with coverage over 80% are shown instead.

5.4.4 Corridors

For the time span of 60 days we are able to rapidly generate a number of detection corridors for each architecture at varying epochs. For illustration, we select a single persistent detection corridor from architecture 3, then present a collection of them for each architecture. Figure 5.11 visualizes a persistent detection corridor with high corridor coverage about the reference trajectory (projected onto the $x-y$ axis). For the detection corridors shown here, we select an LNoDe radius of $0.1DU$ within which we determine the local detectability. This radius can be modified as needed to closer match mission specifications. For example, the corridor radius could be chosen from mission requirements specifying guaranteed coverage within any distance from the reference trajectory. In the case of human spaceflight, one can imagine a scenario in which there might be an issue with a maneuver deviating the spacecraft from its nominal trajectory. Mission planners may want to have knowledge of the local neighborhood of detection around such maneuvers to act as a backstop for potential loss of custody.

Perhaps more revealing, we can plot multiple detection corridors in the same plot, which further elucidates the underlying concept of this work. For each candidate corridor epoch, we select the corridor with the highest coverage over 50% and plot them together. Figures 5.12 - 5.14 show a selection of detection corridors over the 60 day time span for each architecture. These plots display trajectories that traverse through the detection graphs at various epochs, representing the dynamics of detectable regions of Cislunar space over time. Table 5.2 further details the resultant corridors found between the three architectures.

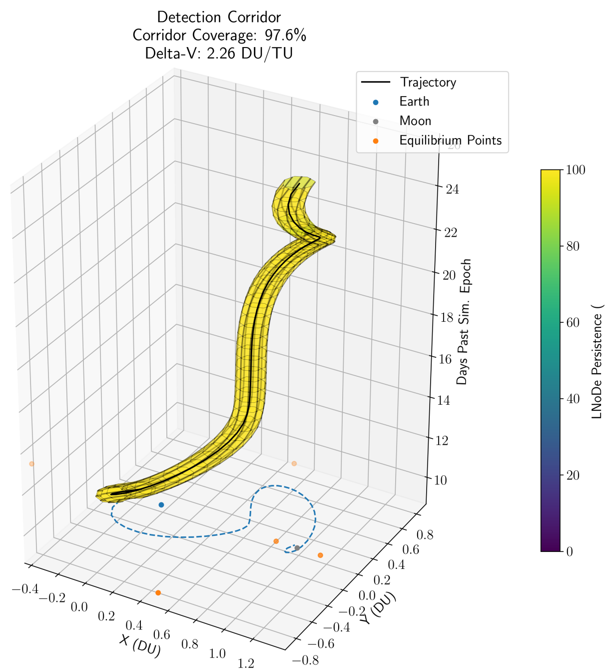


Figure 5.11: Selected persistent detection corridor from architecture 3. The Earth-Moon transfer reference trajectory is projected onto the $x - y$ plane.

Table 5.2: Statistics from corridor generation for each architecture studied. The volume coverage is averaged over the 60 day time span.

Architecture	**Volume Coverage (%)**	**Num. Corridors**	**Num. Corridors (>90%)**	**Avg. Coverage (%)**	**Avg. Delta-V (DU/TU)**
1	32.53	50	4	78.08	2.18
2	16.40	84	0	63.64	1.87
3	61.16	58	21	82.85	2.29

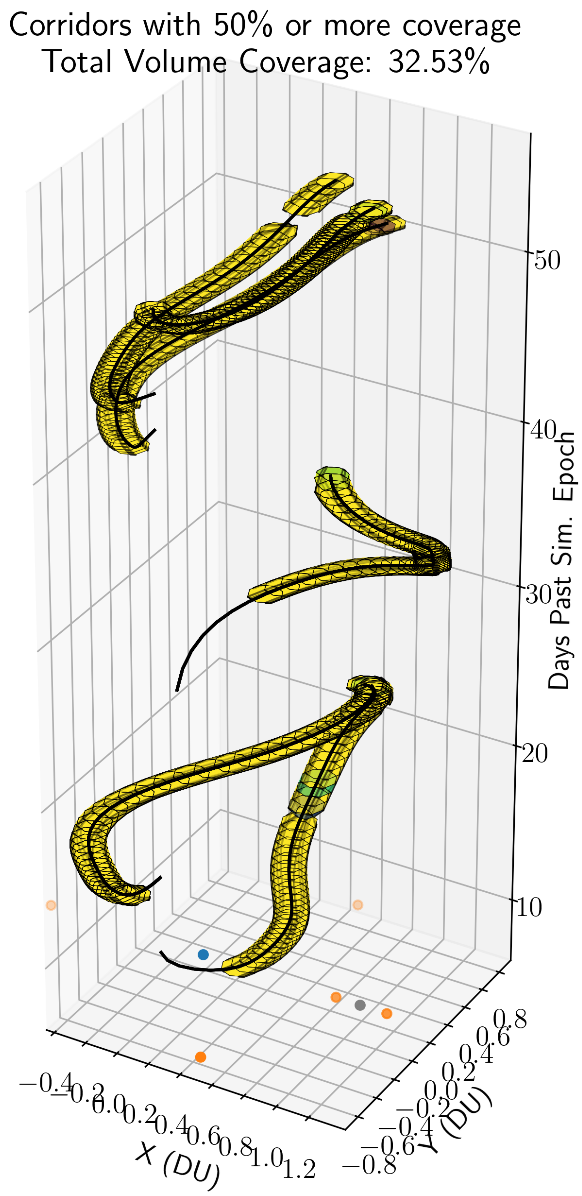


Figure 5.12: Selection of persistent detection corridors for architecture 1.

5.4.5 Discussion

If we select 90% as the minimum corridor coverage required for persistent coverage, then we find that architectures 1 and 3, in Figures 5.12 and 5.14 respectively, admit PDCs with which trajectory designers could expect persistent coverage of a critical asset in a feasible Earth-Moon transfer. Intuitively, architecture 3, which has a higher average volume coverage than architecture

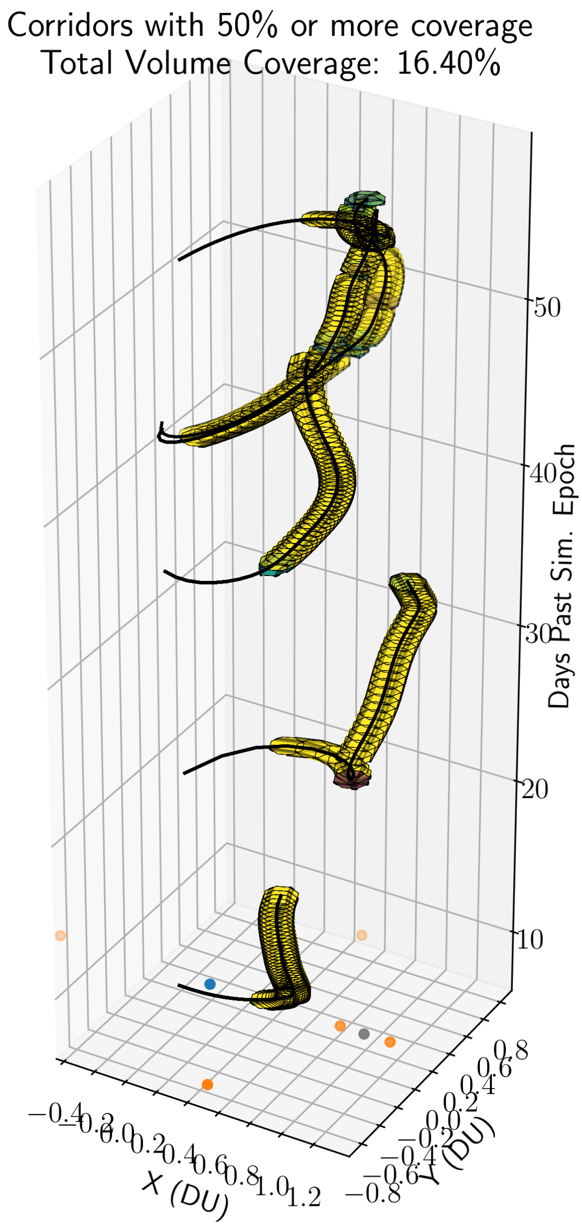


Figure 5.13: Selection of persistent detection corridors for architecture 2.

1, has significantly more PDCs, as seen in Table 5.2. Furthermore, the average coverage of all corridors uncovered in architecture 3's detection graph is relatively high - higher than the total volume coverage, an expected result across the three architectures studied. Architecture 1 has only four PDCs within the 60 day time span due to the less frequent coverage provided to the near-Earth region, as seen in Fig. 5.7b. However, as seen in Fig. 5.12, coverage to the initial departure near

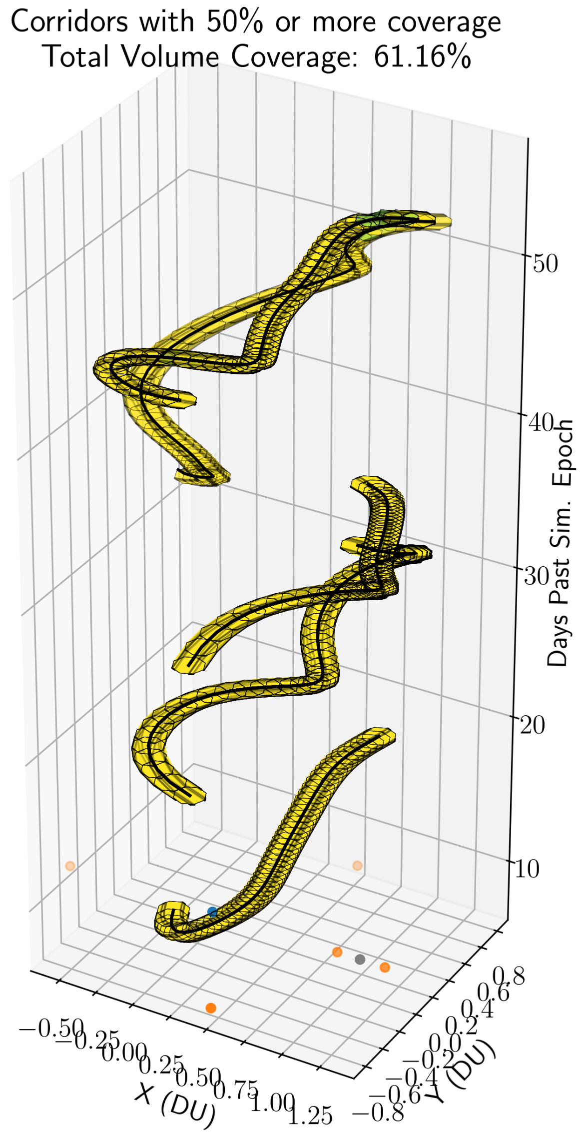


Figure 5.14: Selection of persistent detection corridors for architecture 3.

the Earth periodically returns due to the dynamic solar illumination allowing for PDCs at a couple of epochs within the time span.

In the case of architecture 2, which has a low average volume coverage over the time span, we might have expected that it would be unlikely to uncover detection corridors with high coverage. As seen in Fig. 5.7b, the detectable region is relatively small compared to the total region of interest, mostly remaining around the West Low Prograde observer near the Moon. As such, as seen in Fig. 5.13, the corridors uncovered by our method lack coverage near the Earth, picking up as the reference trajectories near the Moon. However, even with a small and concentrated detection distribution, our method finds multiple corridors exploiting the coverage of this region, with an average corridor coverage of 63.64%. Though this is below our specified 90% coverage to be considered persistent, this elucidates an important result from this work: that low cost architectures with poor performance across the entire Cislunar volume can still be utilized for more focused SSA tasks within the region, such as providing coverage for critical assets in Earth-Moon transfers. This further supports the insights from [64] that suggests an intelligently designed Cislunar SSA architecture must inevitably consider the ways in which cooperative agents will use said architecture.

An added benefit to our method is that each corridor is defined about a reference trajectory modeled with continuous control from which we obtain an estimate of the total delta-v. As an aid for mission design, corridor coverage and delta-v admit a tradespace from which one can identify persistent detection corridors that balance a mission's requirements in safety and cost. This is similar to the tradespace in [64] but requires less computational complexity and deals directly with boolean indicators of detectability, allowing for a more direct representation of coverage in the trajectory optimization process. Furthermore, the statistics in 5.2 can be leveraged as metrics within a Cislunar SSA architecture optimization framework as in [63, 64] to more finely determine the quality of architectures.

It is important to note in this discussion that the total number of persistent detection corridors returned by this method inevitably depends on user's selection of minimum corridor coverage and the corridor radius. For example, Table 5.3 shows the corridor statistics for same three architectures but with a corridor radius of 0.2 DU. The average coverage of the corridors decreases slightly as expected, so the analysis of PDCs may require visual inspection of the corridors. LNoDe persistence is likely to be less uniform along the corridor, and so mission designers may have to evaluate corridors based on desired local detection at individual time steps. Regardless, our method allows for rapid uncovering of corridors, making this type of analysis feasible for mission design.

Table 5.3: Statistics from corridor generation for each architecture studied with a corridor radius of 0.2 DU. Slight decreases of corridor coverage across the architecture are shown in bold.

Architecture	Volume Coverage (%)	Num. Corridors	Num. Corridors (>90%)	Avg. Coverage (%)	Avg. Delta-V (DU/TU)
1	32.53	50	2	**76.16**	2.18
2	16.40	84	0	**61.16**	1.87
3	61.16	58	21	**81.42**	2.29

Chapter 6

Resilience of Cislunar Architectures Using Reachability

6.1 Rapid Low-Thrust Reachable Set Computation

In this contribution, reachable sets for low-thrust spacecraft are approximated using an indirect method from Bowerfind & Taheri [17], similar to one from Patel & Scheers [78]. Bowerfind's algorithm uses an indirect multi-stage formulation (IMF) for solving multiple optimal control problems which approximate the boundary of the reachable set assuming a minimum-time control law. The derivation of the IMF optimal control problem is summarized here.

The flow function $\boldsymbol{F}^i(\boldsymbol{x}^i, \boldsymbol{u}^i, \Delta t)$ from stage i to stage $i+1$ is given by

$$\boldsymbol{x}^{i+1} = \boldsymbol{F}^i(\boldsymbol{x}^i, \boldsymbol{u}^i, \Delta t) = \boldsymbol{x}^{\,i} + \int_{t_i}^{t_{i+1}} \boldsymbol{f}(\boldsymbol{x}(t), t, \boldsymbol{u}(t))dt. \tag{6.1}$$

Next, the Hamiltonian for the minimum-time formulation of the i^{th} stage is written as

$$H^i = \Delta t + (\boldsymbol{F}^i)^\top \boldsymbol{\lambda}^{i+1} + \nu_1(\|\hat{\boldsymbol{\alpha}}^{\,i}\| - 1) \tag{6.2}$$

where Δt is the time step, $\boldsymbol{\lambda}^{i+1}$ is the costate vector at the next stage, and ν_1 is a Lagrange multiplier enforcing the unit norm constraint on the control vector $\hat{\boldsymbol{\alpha}}^{\,i}$. From the Hamiltonian, the costate equations are derived as

$$\boldsymbol{\lambda}^i = \frac{\partial H^i}{\partial \boldsymbol{x}} = H^i_{\boldsymbol{x}} = \boldsymbol{F}_{\boldsymbol{x}}^{iT} \boldsymbol{\lambda}^{i+1} \tag{6.3}$$

and the control law is given by

$$\frac{\partial H^i}{\partial \boldsymbol{u}} = H_u^i = \mathbf{0} = (\boldsymbol{F}_{\boldsymbol{u}}^i)^{\mathrm{T}} \boldsymbol{\lambda}^{i+1} + \nu_1 \left(\frac{\hat{\boldsymbol{\alpha}}^i}{\|\hat{\boldsymbol{\alpha}}^i\|} \right). \tag{6.4}$$

From (6.4), the primer vector optimal control law becomes

$$\hat{\boldsymbol{\alpha}}^i = -\frac{(\boldsymbol{F}_{\boldsymbol{u}}^i)^{\mathrm{T}} \boldsymbol{\lambda}^{i+1}}{\left\|(\boldsymbol{F}_{\boldsymbol{u}}^i)^{\mathrm{T}} \boldsymbol{\lambda}^{i+1}\right\|}. \tag{6.5}$$

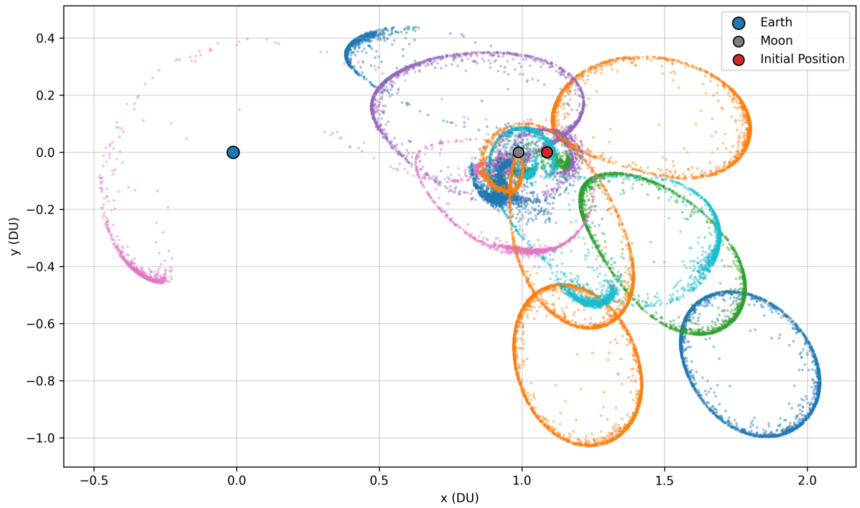
In the CR3BP dynamics, the primer vector is augmented by the maximum thrust, in this work chosen to be low-thrust at 0.18 $DU/TU^2 \approx 4.9 \times 10^{-7} km/s^2$, similar to the resultant control magnitude from [17]. The costate equations are integrated backwards in time using the state transition matrix, $\boldsymbol{F}_{\boldsymbol{x}}^i = \Phi$, and the sensitivity matrix, $\boldsymbol{F}_{\boldsymbol{u}}^i = \Omega$, which are obtained from a forward integration of an unpowered reference trajectory. During the forward integration, these matrices are computed at each time step via the following equations:

$$\boldsymbol{F}_{\boldsymbol{x}}^i = \frac{\partial \boldsymbol{x}^{i+1}}{\partial \boldsymbol{x}^i} = \Phi(t^{i+1}, t^i) = \int_{t^i}^{t^{i+1}} A(t)\Phi(t, t^i) dt \tag{6.6}$$

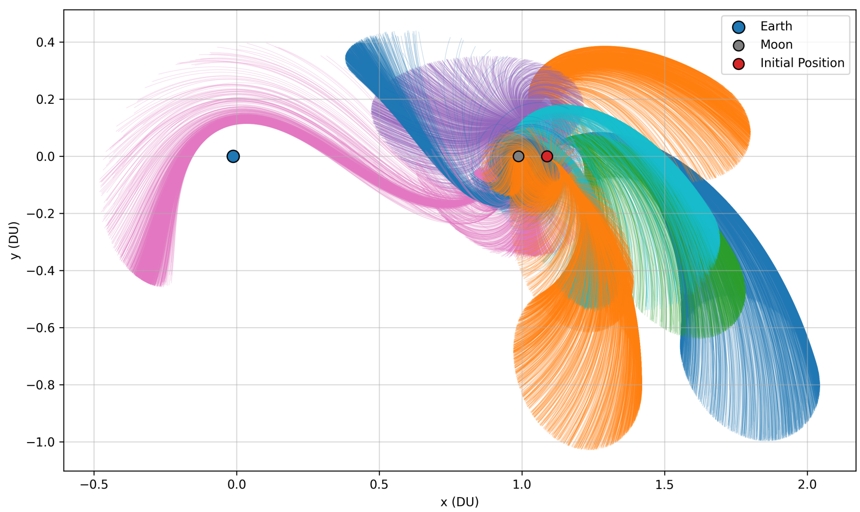
$$\boldsymbol{F}_{\boldsymbol{u}}^i = \frac{\partial \boldsymbol{x}^{i+1}}{\partial \boldsymbol{u}^i} = \Omega(t^{i+1}, t^i) = \int_{t^i}^{t^{i+1}} \big[A(t)\Omega(t, t^i) + C(t)\big] dt \tag{6.7}$$

where $A(t)$ and $C(t)$ are the state and control Jacobians of the CR3BP dynamics, respectively.

The algorithm itself consists of three main steps. First, a ballistic reference trajectory is generated via forward integration, obtaining state transition and sensitivity matrices. Next, a large number of costates are uniformly sampled from a 4D dimensional hypersphere, which are then integrated backwards in time using Eq. (6.4) and used to compute the primer vector in Eq. (6.5). Finally, a forward integration of the state dynamics with stored primer vectors is performed to obtain the reachable set and intermediate states. Figure 6.1 shows a set of trajectories and their final states from an initial position near the L2 Lagrange point with velocity directions uniformly sampled in 2D. For more details on the algorithm, see [17].



(a)



(b)

Figure 6.1: Reachable sets and intermediate states. Colors represent trajectories with unique initial velocities.

6.1.1 Precomputing Reachable Sets

While Bowerfind's algorithm is efficient for computing reachable sets compared to other methods, its computational cost is not insignificant if used in real-time for many architectures over long periods of time. As such, a database of precomputed reachable sets is generated for use in this work to reduce the computational burden of analyzing architecture resilience. A grid of positions is generated to cover the relevant Cislunar volume, expanding out to the zero-velocity surface of the CR3BP at a Jacobi energy allowing for a ballistic trajectory to reach all regions except for L4 and L5. The grid of initial positions is shown in Figure 6.2. Next, velocity magnitudes are sampled based on the range of Jacobi energies that allow for ballistic movement at each position. For a position (x, y), one can solve for velocity from the (planar) Jacobi energy equation:

$$v^2 = (x^2 + y^2) + \frac{2(1-\mu)}{r_1} + \frac{2\mu}{r_2} - J \tag{6.8}$$

for Jacobi energy J evenly spaced in the range $(2.99, 3.6)$. For each point, up to six velocity magnitudes are calculated as long as $v^2 > 0$. Finally, velocity directions uniformly distributed about each position, resulting in six equally spaced velocity directions.

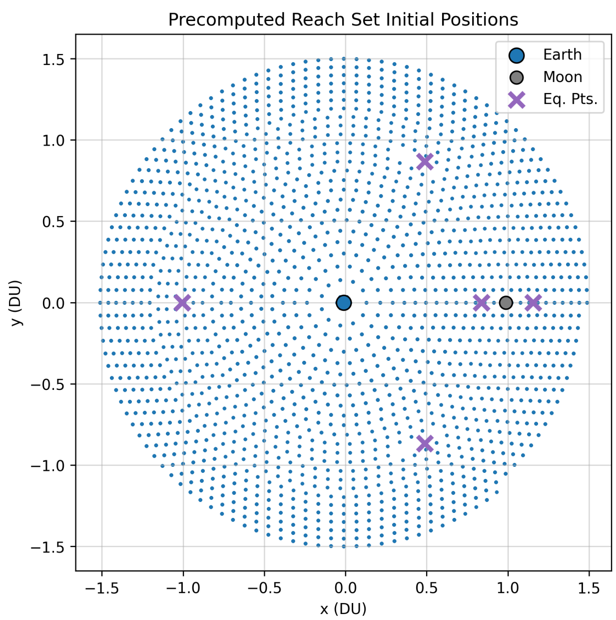


Figure 6.2: Grid of initial positions for precomputed reachable sets.

Using this method, about 30,000 unique initial states are generated. For each reachable set, 2,000 uniformly distributed costate directions are generated, and trajectory data for 200 time steps over a 7-day period is stored. The entire generation process is run in parallel on a high-performance computing cluster, and the resultant database of around 250GB is stored in an HDF5 file, enabling efficient querying. In `Python`, `h5py` is used to read the database through a custom interface, allowing querying of reachable set data by initial position.

6.1.2 Calculating Coverage of Reachable Sets

To quantify architecture resilience to disruptions in the astrodynamics sense, a framework is developed that tracks the coverage of reachable sets within an architecture's field of regard. By tracking the coverage of reachable sets over time, one can quantify how well an architecture's evolution of coverage matches the evolution of spacecraft states over time. This is in contrast to a static volume coverage metric, which determines performance based on the average volume of detectable space over time. A volume coverage, as in [63], keeps track of the amount of times that sampled volume points are detected by an architecture over time, but does not account for the dynamics of a spacecraft at those points. By having access to the reachable sets of spacecraft at those points, a more fundamental quality of coverage is accessed that reveals how well a spacecraft can be covered over time, starting at any initial position in Cislunar space. This paper provides a systematic framework for obtaining this information, which can be presented in a variety of formats.

The processs of determining the coverage of reachable sets is essentially the same as for volume coverage, with some extra steps to obtain the relevant data. First, a planar point in Cislunar space, p_s, is sampled and the nearest initial position, p_0, in the database is found through the HDF5 interface. Then, all reachable sets for that position are obtained (here, up to 36 unique initial states per position), collected in an array $\boldsymbol{X}_q^{p_s} \in \mathbb{R}^{n\times 2000\times 200\times 4}$ for n unique initial states. For each time $t_i \in T = [t_0, t_1, ..., t_N]$, the corresponding trajectory state is checked to determine if it meets the detection threshold by the architecture (exceeding minimum SNR of 6). Ones and zeros, representing a detectable and undetectable state, respectively, are stored in an array, $\boldsymbol{M}_{t_i}^{p_s} \in \mathbb{R}^{n\times 2000}$. To obtain the average coverage of all reachable set trajectories eminating from p_s over T, the concatenation of all masks flattened ($\bar{\boldsymbol{M}} = \text{flatten}(M)$) is averaged:

$$\eta(p_s, T) = \text{mean}\left(\bar{\boldsymbol{M}}_{t_0}^{p_s}, \bar{\boldsymbol{M}}_{t_1}^{p_s}, ..., \bar{\boldsymbol{M}}_{t_N}^{p_s}\right) \tag{6.9}$$

The final result is a coverage metric η associated with the initial sampled point at time t_0 , indi-

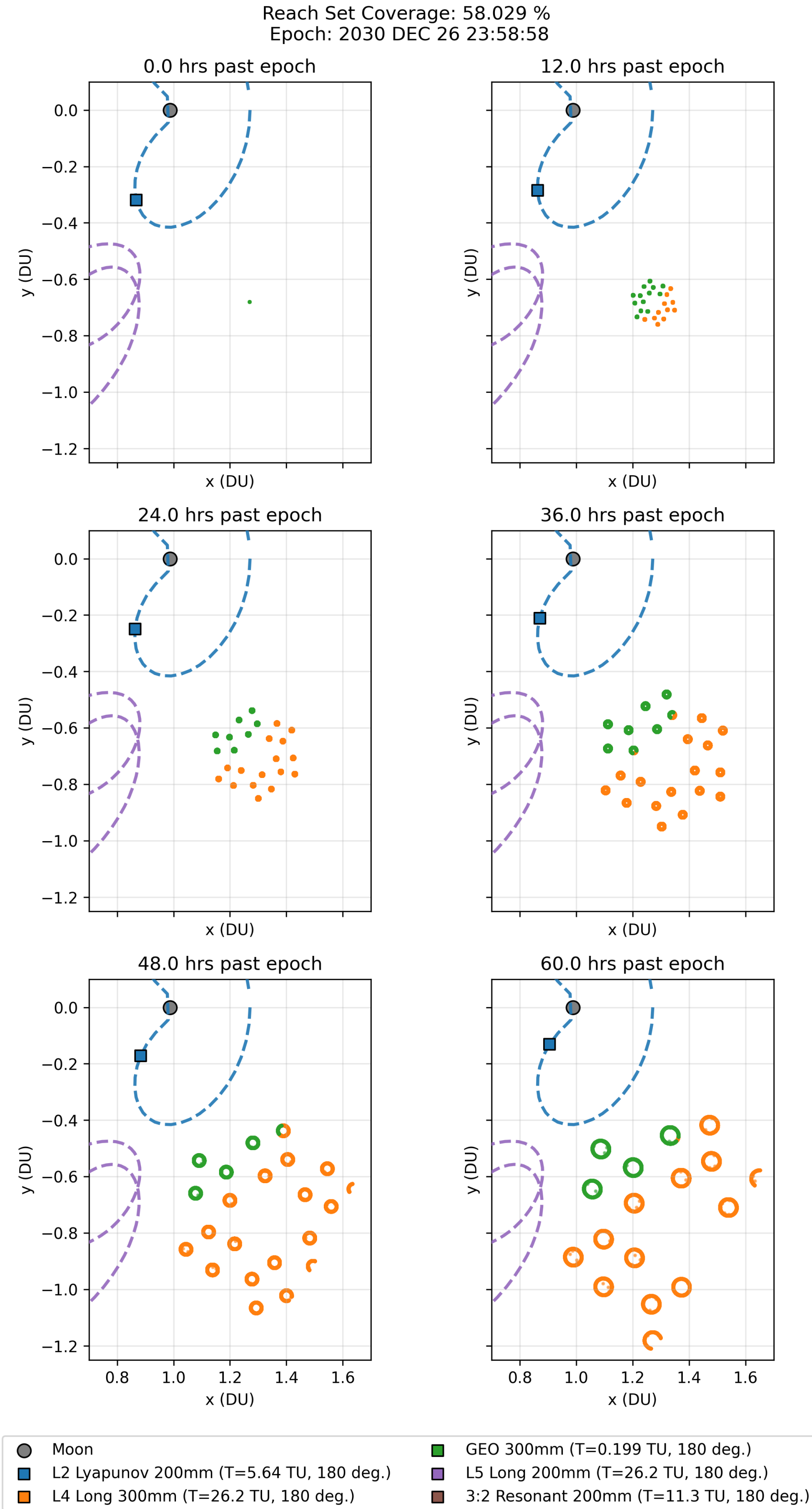


Figure 6.3: Evolution of reachable set over time. Green states are detectable by the architecture, while orange states are not.

cating how well the architecture performs at that specific location over the time horizon T. This process can also be conceptualized as tracking the intersection of reachable sets and the evolution of architecture coverage over time:

$$\eta(p_s, T) = \frac{|\{\boldsymbol{x}_r \in \mathcal{R}(p_s, T) : \boldsymbol{x}_r \text{ is detectable}\}|}{|\mathcal{R}(p_s, T)|} \tag{6.10}$$

In Figure 6.3, an initial position is sampled near the Moon, and is shown with its queried reachable sets. Over time, the boundary of each set expands, as expected, and the states' coverage by the sample architecture is tracked. Notably, not all of the reachable sets are covered by the architecture at all times. This is a key insight not captured by typical coverage metrics but ellucidated here: that no position is static and just because a region is detectable at one time does not mean it will continue to be in the future. The framework presented here allows not only for quantification of this effect, but also for an intuitive visualization of it.

6.2 Resilience of Cislunar Architectures

6.2.1 Resilience Maps

The primary method by which one can quickly assess the resilience of an architecture is through the use of a heat map. The process is repeated for a grid of 2D positions throughout Cislunar space. A percentage of coverage of reachable sets is then returned for each position, indicating how well the architecture performs at that location. The data can then be visualized as a heat map, here referred to as a resilience map. Figure 6.4 shows a map for the sample architecture previously discussed over a 7-day period. In contrast to a coverage map as in Figure 6.6, the resilience map shows the intersection of an architecture's coverage and a spacecraft's reachable set throughout Cislunar space. Notably, the resilience map does not directly show the distribution of coverage over the volume, but rather the distribution of coverage over the reachable set trajectories.

Given that the reachable set trajectories are computed up to 7 days, a map generated at time t_i is a snapshot of architecture resilience from t_i to $t_i + 7$ days. This can be extended to a

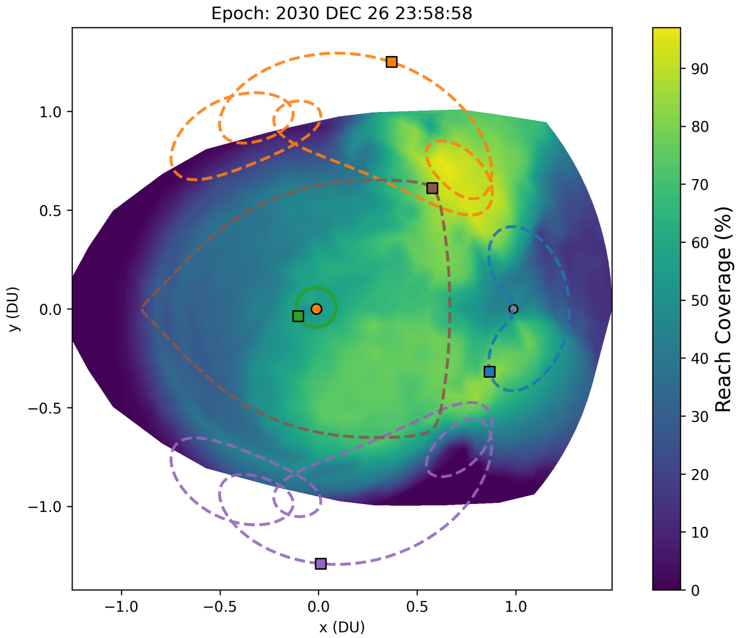


Figure 6.4: Resilience map for a sample architecture over a 7-day period.

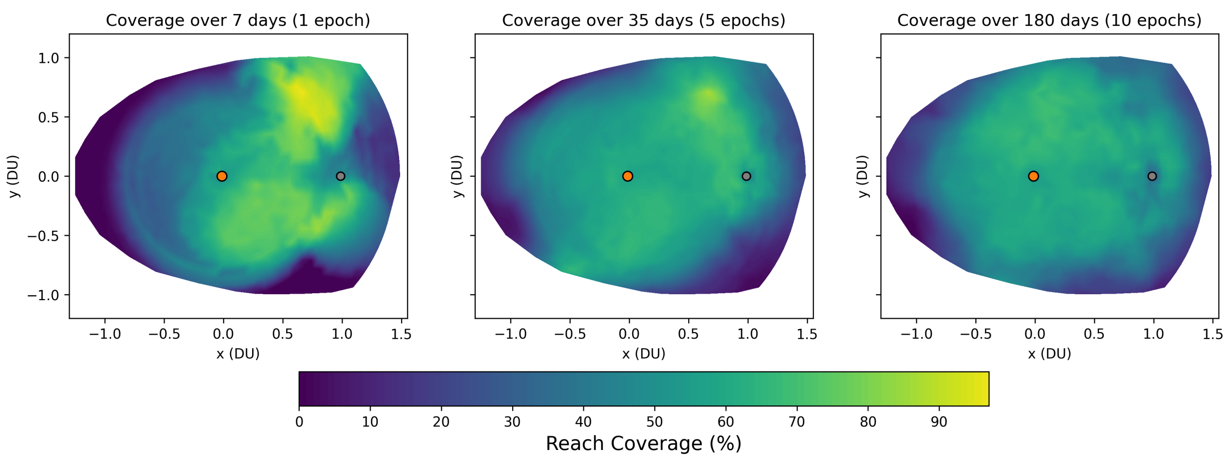


Figure 6.5: Resilience maps for varying time horizons and coverage epoch frequencies.

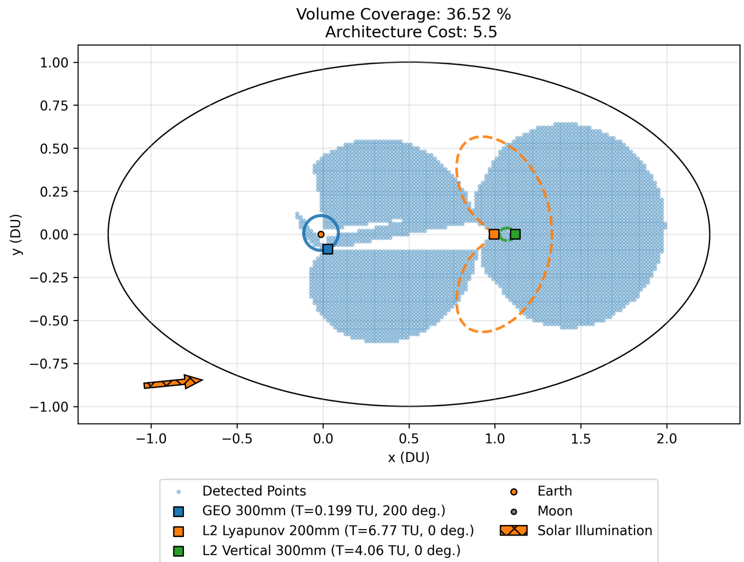
longer time horizon by generating a map at multiple time steps and averaging the results. The total duration and relative frequency of evaluating set coverage within the duration leads to different levels of map fidelity, and averaging over longer time horizons can lead to smoothing out potentially interesting features. Figure 6.5 shows three different resilience maps for the same architecture but with varying time horizons and coverage epoch frequencies. Generally, a resilience map generated over a shorter horizon lends itself to capturing more detailed features of local regions, while a longer horizon can provide a more global view of resilience. The choice of time horizon and frequency of coverage epoch calculation should be made based on the specific application and desired level of detail.

The resilience maps can be used to identify areas of weakness in an architecture's coverage distribution and evolution, allowing for a localized analysis of performance that is not possible with a volume coverage metric. For example, in a hypothetical mission with a high value asset, there may be a disruption to the spacecraft that causes it to drift from its planned trajectory. In such a case, a resilience map can be used to determine the likelihood of the spacecraft being covered by an architecture, assuming observers are tasked accordingly. In that context, the resilience map can be interpretted as a probability distribution of detecting an uncooperative spacecraft starting at a given position. If an area has low resilience, then it can be said that a trajectory starting in that area is less likely to be detected than one starting in a region with high resilience.

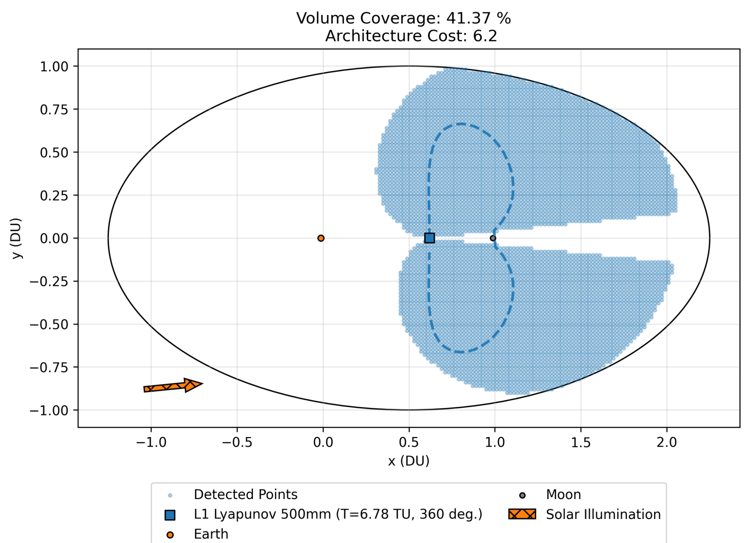
6.3 Results and Discussion

To demonstrate the utility of this novel resilience framework, it is applied to a collection of previously studied architectures for Cislunar SSA. These architectures exist along a Pareto front, representing a trade-off between cost and coverage of the Cislunar volume. The resultant resilience maps are used to identify areas of weakness in coverage distribution and evolution, allowing for a localized analysis of architecture performance that is not possible with a volume coverage metric. Furthermore, resilience maps can be used in the analysis of individual observers, aiding in optimal architecture design and fine-tuning.

The three architectures selected for analysis are shown in Figure 6.6. An epoch of analysis is arbitrarily chosen to be June 30, 2030 00:00:00 UTC and the resilience maps are generated within a 1080 day period to sufficiently showcase the affects of varying solar illumination. The architectures are selected to represent a range of coverage and cost as discussed in [63, 64].

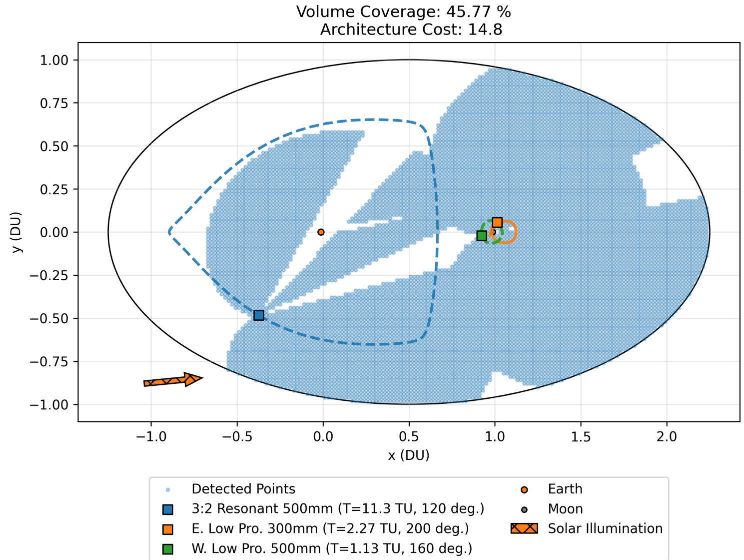


(a) Architecture 1.



(b) Architecture 2.

Figure 6.6: Selected Pareto-optimal architectures for analysis (part 1).

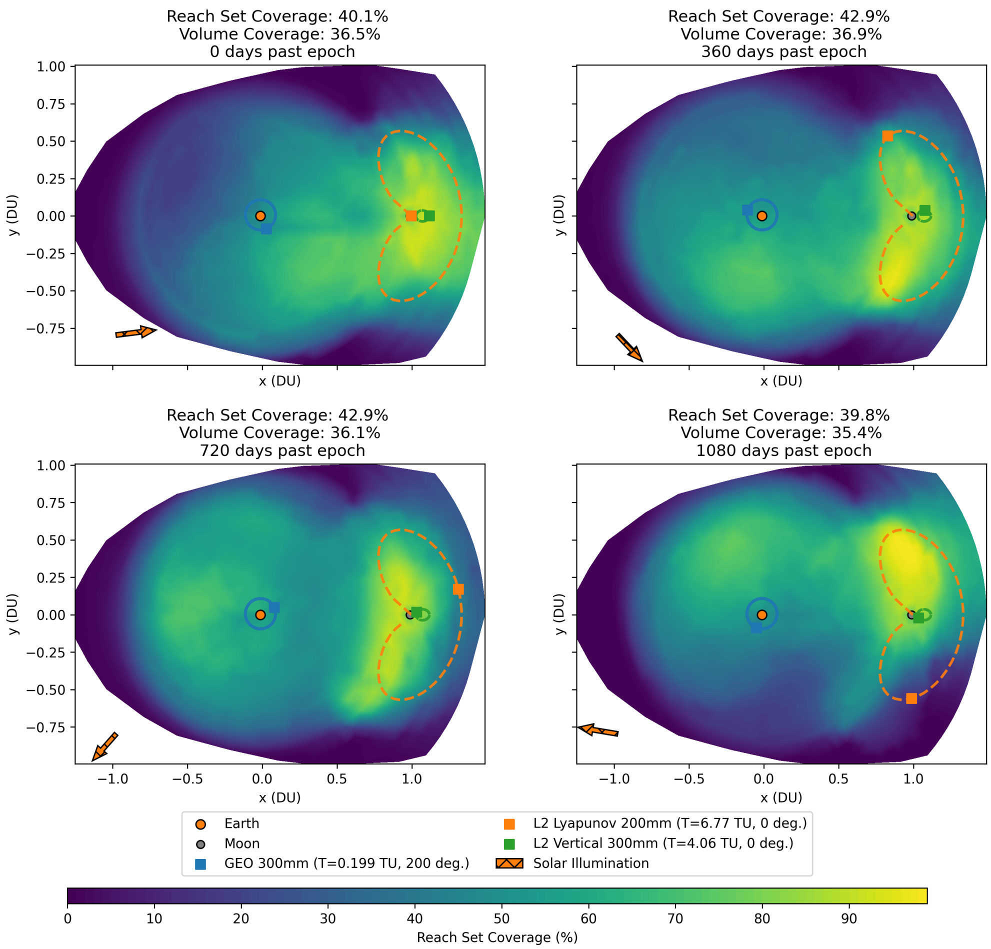


(c) Architecture 3.

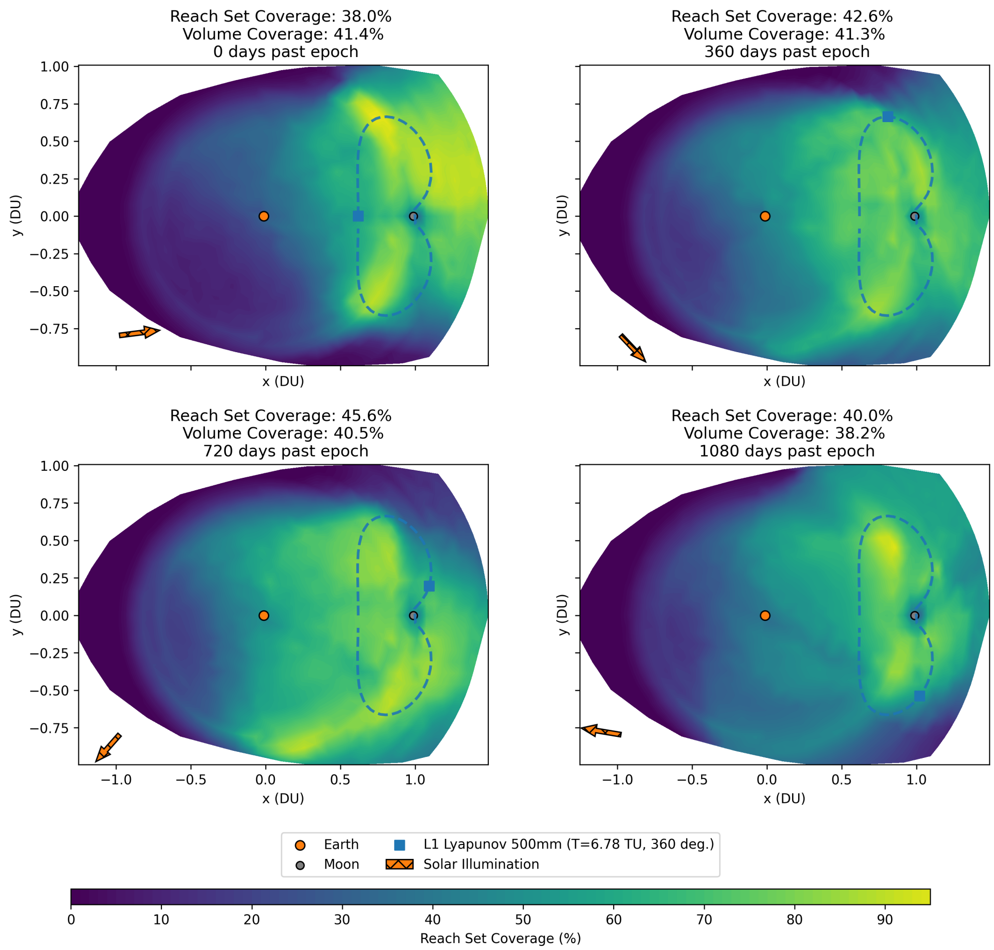
Figure 6.6: Selected Pareto-optimal architectures for analysis (continued).

6.3.1 Resilience Maps for Selected Architectures

Figure 6.7 shows the resultant resilience heat maps for the three selected architectures at four times starting from the epoch of analysis. As expected, the architecture with the highest volume coverage (and highest cost) in Figure 6.7c (architecture 3) is able to cover the most reachable set trajectories throughout the Cislunar space over the long time horizon. However, the distribution of reachable set coverage varies quite significantly due to the long period of the 3:2 Resonant observer and the varying solar illumination. While this architecture has a high volume coverage averaged over time, the resilience map shows a relatively large gap in coverage especially at the initial epoch. This is especially interesting in that the coverage distribution at the initial epoch in Figure 6.6c shows a significant amount of coverage near the Earth, but the architecture is not able to cover reachable sets eminating from this detectable region. Here, though this region is initially detectable, the CR3BP dynamics are misaligned with the architecture's coverage evolution. This mismatch is not necessarily captured by analyzing only the coverage distribution in Figure 6.6c, but with the resilience map one can identify this vulnerability and take steps to address it.

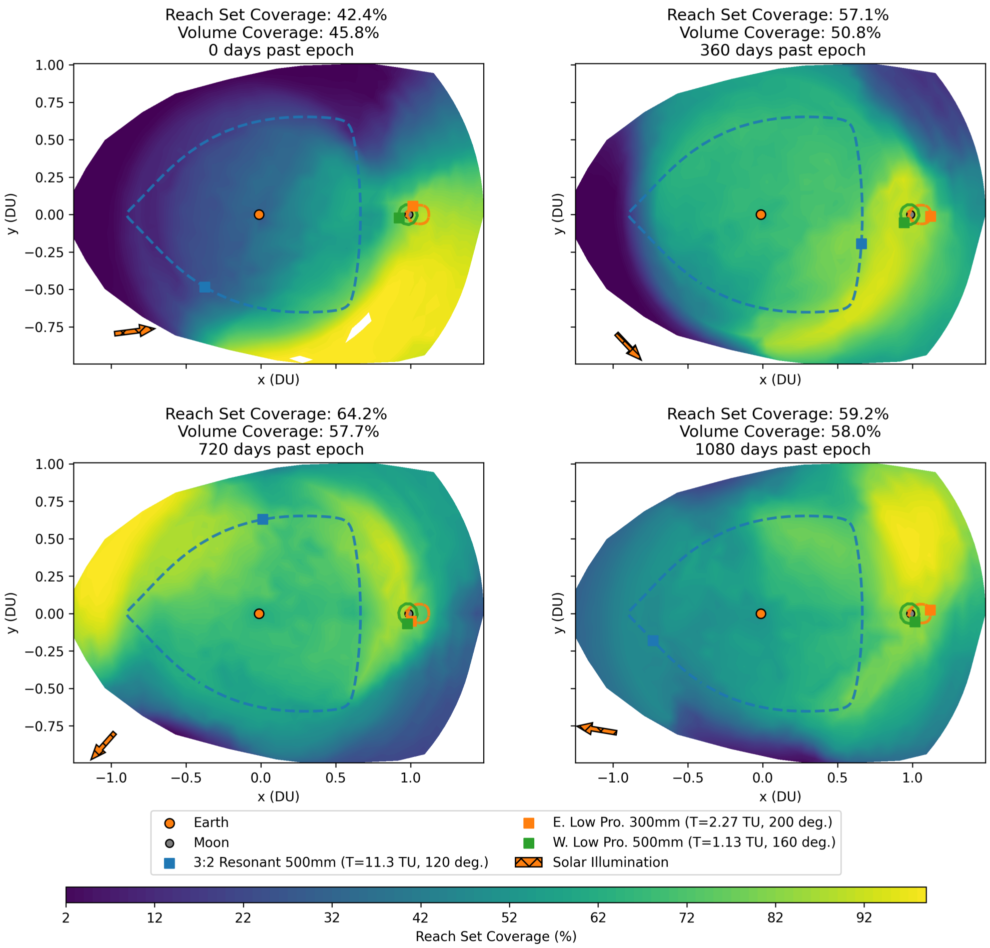

Reach Set Coverage: 40.1%
Volume Coverage: 36.5%
0 days past epoch
Reach Set Coverage: 42.9%
Volume Coverage: 36.9%
360 days past epoch
Reach Set Coverage: 42.9%
Volume Coverage: 36.1%
720 days past epoch
Reach Set Coverage: 39.8%
Volume Coverage: 35.4%
1080 days past epoch
y (DU)
x (DU)
Earth
Moon
GEO 300mm (T=0.199 TU, 200 deg.)
L2 Lyapunov 200mm (T=6.77 TU, 0 deg.)
L2 Vertical 300mm (T=4.06 TU, 0 deg.)
Solar Illumination
Reach Set Coverage (%)

(a)



(b)

Figure 6.7: Selected Pareto-optimal architectures for analysis (part 1).



(c)

Figure 6.7: Selected Pareto-optimal architectures for analysis (continued).

An important point to reiterate in these maps is that they are not showing the distribution of coverage over the volume as in Figure 6.6. Rather, they are showing how well the architecture is able to detect reachable sets eminating from each point in Cislunar space over time. As such, in all the selected architectures in Figure 6.7 there are regions where there is no coverage, but from these regions there are reachable sets with trajectories that eventually become detectable. This is an intuitive concept that is easily captured by the resilience maps, providing a visualization in a single plot of how well an architecture can detect spacecraft over time, accounting for the dynamics of Cislunar space. For example, architecture 2 in Figure 6.6b has a fairly low volume coverage at the initial epoch, covering only the right half of the volume. However, its resilience map in Figure 6.7b over the 1080 day period shows that there are many trajectories outside of this initial region that eventually become detectable by the architecture - even some starting near L3. This architecture eventually covers a large portion of Cislunar space, but due to the solar illumination at the initial epoch has a low volume coverage. However, due to this architecture consisting of an observer only near the Moon, even as the solar illumination changes, the resilience map shows a consistent lack of coverage of reachable sets within the so-called Earth-Moon corridor. This is an indication that the coverage within the field of regard of the observers in this particular architecture is mismatched to spacecraft dynamics in the region, even if the volume coverage is generally high.

On the other hand, the architecture 1 in Figure 6.6a has a lower cost than architecture 2, and therefore a lower total volume coverage over long periods of time as discussed in [63]. It may be expected that the resilience of this architecture would generally be lower as well, but Figure 6.7a shows that the inclusion of a GEO observer allows for a larger portion of trajectories emanating from the Earth-Moon corridor to be covered. In fact, the total coverage of reachable sets in this lower cost architecture is comparable to the higher cost architecture in Figure 6.7b. In this case, it seems to be primarily due to the inclusion of a GEO observer, whose coverage distribution repeats more frequently allowing for a more consistent coverage of reachable sets over time in this specific region.

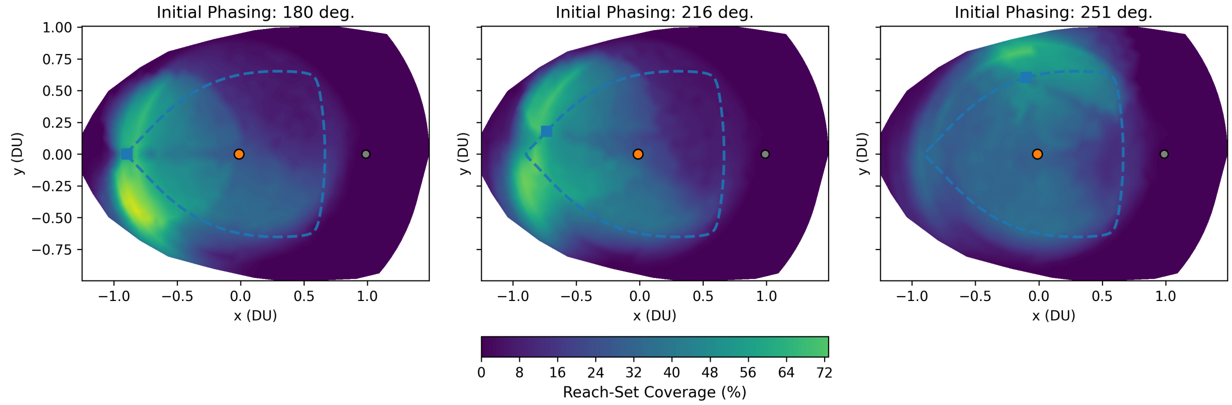


Figure 6.8: Observer in 3:2 Resonant orbit with 300mm telescope.

6.3.1.1 Resilience Informed Architecture Fine Tuning

With these resilience maps, the quality of coverage provided by an architecture can be more locally assessed, allowing for targeted improvements. For example, as in architecture 3 in Figure 6.7c, there is a significant lack of reachable set coverage near the Earth at the initial epoch, even though that region is detectable by the architecture as seen in Figure 6.6c. This can be addressed by adding an observer to the architecture whose resilience map is more aligned with the Cislunar dynamics at that time. As an example, resilience maps are created for a 300mm observer in the same 3:2 Resonant orbit but varying the initial phasing of the orbit. These are shown in Figure 6.8, and demonstrate the significance of the initial phasing of an observer on its coverage distribution as previously discussed.

From this brief analysis, it appears that placing an observer into the 3:2 Resonant orbit with an initial phasing of 216° would increase reachable set coverage and thus resilience in the region near the Earth. The resultant modified architecture is shown in Figure 6.9, which shows an increase in reachable set coverage near the Earth from around 20% to around 50%. As an illustrative example, this likely isn't the optimal solution to maximize resilience in this region, but it does demonstrate how the framework can be used to fine-tune architectures to improve resilience, where with a coverage distribution map this original vulnerability would not be easily identified.

This sort of analysis can be leveraged to fit within a larger framework for architecture design and analysis, such as those in [79, 64, 6, 97, 89]. By systematically generating resilience maps for each potential observer in Cislunar space, one can identify and categorize observers based on the spatial distribution of their reachable set coverage. These resilience maps can be categorized using a clustering method such as in [64] to identify groups of observers with similar resilience distributions. Combined with metrics such as volume coverage, cost, capacity, and others, a more comprehensive framework for architecture optimization can be developed to improve the quality of Cislunar SSA activities.

6.3.2 Resilience Maps for Selected Observer Orbits

As discussed throughout the literature [94, 63, 89] observers with periods resonant with the Earth-Moon synodic period tend to have favorable characteristics for Cislunar SSA. This conclusion comes from an analysis of the volume coverage distribution over time. However, this type of analysis does not necessarily quantify how well an observer's coverage distribution matches the dynamics of spacecraft in Cislunar space, in the context of persistent tracking of space objects. A resilience

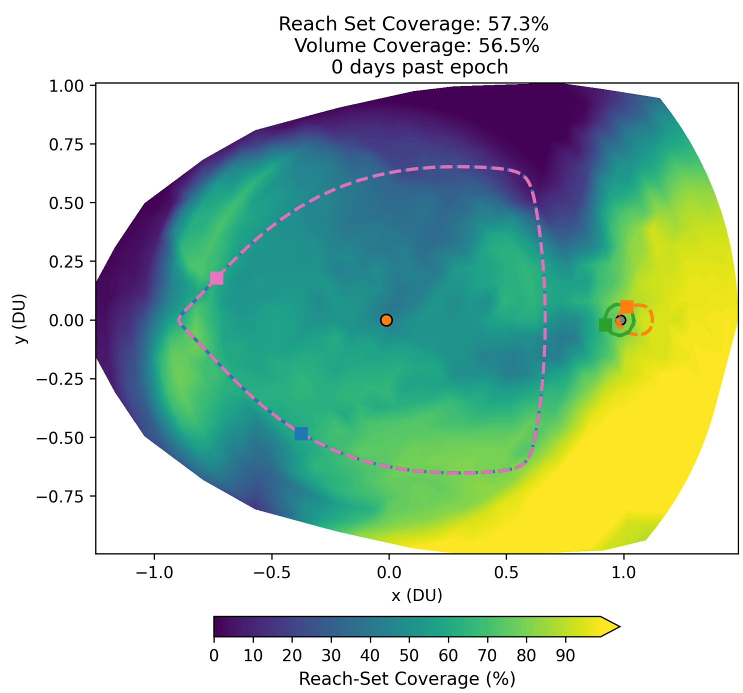


Figure 6.9: Modified architecture resilience map.

map can be used to quantify this effect, and to identify which observers best track reachable sets of spacecraft over time. As an example of this type of analysis, three orbits are selected for analysis: a 1:1 synodic resonant L1 Lyapunov orbit, a 3:4 synodic resonant 3:2 Resonant orbit (3:2 resonance with the lunar sidereal period), and a 1:1 synodic resonant Distant Prograde orbit (DPO). The maps are calculated over the entire resonant period of each orbit to capture the extent of their coverage distribution, with five coverage epochs evenly spaced throughout the period. As previously discussed, averaging the reachable set coverage over these longer periods of time lend to smooth out potentially interesting features, but reveal a more global view of resilience. Furthermore, the initial phasing of each observer is varied, demonstrating how initial position and solar illumination interact to impact coverage distribution. The resultant resilience maps are shown in Figures 6.10, 6.11, and 6.12, with a single contour level at 25% plotted on each to represent a minimum level of coverage or probability of detection one might desire for resilience analysis.

These contrasting resilience maps show how different geometries between observer orbits result in varying abilities to cover spacecraft reachable sets over time, even though all orbits have resonance with the synodic period. Each observer has a unique initial phasing that maximizes its total coverage of the reachable sets, but the distribution of regions of coverage over the 25% minimum for analysis varies significantly, both between phasings and across orbits. The L1 Lyapnuov (Figure 6.10) and DPO (Figure 6.12) have the same period and traverse a similar region of Cislunar space, but due to the varied geometry of the orbits, the DPO tends to have a more consistent distribution of reachable set coverage with varying initial phasing. On the other hand, the L1 Lyapunov observer has more significant variation of high and low coverage regions over varying its initial phasing. This is an indication that the DPO may be more resilient to drift from its 1:1 resonance with the synodic period, while the L1 Lyapunov's coverage of spacecraft trajectory is more sensitive to the initial phasing of the orbit.

The 3:2 Resonant observer in Figure 6.11 has a vastly different geometry than the other two, and as such has a very different distribution of reachable set coverage throughout the region. Earth-Moon sidreal resonant orbits such as this one are generally considered beneficial for Cislunar

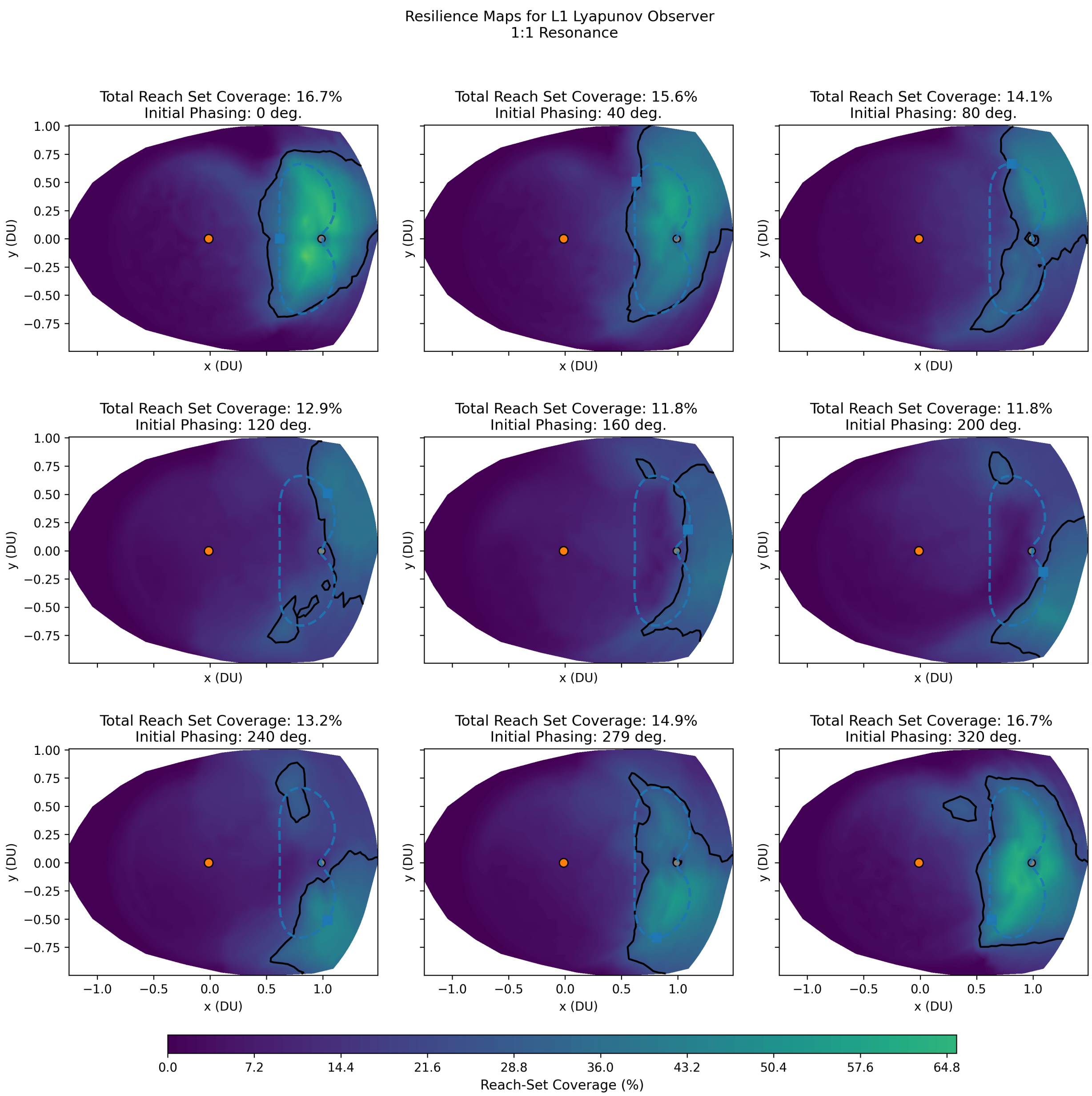


Figure 6.10: Resilience maps for 1:1 synodic resonant 300mm L1 Lyapunov observer with varying initial phasing.

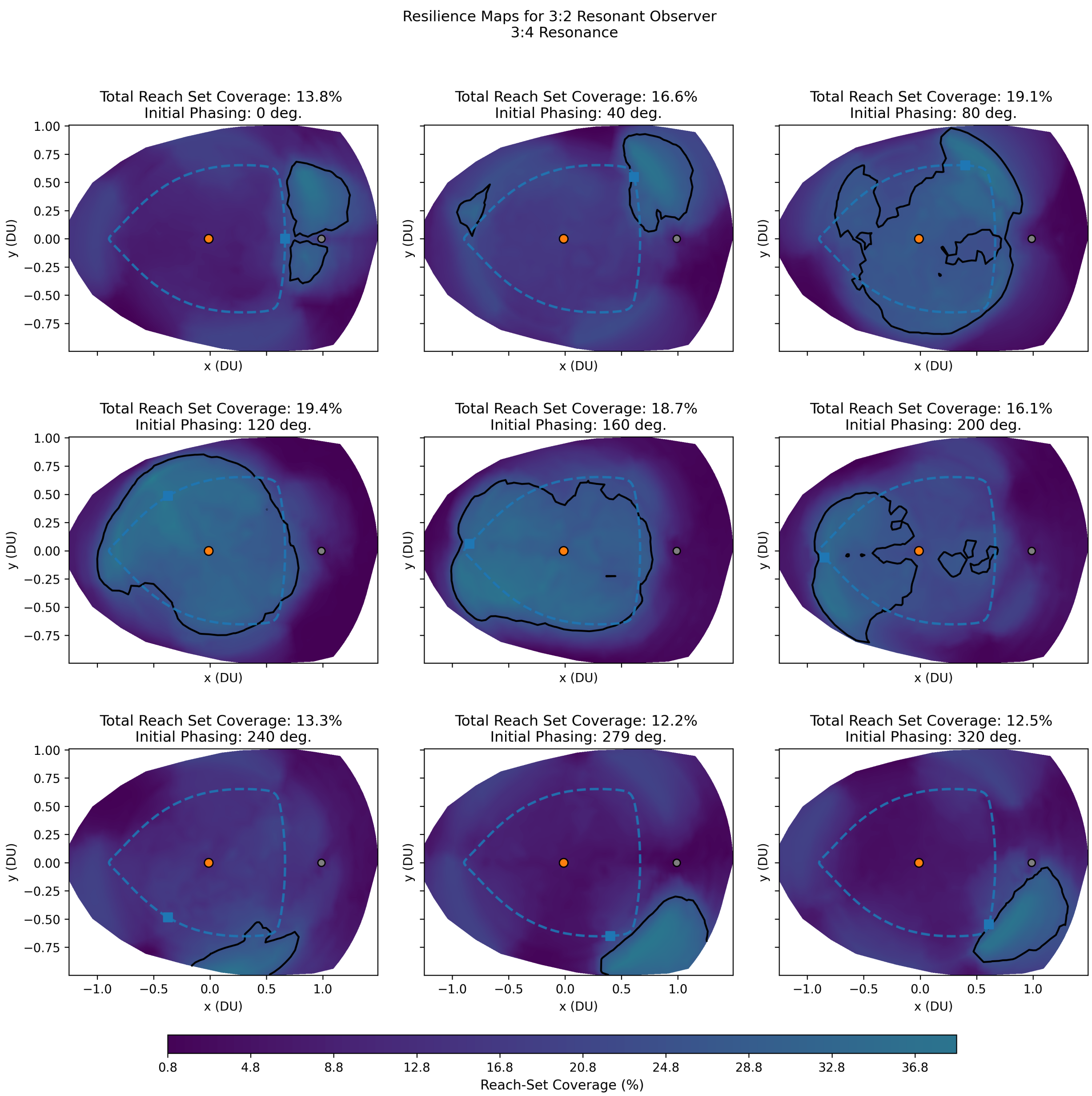


Figure 6.11: Resilience maps for 3:4 synodic resonant 300mm 3:2 Resonant observer with varying initial phasing.

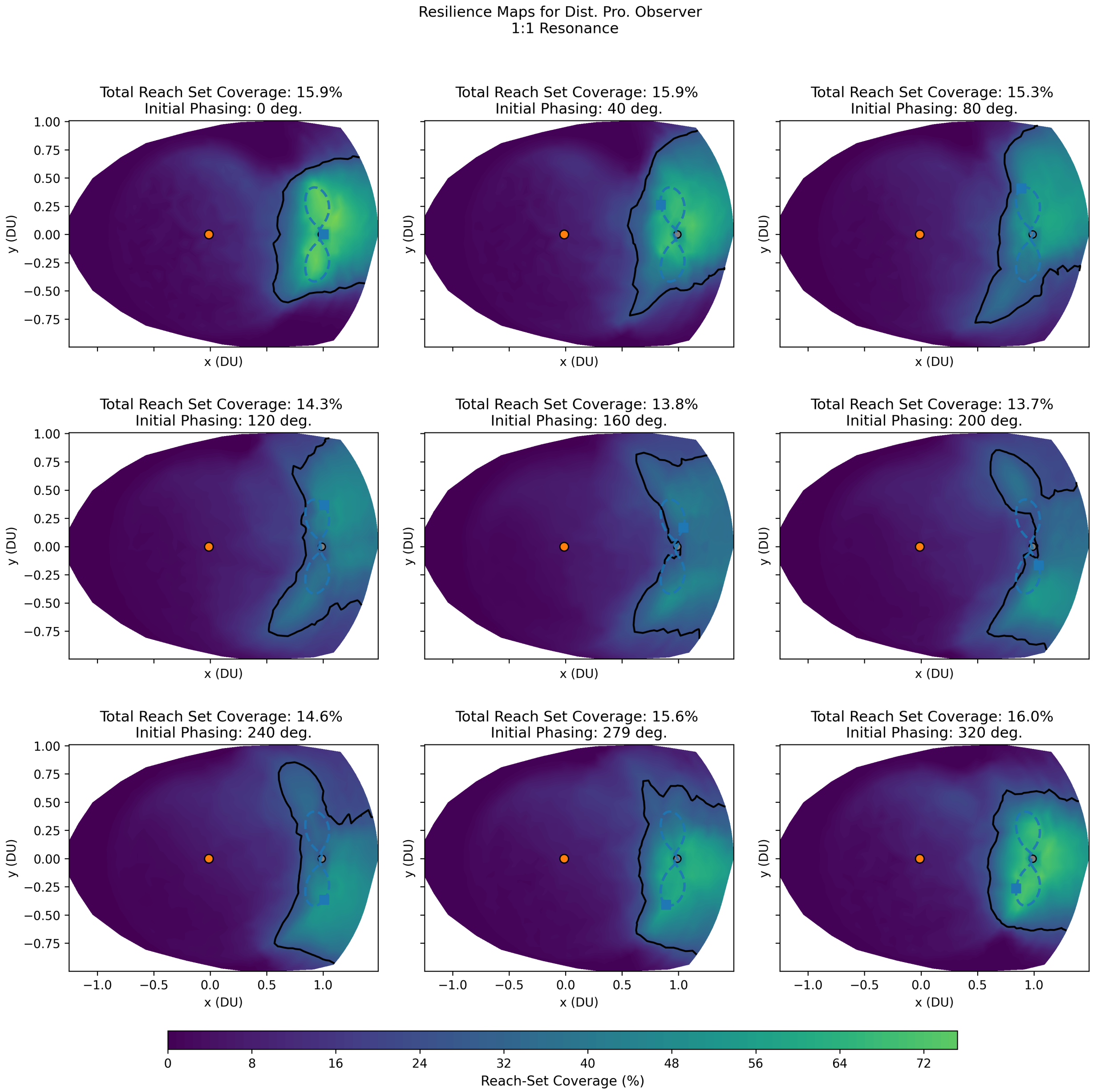


Figure 6.12: Resilience maps for 1:1 synodic resonant 300mm Distant Prograde observer with varying initial phasing.

SSA due to their wide traversal of the the Cislunar volume, but the resilience maps show that the effective coverage considering Cislunar dynamics is quite inconsistent across initial phasings. The 3:2 Resonant observer with an initial phasing of 120° has the highest reachable set coverage, and a wide distribution of reachable set coverage over the 25% minimum for analysis purposes. To reiterate, this map is averaged over the entire resonant period of the orbit capturing the large extent of its coverage evolution. The regions of space within the 25% contour represent the proportion of all possible trajectories starting from those initial positions that are persistently detectable by the observer. This can similarly be interpretted as the probability of detecting a spacecraft starting from any given position. Thus, another tradespace emerges from this analysis, comparing architectures with broad distribution of relatively low reachable set coverage (probability of detection) such as the 3:2 Resonant observer, with architectures with more localized regions of high reachable set coverage such as the DPO and L1 Lyapunov observers.

6.3.3 Discussion

Similar to the novel framework presented in the previous contribution, quantifying resilience via reachable sets can be leveraged within various parts of Cislunar SSA missions. For example, architecture designers can implement a resilience-based metric to evaluate architecture iterations within an optimization framework, such as those proposed in previous contributions. Additionally, mission planners can use resilience maps to inform the sensor tasking of already existing architectures, allowing for tasking of sensors towards regions of high reachable set coverage, thus increasing the probability of detecting spacecraft in those regions. On the other hand, mission planners can also use resilience maps to identify potential gaps in coverage and reallocate resources or adjust sensor configurations to improve overall mission effectiveness.

The most important aspect of this framework is that is provides a systematic process to quantify the ability of an architecture to detect spacecraft over time, directly accounting for the dynamics of low-thrust spacecraft in Cislunar space. This is in contrast to our previous contributions that focused on volume coverage or coverage of individual trajectories. With resilience maps tracking coverage of reachable sets over time, we are able to more directly quantify the quality of coverage provided by an architecture, allowing for the direct identification of vulnerabilities in the evolution of architecture coverage. As shown in the results, it can be difficult to extend a volume coverage metric to account for how spacecraft in Cislunar space actually operate through the regime. Our framework provides a systematic way to quantify this effect, allowing for an intuitive visualization for mission designers and decision makers in Cislunar space.

Chapter 7

Conclusions and Future Work

This thesis represents an indepth exploration of the preliminary aspects of Cislunar SSA architecture design and analysis. Included in this work is a novel approach for uncovering Pareto-optimal sets of distributed electro-optcial observers using a multi-objective optimization framework, optimizing over a variety of metrics. This includes the modeling of electro-optical sensors returning detection of space objects using an SNR threshold, and the consideration of how future architectures will inevitably be leveraged by mission planners to support a variety of mission objectives. Furthermore, a focus of this thesis was on the development of novel tools that can aid in the optimization of architectures, and also be used in mission planning with an established architecture.

7.1 Optimal Architecture Design Using Monte Carlo Tree Search

In this contribution, we leveraged multi-objective Monte Carlo Tree Search to explore the design space of distributed electro-optical SSA architectures. We demonstrated that this approach can be used to uncover Pareto-optimal sets of architectures, and that these architectures can be used to support a variety of mission objectives. Furthermore, by assessing detectability using an SNR threshold, we were able to provide a more realistic assessment of the performance of these architectures while simultaneously considering the nonlinear increase of cost associated with increasing observer capabilities. This work laid the foundation and future direction of the proceeding contributions, and significantly contributed to the study of distributed architecture optimization for Cislunar SSA.

7.2 Architecture Design for Cooperative Agents

In this contribution, we introduced a novel metric to the architecture design problem that simulated the concept of the cost incurred by a cooperative agent to use a distributed architecture in support of a mission objective. This metric was used as a third objective in the multi-objective optimization framework as a way to evaluate how cost is distributed between an architecture and a cooperative agent using the architecture. This work was the first to consider the cost through control effort of using an architecture in support of a mission objective, and demonstrated that in order for a cooperative agent to remain detectable above a certain threshold, either the architecture must generally be more expensive, or the cooperative agent must expend more control effort to remain detectable. In this work, we also leveraged clustering techniques across the resultant state space histories to identify patterns in observer placement and resultant impact on Pareto-optimality. Through this contribution, we offered a new and important perspective for the optimal design of distributed architectures, and how they must inevitably consider how real world actors will use them in support of mission objectives.

7.3 Persistent Detection Corridors for Crewed Missions

In this contribution, we explored insights from the previous contribution - namely, the concept of corridors within Cislunar space along which a trajectory can reamin persistently detectable by a distributed architecture - termed persistent detection corridors. To uncover these corridors, we first captured the evolution of an architecture's coverage over time by encoding detectable regions within a graph, with nodes representing detectable points in space, connected forward in time to nearby nodes. Then, the A* path finding algorithm was used to identify feasible paths through the graph, representing approximate natural trajectories traversing primarily through detectable regions of Cislunar space. To ensure the paths were truly dynamically feasible, we used pseudospectral collocation to return a controlled trajectory near the path, and then evaluated the local neighborhood of detection about the trajectory to uncover the coverage about the trajectory. The entire process was automated to allow for the rapid exploration of a large number of paths through the graph allowing for the recovery of various persistent detection corridors. This work introduced persistent detection corridors as a new concept in Cislunar SSA and demonstrated a rapid and automated approach to uncovering them for any arbitrary distributed architecture by encoding the evolution of coverage within an architecture's field-of-regard. Furthermore, this contribution demonstrated how these persistent detection corridors can be used to support the planning of crewed missions through Cislunar space, or other missions where persistent detection is paramount to mission success. Our novel approach here can also be used within an architecture design framework such as the previous two contributions, allowing for the design of architectures that ensures the frequent occurance of low-cost persistent detection corridors for Cislunar missions.

7.4 Resilience of Architectures Using Reachability

, In this final contribution, we explored how the distribution of coverage provided by a distributed architecture evolves over time within Cislunar space, and how well this evolution matches that of a controled spacecraft. We introduced the concept of architecture resilience to disturbances in the spacecraft being tracked by an architecture, i.e. loss of control authority in a crewed mission in Cislunar space. We first generated a massive dataset of reachable sets within Cislunar space with initial positions distributed about the region, and velocity magnitudes sampled according to a range of Jacobi constant values. Then, the coverage of all reachable sets and their intermediate states was evaluated for detectability by a distributed architecture, resulting in a heatmap - called a resilience map - that visualizes how well an architecture's evolution of coverage actually matches the evolution of low-thrust spacecraft dynamics. We showed how our novel method encodes a measure of resilience within distributed architectures, and in particular how limited a typical volume coverage metric is for determing the quality of an architecture's coverage distribution. Furthermore, we demonstrated how, by using resilience maps, mission planners and SSA decision makers can quickly identify vulnerabilities in an architecture and take steps to allocate resources to alleviate them. Our resilience maps can also be used to inform sensor tasking for new object discovery, as they can be interpretted in a probabilistic sense, showing the proportion of all potential spacecraft trajectories eminating from localized region that an architecture can detect. In summary, this contribution presents another novel tool and metric that can be used in both Cislunar SSA architecture optimization and during real world mission planning and sensor tasking.

7.5 Potential Future Work

There are a number of potential future directions for this work that could be explored with the goal of improving the design and analysis of Cislunar SSA architectures. These include:

- Optimal sequential construction of distributed architectures using MO-MCTS:
  - ∗ In the first contribution, we demonstrated that MO-MCTS can be used to uncover Pareto-optimal sets of distributed architectures. However, this approach could be extended to consider the sequential construction of architectures as will likely be the case in the real world. This would involve considering the proper order of adding observers to an architecture over a long tim horizon, and considerations for modularity and reconfigurability of the architecture as new observers are added. The architecture design problem would be formulated as a sequential decision making process and use MO-MCTS in the traditional sense to explore the design space.
- Identification of persistent detection corridors with reachability analysis:
  - ∗ Briefly mentioned within the third contribution was the idea that the width of a persistent detection corridor could be chosen to represent a sort of safety region about the reference trajectory, within which mission planners might expect that a slight deviation would remain detectable. Reachability theory can be leveraged to explore this concept, identifying a corridor radius at each time step that represents what a deviation from the reference trajectory would look like, and how well the rechable set is covered by the distributed architecture.

- Resilience maps informing observer tasking and probability of detection:

  * The resilience maps generated in the fourth contribution were done so by using an architecture's field-of-regard, which, while useful for high level analysis of performance, lacks the lower-level specificity of individual sensor tasking using individual observers' field-of-view (FoV). There is a lot of potential for leveraging these maps within an optimal sensor tasking framework, as our resilience maps describe the ability of an architecture to continually detect low-thrust spacecraft across the Cislunar regime. This work would include using these maps to directly inform sensor tasking for new object discovery and track maintenance.

- Optimal distributed architecture design with resilience, persistent detection corridor metrics:

  * The MO-MCTS/NSGA-II hybrid algorithm can be used in a similar manner to the second contribution for identifying Pareto-optimal architectures but with additional metrics included from the third and fourth contributions. A resilience metric can be included quite easily by averaging the coverage of all reachable sets over time, allowing for an anlysis of how resilience correlates with other metrics such as volume coverage. Futhermore, our tool for uncovering detection corridors can be similarly easily implemented within this framework and can be included as an objective where it is desirable to have as many persistent detection corridors as possible over some minimum threshold.

Bibliography

[1] Challenges and Potential in Space Domain Awareness. Journal of Guidance, Control, and Dynamics, 41(1):15–18, January 2018.

[2] National Cislunar Science and Technology Strategy. Technical report, CISLUNAR TECHNOLOGY STRATEGY INTERAGENCY WORKING GROUP, National Science and Technology Council, November 2022.

[3] 2024 Space Safety Compendium - Collaborating for Sustainable Space Futures. Technical report, The Aerospace Corporation - Space Safety Institute, 2024.

[4] Andrew M Annex, Ben Pearson, Benoît Seignovert, Brian T Carcich, Helge Eichhorn, Jesse A Mapel, Johan L Freiherr Von Forstner, Jonathan McAuliffe, Jorge Diaz Del Rio, Kristin L Berry, et al. Spiceypy: A pythonic wrapper for the spice toolkit. Journal of Open Source Software, 5(46):2050, 2020.

[5] T Antony, M Grant, and M Sparapany. Beluga.

[6] Gregory Badura, Yuri Shimane, Alaric Gregoire, Rohan Patel, Matthew Gilmartin, Kunal Gangolli, Lois Visonneau, Joshua Tysor, Saikrishna Manojkumar, Chris Valenta, Reilly Blair, Nelson Lourenco, Jason Hodkin, Mariel Borowitz, Brian Gunter, John Christian, and Koki Ho. SYSTEM DESIGN AND ANALYSIS FOR CISLUNAR SPACE DOMAIN AWARENESS THROUGH DISTRIBUTED SENSORS. 2022.

[7] Fangyun Bai, Xinglong Ju, Shouyi Wang, Wenyong Zhou, and Feng Liu. Wind farm layout optimization using adaptive evolutionary algorithm with Monte Carlo Tree Search reinforcement learning. Energy Conversion and Management, 252:115047, January 2022.

[8] Brian Baker-McEvilly, Surabhi Bhadauria, David Canales, and Carolin Frueh. A Comprehensive Review on Cislunar Expansion and Space Domain Awareness. Progress in Aerospace Sciences, 147:101019, May 2024.

[9] Surabhi Bhadauria, Arly Black, and Carolin Frueh. Cislunar Surveillance Optimization and Key Region Identification. In Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS), 2024.

[10] Surabhi Bhadauria and Carolin Frueh. Optical Observation Regions in Cislunar Space Using the Bi-Circular Restricted Four Body Problem Geometry. In Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS), 2022.

[11] Mark Bolden, Timothy Craychee, and Erin Griggs. An Evaluation of Observing Constellation Orbit Stability, Low Signal-to-Noise, and the Too-Short-Arc Challenges in the Cislunar Domain. In Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS), 2020.

[12] Béla Bollobás. Modern Graph Theory, volume 184 of Graduate Texts in Mathematics. Springer New York, New York, NY, 1998.

[13] Mariel Borowitz, Althea Noonan, and Reem El Ghazal. U.S. strategic interest in the moon: An assessment of economic, national security, and geopolitical drivers. Space Policy, 69:101548, 2024.

[14] Natasha Bosanac. Data-Mining Approach to Poincaré Maps in Multi-Body Trajectory Design. Journal of Guidance, Control, and Dynamics, 43(6):1190–1200, June 2020.

[15] Natasha Bosanac. DATA-DRIVEN SUMMARY OF NATURAL SPACECRAFT TRAJECTORIES IN THE EARTH-MOON SYSTEM. In AAS/AIAA Astrodynamics Specialist Conference, Big Sky, MT, USA, 2023.

[16] Natasha Bosanac and Maxwell Joyner. DATA-DRIVEN SUMMARY OF CONTINUOUS THRUST TRAJECTORIES IN A LOW-FIDELITY MODEL OF CISLUNAR SPACE. In 2024 AAS/AIAA Astrodynamics Specialist Conference, Broomfield, CO, 2024.

[17] Sean Bowerfind and Ehsan Taheri. Rapid Approximation of Low-Thrust Spacecraft Reachable Sets within Complex Two-Body and Cislunar Dynamics. Aerospace, 11(5):380, May 2024.

[18] Julian Brew. USING SAMPLE-BASED CONTINUATION TECHNIQUES TO EFFICIENTLY COMPUT PhD thesis, Georgia Institute of Technology, 2019.

[19] Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A Survey of Monte Carlo Tree Search Methods. IEEE Transactions on Computational Intelligence and AI in Games, 4(1):1–43, March 2012.

[20] Kristen L Bruchko and Natasha Bosanac. A PRELIMINARY EXPLORATION OF PATH PLANNING FOR INITIAL GUESS CONSTRUCTION IN MULTI-BODY SYSTEMS. In AAS/AIAA Astrodynamics Specialist Conference, 2021.

[21] Kristen L Bruchko and Natasha Bosanac. ADAPTIVE ROADMAP GENERATION FOR TRAJECTORY DESIGN IN THE EARTH-MOON SYSTEM. In AAS/AIAA Space Flight Mechanics Meeting, Austin, TX, 2023.

[22] Kristen L Bruchko and Natasha Bosanac. RAPID TRAJECTORY DESIGN IN MULTI-BODY SYSTEMS USING SAMPLING-BASED KINODYNAMIC PLANNING. In AAS/AIAA Astrodynamics Specialist Conference, Big Sky, Montana, 2023.

[23] Guillaume M. J-B. Chaslot, Mark H. M. Winands, H. Jaap Van Den Herik, Jos W. H. M. Uiterwijk, and Bruno Bouzy. PROGRESSIVE STRATEGIES FOR MONTE-CARLO TREE SEARCH. New Mathematics and Natural Computation, 04(03):343–357, November 2008.

[24] T Henry Clareson, Matthew C Fox, Dominic Amato, and Hang Woon Lee. OPTIMIZATION FRAMEWORK FOR SPACE-BASED MULTI-SENSOR SYSTEMS IN CISLUNAR SPACE DOMAIN AWARENESS. In AAS/AIAA Astrodynamics Specialist Conference, 2023.

[25] Ryan D. Coder and Marcus J. Holzinger. Multi-objective design of optical systems for space situational awareness. Acta Astronautica, 128:669–684, November 2016.

[26] Adrien Couëtoux, Jean-Baptiste Hoock, Nataliya Sokolovska, Olivier Teytaud, and Nicolas Bonnard. Continuous Upper Confidence Trees. In Carlos A. Coello Coello, editor, Learning and Intelligent Optimization, Lecture Notes in Computer Science, pages 433–445, Berlin, Heidelberg, 2011. Springer.

[27] T. Cover and P. Hart. Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1):21–27, January 1967.

[28] Jacob A Dahlke, Robert A Bettinger, and Rachel Oliver. Enhanced Heuristic Algorithm for Optimal Cislunar Space Situational Awareness Architectures. In The Advanced Maui Optical and Space Surveillance Technologies Conferences, 2024.

[29] Jacob A Dahlke, Adam P Wilmer, and Robert A Bettinger. PRELIMINARY COMPARATIVE ASSESSMENT OF L2 AND L3 SURVEILLANCE USING SELECT CISLUNAR PERIODIC ORBITS. In AAS/AIAA Astrodynamics Specialist Conference, 2022.

[30] Ashwati Das-Stuart, Kathleen Howell, and David C Folta. A rapid trajectory design strategy for complex environments leveraging attainable regions and low-thrust capabilities. In 68th International Astronautical Congress, 2017.

[31] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2):182–197, April 2002.

[32] E. W. Dijkstra. A note on two problems in connexion with graphs. Numerische Mathematik, 1(1):269–271, December 1959.

[33] Bruce Donald, Patrick Xavier, John Canny, and John Reif. Kinodynamic Motion Planning. Journal of the Association for Computing Machinery, 1993.

[34] Naomi Owens Fahrner, Jeremy Correa, and Joshua Wysack. Capacity-based Cislunar Space Domain Awareness Architecture Optimization. In Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS), 2022.

[35] Samuel J. Fedeler and this link will open in a new tab Link to external site. Optical Sensor Tasking Using Monte Carlo Tree Search. PhD thesis, University of Colorado at Boulder, United States – Colorado, 2023.

[36] David Folta, Natasha Bosanac, Ian Elliott, Laurie Mann, Rebecca Mesarch, and Jose Rosales. Astrodynamics Convention and Modeling Reference for Lunar, Cislunar, and Libration Point Orbits, October 2022.

[37] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP: Evolutionary algorithms made easy. Journal of Machine Learning Research, 13:2171–2175, jul 2012.

[38] Erin E Fowler, Stella B Hurtt, and Derek A Paley. Orbit Design for Cislunar Space Domain Awareness. In 2nd IAA Conference on Space Situational Awareness (ICSSA), 2020.

[39] Erin E. Fowler and Derek A. Paley. Observability Metrics for Space-Based Cislunar Domain Awareness. The Journal of the Astronautical Sciences, 70(2):10, March 2023.

[40] C Frueh, K Howell, K J DeMars, S Bhadauria, and M Gupta. CISLUNAR SPACE TRAFFIC MANAGEMENT: SURVEILLANCE THROUGH EARTH-MOON RESONANCE ORBITS. In 8th European Conference on Space Debris, 2021.

[41] Carolin Frueh, Kathleen Howell, Kyle J DeMars, and Surabhi Bhadauria. CISLUNAR SPACE SITUATIONAL AWARENESS. In 31st AIAA/AAS Space Flight Mechanics Meeting, 2021.

[42] Divya Garg, Michael A. Patterson, Camila Francolin, Christopher L. Darby, Geoffrey T. Huntington, William W. Hager, and Anil V. Rao. Direct trajectory optimization and costate estimation of finite-horizon and infinite-horizon optimal control problems using a Radau pseudospectral method. Computational Optimization and Applications, 49(2):335–358, June 2011.

[43] Eric Gorman, Collin Deans, Mohamed Nassif, and Elizabeth Frank. QuantumNet: A scalable cislunar space domain awareness constellation. In Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference, 2023.

[44] Namrata Goswami and Bleddyn Bowen. HIGH GROUND OR HIGH FANTASY: DEFENSE UTILITY OF CISLUNAR SPACE. Technical report, The Aerospace Corporation - Center for Space Policy and Strategy, 2024.

[45] Jesse A Greaves and Daniel J Scheeres. Relative Estimation in the Cislunar Regime using Optical Sensors. In The Advanced Maui Optical and Space Surveillance Technologies Conference, 2021.

[46] Maaninee Gupta and Kathleen C Howell. CISLUNAR ECLIPSE MITIGATION STRATEGIES FOR RESONANT PERIODIC ORBITS. In AAS/AIAA Astrodynamics Specialist Conference, 2023.

[47] Maaninee Gupta, Kathleen C Howell, and Carolin Frueh. LONG-TERM CISLUNAR SURVEILLANCE VIA MULTI-BODY RESONANT TRAJECTORIES. In AAS/AIAA Astrodynamics Specialist Conference, 2022.

[48] Maaninee Gupta, Kathleen C Howell, and Carolin Frueh. CONSTELLATION DESIGN TO SUPPORT CISLUNAR SURVEILLANCE LEVERAGING SIDEREAL RESONANT ORBITS. In 33rd AAS/AIAA Space Flight Mechanics Meeting, Austin, Texas, 2023.

[49] Aric Hagberg, Pieter J. Swart, and Daniel A. Schult. Exploring network structure, dynamics, and function using NetworkX. Technical Report LA-UR-08-05495; LA-UR-08-5495, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), January 2008.

[50] Zach Hall and Puneet Singla. A PROBABILISTIC APPROACH FOR REACHABILITY SET COMPUTATION FOR EFFICIENT SPACE SITUATIONAL AWARENESS.

[51] Zach Hall and Puneet Singla. A PROBABILISTIC APPROACH FOR REACHABILITY SET COMPUTATION FOR EFFICIENT SPACE SITUATIONAL AWARENESS. In 29th AAS/AIAA Space Flight Mechanics Meeting, 2019.

[52] Zach Hall and Puneet Singla. Reachability Analysis Based Tracking: Applications to Non-cooperative Space Object Tracking. In Frederica Darema, Erik Blasch, Sai Ravela, and Alex Aved, editors, Dynamic Data Driven Applications Systems, volume 12312, pages 200–207. Springer International Publishing, Cham, 2020.

[53] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics, 4(2):100–107, July 1968.

[54] Jason Hay, Patrick Craven, Benjamin Merrel, Phillip Williams, and Grace Wusk. Resiliency in Future Cislunar Space Architectures. Technical report, December 2022.

[55] Daniel Hennes and Dario Izzo. Interplanetary trajectory planning with Monte Carlo tree search. In Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI'15, pages 769–775, Buenos Aires, Argentina, July 2015. AAAI Press.

[56] M J Holzinger, C C Chow, and P Garretson. A Primer on Cislunar Space. Technical report, Air Force Research Laboratory, 2021.

[57] Marcus J. Holzinger and Daniel J. Scheeres. Reachability Results for Nonlinear Systems with Ellipsoidal Initial Sets. IEEE Transactions on Aerospace and Electronic Systems, 48(2):1583–1600, 2012.

[58] Xi Huang, Ziyu Shao, and Yang Yang. MIPS: Instance Placement for Stream Processing Systems Based on Monte Carlo Tree Search. In ICC 2019 - 2019 IEEE International Conference on Communications (ICC), pages 1–6, Shanghai, China, May 2019. IEEE.

[59] Matthew Kelly. An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation. SIAM Review, 59(4):849–904, January 2017.

[60] M. Klonowski, N. O. Fahrner, C. Heidrich, and M. Holzinger. Robust Cislunar Architecture Design Optimization for Cooperative Agents. In Proceedings of the Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference, page 15, September 2023.

[61] Michael Klonowski, Casey Heidrich, Naomi Owens-Fahrner, and Marcus J Holzinger. Analysis of Persistent Detection Corridors for Cislunar Space Situational Awareness. In The Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference, 2024.

[62] Michael Klonowski, Marcus J. Holzinger, and Naomi Owens Fahrner. Optimal Cislunar Architecture Design Using Monte Carlo Tree Search Methods. In The Advanced Maui Optical and Space Surveillance Conference, volume 70, page 17, September 2022.

[63] Michael Klonowski, Marcus J. Holzinger, and Naomi Owens Fahrner. Optimal Cislunar Architecture Design Using Monte Carlo Tree Search Methods. The Journal of the Astronautical Sciences, 70(3):17, June 2023.

[64] Michael Klonowski, Naomi Owens-Fahrner, Casey Heidrich, and Marcus Holzinger. Cislunar space domain awareness architecture design and analysis for cooperative agents. The Journal of the Astronautical Sciences, 71(5):47, 2024.

[65] Simon R Knister. Evaluation Framework for Cislunar Space Domain Awareness (SDA) Systems. PhD thesis, Air Force Institute of Technology, 2020.

[66] Mykel J. Kochenderfer, Tim A. Wheeler, and Kyle H. Wray. Algorithms for Decision Making. MIT Press, August 2022.

[67] Levente Kocsis and Csaba Szepesvári. Bandit Based Monte-Carlo Planning. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, Machine Learning: ECML 2006, Lecture Notes in Computer Science, pages 282–293, Berlin, Heidelberg, 2006. Springer.

[68] J. P. Laumond and M. Thoma, editors. Robot Motion Planning and Control, volume 229 of Lecture Notes in Control and Information Sciences. Springer, Berlin, Heidelberg, 1998.

[69] Steven LaValle. Rapidly-exploring random trees: A new tool for path planning. Research Report 9811, 1998.

[70] R.T. Marler and J.S. Arora. Survey of multi-objective optimization methods for engineering. Structural and Multidisciplinary Optimization, 26(6):369–395, April 2004.

[71] Dimitri N. Mavris, Bradford E. Robertson, Michael Balchanos, and Mark Whorton. The Georgia Tech Cislunar Architecture Study. In ASCEND 2022, Las Vegas, Nevada & Online, October 2022. American Institute of Aeronautics and Astronautics.

[72] Robert J. Moss, Anthony Corso, Jef Caers, and Mykel J. Kochenderfer. BetaZero: Belief-State Planning for Long-Horizon POMDPs using Learned Approximations, June 2023.

[73] Joaquim Ortiz-Haro, Wolfgang Hönig, Valentin N. Hartmann, Marc Toussaint, and Ludovic Righetti. iDb-RRT: Sampling-based Kinodynamic Motion Planning with Motion Primitives and Trajectory Optimization, March 2024.

[74] Hae-Sang Park and Chi-Hyuck Jun. A simple and fast algorithm for K-medoids clustering. 36:3336–3341.

[75] Jeffrey S Parker and Rodney L Anderson. Low-energy lunar trajectory design, volume 12. John Wiley & Sons, 2014.

[76] Steve Parr. Cislunar Security National Technical Vision. Technical report, Applied Physics Laboratory, November 2022.

[77] Malav Patel, Yuri Shimane, Hang Woon Lee, and Koki Ho. Cislunar Satellite Constellation Design Via Integer Linear Programming, November 2023.

[78] Prashant R. Patel and Daniel J. Scheeres. Rapid and Automatic Reachability Estimation of Electric Propulsion Spacecraft. The Journal of the Astronautical Sciences, 70(6):45, November 2023.

[79] Smriti Nandan Paul, Siwei Fan, and Igor Panfil. Advancing cislunar space domain awareness through robust optimization framework for optical sensors-based autonomous satellite systems. In AIAA SCITECH 2025 Forum, page 1218, 2025.

[80] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

[81] Daniel Pipkins. Randomized Kinodynamic Planning. The International Journal of Robotics Research, 2001.

[82] Nery Riquelme, Christian Von Lucken, and Benjamin Baran. Performance metrics in multi-objective optimization. In 2015 Latin American Computing Conference (CLEI), pages 1–11, Arequipa, Peru, October 2015. IEEE.

[83] Leonardo Rossi, Mark H. M. Winands, and Christoph Butenweg. Monte Carlo Tree Search as an intelligent search tool in structural design problems. Engineering with Computers, 38(4):3219–3236, August 2022.

[84] Maarten P. D. Schadd, Mark H. M. Winands, H. Jaap Van Den Herik, Guillaume M. J. B. Chaslot, and Jos W. H. M. Uiterwijk. Single-Player Monte-Carlo Tree Search. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, H. Jaap Van Den Herik, Xinhe Xu, Zongmin Ma, and Mark H. M. Winands, editors, Computers and Games, volume 5131, pages 1–12. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[85] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering Atari, Go, chess and shogi by planning with a learned model. Nature, 588(7839):604–609, December 2020.

[86] David Schwab, Roshan Eapen, and Puneet Singla. Reachability Analysis to Track Non-cooperative Satellite in Cislunar Regime. In Erik Blasch, Frederica Darema, and Alex Aved, editors, Dynamic Data Driven Applications Systems, volume 13984, pages 105–113. Springer Nature Switzerland, Cham, 2024.

[87] Mushfeq-Us-Saleheen Shameem and Raihana Ferdous. An efficient k-means algorithm integrated with Jaccard distance measure for document clustering. In 2009 First Asian Himalayas International Conference on Internet, pages 1–6.

[88] Yuri Shimane, Kento Tomita, and Koki Ho. Strategic Regions for Monitoring Incoming Low-Energy Transfers to Low-Lunar Orbits. In Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS), 2023.

[89] Yuri Shimane, Kento Tomita, and Koki Ho. Cislunar Space Situational Awareness Constellation Design and Planning with Facility Location Problem. Journal of Spacecraft and Rockets, pages 1–23, June 2025.

[90] Marshall Smith, Douglas Craig, Nicole Herrmann, Erin Mahoney, Jonathan Krezel, Nate McIntyre, and Kandyce Goodliff. The Artemis Program: An Overview of NASA's Activities to Return Humans to the Moon. In 2020 IEEE Aerospace Conference, pages 1–10, March 2020.

[91] Thomas Raymond Smith. Using Motion Primitives to Rapidly Design Trajectories in Multi-Body Systems. PhD thesis, University of Colorado at Boulder, Boulder, CO, 2023.

[92] Maciej Świechowski, Konrad Godlewski, Bartosz Sawicki, and Jacek Mańdziuk. Monte Carlo Tree Search: A review of recent modifications and applications. Artificial Intelligence Review, 56(3):2497–2562, March 2023.

[93] James A. Vedda. Cislunar Development: What to Build - and Why? Technical report, The Aerospace Corporation, April 2018.

[94] Jacob K. Vendl and Marcus J. Holzinger. Cislunar Periodic Orbit Analysis for Persistent Space Object Detection Capability. Journal of Spacecraft and Rockets, 58(4):1174–1185, July 2021.

[95] Shanu Verma, Millie Pant, and Vaclav Snasel. A Comprehensive Review on NSGA-II for Multi-Objective Combinatorial Optimization Problems. IEEE Access, 9:57757–57791, 2021.

[96] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17:261–272, 2020.

[97] Lois Visonneau, Yuri Shimane, and Koki Ho. Optimizing Multi-Spacecraft Cislunar Space Domain Awareness Systems via Hidden-Genes Genetic Algorithm, February 2023.

[98] Weijia Wang and Michèle Sebag. Multi-objective Monte-Carlo Tree Search. In Proceedings of the Asian Conference on Machine Learning, pages 507–522. PMLR, November 2012.

[99] Weijia Wang and Michèle Sebag. Hypervolume indicator and dominance reward based multi-objective Monte-Carlo Tree Search. Machine Learning, 92(2):403–429, September 2013.

[100] Dustin J. Webb and Jur van den Berg. Kinodynamic RRT*: Optimal Motion Planning for Systems with Linear Differential Constraints, May 2012.

[101] Ryan Whitley and Roland Martinez. Options for staging orbits in cislunar space. In 2016 IEEE Aerospace Conference, pages 1–9, Big Sky, MT, USA, March 2016. IEEE.

[102] Adam P. Wilmer, Robert A. Bettinger, and Bryan D. Little. Cislunar Periodic Orbits for Earth–Moon L1 and L2 Lagrange Point Surveillance. Journal of Spacecraft and Rockets, 59(6):1809–1820, 2022.

[103] Adam P Wilmer, Robert A Bettinger, Liberty M Shockley, and Marcus J Holzinger. Preliminary investigation and proposal of periodic orbits and their utilization for logistics in the cislunar regime. Space Policy, 71:101635, 2025.

[104] Raymond H Wright, Naomi Owens-Fahrner, Luke Tafur, and Joshua Wysack. Applications of Poincare´ Search Maps for Space-Based Cislunar SDA Detection. In Advanced Maui Optical and Space Surveillance Technologies Conference, 2024.

[105] Joshua Wysack and Kevin Ferrant. Contrasting Architectures for Cislunar SDA and STM. In Advanced Maui Optical and Space Surveillance Technologies Conference, 2023.

[106] Victor A. Yang, Jeffrey T. McNabb, Michael G. Balchanos, and Dimitri Mavris. Preliminary Approach to Resilient Cislunar Space Domain Awareness Architecture Design. In AIAA SCITECH 2025 Forum. American Institute of Aeronautics and Astronautics.

[107] Jin Y. Yen. Finding the K Shortest Loopless Paths in a Network. Management Science, July 1971.

Appendix A

Initial Conditions

Table A.1: CR3BP periodic orbit initial conditions.

Orbit Family	Orbit Index	Initial Conditions (DU)	Period (TU)	Jacobi Constant	Resonance (≈)
L5 Planar	0	0.904300, 0.866025, 0.000000, 0.175400, -0.546700, 0.000000	6.4836	2.8329	1:1
L5 Axial	0	0.813062, -0.123609, 0.100000, 0.083108, 0.385862, -0.237245	3.4015	2.9180	2:1
	1	0.813152, -0.123482, 0.100000, 0.083136, 0.385745, -0.237322	3.4004	2.9181	2:1
	2	0.929052, -0.002808, 0.100000, 0.129077, 0.133358, -0.426582	2.2683	2.9435	3:1
	3	0.929167, -0.002721, 0.100000, 0.129020, 0.132947, -0.426792	2.2676	2.9435	3:1
	4	0.929281, -0.002635, 0.100000, 0.128963, 0.132536, -0.427001	2.2670	2.9435	3:1
	5	0.929396, -0.002549, 0.100000, 0.128905, 0.132124, -0.427210	2.2664	2.9436	3:1
	6	0.716856, -0.246384, 0.100000, 0.120205, 0.488393, -0.214960	4.5348	2.8855	3:2
	7	0.651913, -0.295424, 0.100000, 0.198076, 0.549383, -0.246719	5.1021	2.8563	4:3
	8	0.652112, -0.295315, 0.100000, 0.197791, 0.549200, -0.246574	5.1006	2.8564	4:3
	9	0.871520, -0.051808, 0.100000, 0.116753, 0.291325, -0.314073	2.7217	2.9330	5:2
	10	0.871628, -0.051699, 0.100000, 0.116823, 0.291104, -0.314262	2.7206	2.9330	5:2
	11	0.758699, -0.199034, 0.100000, 0.089264, 0.447181, -0.211849	4.0815	2.9007	5:3
	12	0.603519, -0.315171, 0.100000, 0.275657, 0.592291, -0.291907	5.4425	2.8266	5:4
	13	0.603696, -0.315123, 0.100000, 0.275344, 0.592142, -0.291701	5.4414	2.8267	5:4
	14	0.566091, -0.321800, 0.100000, 0.347601, 0.621542, -0.344990	5.6687	2.7949	6:5
	15	0.566243, -0.321787, 0.100000, 0.347286, 0.621436, -0.344730	5.6679	2.7951	6:5
L2 Halo	0	1.174517, 0.000000, 0.074567, -0.000000, -0.181513, 0.000000	3.3685	3.1291	12
	1	1.073525, -0.000000, 0.201991, 0.000000, -0.190368, 0.000000	2.2409	3.0161	1:3
	2	1.103863, -0.000000, 0.197466, -0.000000, -0.218220, -0.000000	2.6890	3.0202	5:2
2:3 Resonant	0	0.876541, 0.000000, 0.000000, 0.000000, 0.603693, 0.000000	17.0058	2.8454	2:5
	1	0.876607, 0.000000, 0.000000, 0.000000, 0.603720, 0.000000	17.0053	2.8454	2:5
	2	0.876672, 0.000000, 0.000000, 0.000000, 0.603747, 0.000000	17.0049	2.8455	2:5
	3	0.876737, 0.000000, 0.000000, -0.000000, 0.603774, 0.000000	17.0044	2.8455	2:5
L1 Axial	0	0.930600, 0.000000, 0.000000, -0.000000, -0.603968, -0.000732	3.9500	3.0214	5:3
1:2 Resonant	0	0.840523, 0.000000, 0.000000, 0.000000, 0.640944, 0.000000	11.3376	2.7777	3:4
	1	0.840614, 0.000000, 0.000000, 0.000000, 0.640894, 0.000000	11.3370	2.7778	3:4
	2	0.840704, 0.000000, 0.000000, 0.000000, 0.640844, 0.000000	11.3365	2.7778	3:4
	3	0.840794, 0.000000, 0.000000, 0.000000, 0.640794, 0.000000	11.3360	2.7779	3:4
1:3 Resonant	0	3.226844, 0.000000, 0.000000, -0.000000, -2.862563, 0.000000	17.0042	2.8391	2:5
	1	3.226849, 0.000000, 0.000000, -0.000000, -2.862566, 0.000000	17.0045	2.8391	2:5
	2	3.226854, 0.000000, 0.000000, -0.000000, -2.862570, 0.000000	17.0047	2.8391	2:5
	3	3.226859, 0.000000, 0.000000, -0.000000, -2.862574, 0.000000	17.0050	2.8391	2:5
	4	3.226864, 0.000000, 0.000000, -0.000000, -2.862578, 0.000000	17.0052	2.8391	2:5
	5	3.226869, 0.000000, 0.000000, -0.000000, -2.862582, 0.000000	17.0055	2.8391	2:5
	6	3.226874, 0.000000, 0.000000, -0.000000, -2.862586, 0.000000	17.0057	2.8391	2:5
	7	3.226879, 0.000000, 0.000000, -0.000000, -2.862590, 0.000000	17.0060	2.8391	2:5
L1 Vertical	0	0.931375, -0.000000, 0.286205, -0.000000, -0.069016, -0.000000	4.5183	2.9498	3:2
	1	0.926430, 0.000000, 0.359816, -0.000000, -0.054942, -0.000000	5.0840	2.8873	4:3
	2	0.922824, 0.000000, 0.241836, -0.000000, -0.065649, -0.000000	4.0832	2.9901	5:3
	3	0.908356, 0.000000, 0.423334, 0.000000, -0.029214, -0.000000	5.4178	2.8307	5:4
	4	0.883182, -0.000000, 0.482350, -0.000000, 0.002472, 0.000000	5.6364	2.7719	6:5
L1 Lyapunov	0	0.619269, 0.000000, -0.000000, -0.000000, 0.826471, 0.000000	6.7792	2.8953	1:1
	1	0.798246, 0.000000, -0.000000, 0.000000, 0.368685, 0.000000	3.3857	3.0674	2:1
	2	0.762301, -0.000000, 0.000000, -0.000000, 0.498318, -0.000000	4.5136	2.9916	3:2
	3	0.739605, 0.000000, 0.000000, -0.000000, 0.551442, -0.000000	5.0747	2.9689	4:3
	4	0.777494, 0.000000, -0.000000, -0.000000, 0.456401, 0.000000	4.0769	3.0137	5:3
	5	0.723690, 0.000000, 0.000000, 0.000000, 0.586512, -0.000000	5.4119	2.9567	5:4
	6	0.711636, -0.000000, 0.000000, -0.000000, 0.612846, -0.000000	5.6398	2.9485	6:5
L4 Axial	0	0.813062, 0.123609, 0.100000, -0.083108, 0.385862, 0.237245	3.4015	2.9180	2:1
	1	0.813152, 0.123482, 0.100000, -0.083136, 0.385745, 0.237322	3.4004	2.9181	2:1
	2	0.929052, 0.002808, 0.100000, -0.129077, 0.133358, 0.426582	2.2683	2.9435	3:1
	3	0.929167, 0.002721, 0.100000, -0.129020, 0.132947, 0.426792	2.2676	2.9435	3:1
	4	0.929281, 0.002635, 0.100000, -0.128963, 0.132536, 0.427001	2.2670	2.9435	3:1
	5	0.929396, 0.002549, 0.100000, -0.128905, 0.132124, 0.427210	2.2664	2.9436	3:1
	6	0.716856, 0.246384, 0.100000, -0.120205, 0.488393, 0.214960	4.5348	2.8855	3:2
	7	0.651913, 0.295424, 0.100000, -0.198076, 0.549383, 0.246719	5.1021	2.8563	4:3
	8	0.652112, 0.295315, 0.100000, -0.197791, 0.549200, 0.246574	5.1006	2.8564	4:3
	9	0.871520, 0.051808, 0.100000, -0.116753, 0.291325, 0.314073	2.7217	2.9330	5:2
	10	0.871628, 0.051699, 0.100000, -0.116823, 0.291104, 0.314262	2.7206	2.9330	5:2
	11	0.758699, 0.199034, 0.100000, -0.089264, 0.447181, 0.211849	4.0815	2.9007	5:3
	12	0.603519, 0.315171, 0.100000, -0.275657, 0.592291, 0.291907	5.4425	2.8266	5:4
	13	0.603696, 0.315123, 0.100000, -0.275344, 0.592142, 0.291701	5.4414	2.8267	5:4
	14	0.566091, 0.321800, 0.100000, -0.347601, 0.621542, 0.344990	5.6687	2.7949	6:5
	15	0.566243, 0.321787, 0.100000, -0.347286, 0.621436, 0.344730	5.6679	2.7951	6:5
L2 Axial	0	1.219977, -0.000000, 0.000000, 0.000000, -0.427494, -0.000363	4.3105	3.0138	3:2
GEO	0	-0.012151, 0.109688, 0.000000, 3.001001, 0.000000, 0.000000	0.1991	9.0423	30:1

Table A.1: CR3BP periodic orbit initial conditions (continued).

Orbit Family	Orbit Index	Initial Conditions (DU)	Period (TU)	Jacobi Constant	Resonance (≈)
S. Butterfly	0	0.997458, 0.000000, -0.180887, 0.000000, -0.267825, -0.000000	6.8018	2.9836	1:1
	1	0.909486, 0.000000, -0.147987, 0.000000, -0.029831, -0.000000	3.4014	3.0880	2:1
	2	1.029694, 0.000000, -0.319069, 0.000000, -0.257194, -0.000000	10.2035	2.8829	2:3
	3	0.968194, 0.000000, -0.462906, 0.000000, -0.161387, 0.000000	11.3371	2.7862	3:4
	4	0.906875, 0.000000, -0.143163, -0.000000, -0.080985, -0.000000	4.0817	3.0878	5:3
	5	0.947594, 0.000000, -0.157279, -0.000000, -0.202470, -0.000000	5.4420	3.0381	5:4
	6	0.956277, 0.000000, -0.160945, 0.000000, -0.217203, -0.000000	5.6687	3.0280	6:5
S. Dragonfly	0	1.142788, 0.000000, -0.104701, 0.009960, -0.336817, -0.263838	6.8024	2.9564	1:1
	1	1.116741, 0.000000, -0.174454, -0.009210, -0.246609, -0.119101	5.6693	3.0137	6:5
	2	1.116716, 0.000000, -0.174519, -0.009212, -0.246526, -0.118913	5.6687	3.0137	6:5
	3	1.116690, 0.000000, -0.174584, -0.009214, -0.246442, -0.118724	5.6681	3.0137	6:5
	4	1.116665, 0.000000, -0.174649, -0.009215, -0.246359, -0.118535	5.6676	3.0138	6:5
Dist. Pro.	0	1.004196, 0.000000, 0.000000, 0.000000, 1.204892, 0.000000	6.8026	2.9872	1:1
	1	1.027007, 0.000000, 0.000000, -0.000000, 0.728667, 0.000000	4.5338	3.0456	3:2
	2	0.993576, 0.000000, 0.000000, 0.000000, 2.060847, 0.000000	9.0697	2.9485	3:3
	3	1.019261, 0.000000, 0.000000, 0.000000, 0.837234, 0.000000	5.1008	3.0271	4:3
	4	0.995685, 0.000000, 0.000000, 0.000000, 1.759259, 0.000000	8.5028	2.9582	4:5
	5	1.059033, 0.000000, 0.000000, 0.000000, 0.429152, 0.000000	2.7207	3.1232	5:2
	6	1.034501, 0.000000, 0.000000, -0.000000, 0.644767, 0.000000	4.0806	3.0630	5:3
	7	1.015401, 0.000000, 0.000000, 0.000000, 0.904593, 0.000000	5.4409	3.0175	5:4
	8	0.997091, 0.000000, 0.000000, 0.000000, 1.617872, 0.000000	8.1625	2.9638	5:6
	9	1.045401, 0.000000, 0.000000, 0.000000, 0.432515, 0.000000	1.1347	3.1962	6:1
	10	1.045437, 0.000000, 0.000000, 0.000000, 0.432143, 0.000000	1.1337	3.1963	6:1
	11	1.045473, 0.000000, 0.000000, 0.000000, 0.431774, 0.000000	1.1327	3.1964	6:1
	12	1.013100, 0.000000, 0.000000, -0.000000, 0.950893, 0.000000	5.6686	3.0116	6:5
N. Dragonfly	0	1.142788, 0.000000, 0.104701, 0.009960, -0.336817, 0.263838	6.8024	2.9564	1:1
	1	1.116741, 0.000000, 0.174454, -0.009210, -0.246609, 0.119101	5.6693	3.0137	6:5
	2	1.116716, 0.000000, 0.174519, -0.009212, -0.246526, 0.118913	5.6687	3.0137	6:5
	3	1.116690, 0.000000, 0.174584, -0.009214, -0.246442, 0.118724	5.6681	3.0137	6:5
	4	1.116665, 0.000000, 0.174649, -0.009215, -0.246359, 0.118535	5.6676	3.0138	6:5
L5 Long	0	0.508790, -0.866025, 0.000000, -0.013361, -0.009083, 0.000000	21.0757	2.9881	1:3
	1	0.832553, -0.866025, -0.000000, -0.155334, -0.321470, 0.000000	26.1775	2.9764	1:4
L2 Vertical	0	1.110697, -0.000000, -0.000000, 0.000000, -0.195148, 0.446795	4.5387	2.9533	3:2
	1	1.106768, -0.000000, -0.000000, 0.000000, -0.251012, 0.497475	5.0774	2.8845	4:3
	2	1.118303, 0.000000, -0.000000, 0.000000, -0.130095, 0.391505	4.0628	3.0144	5:3
	3	1.103870, 0.000000, -0.000000, 0.000000, -0.286534, 0.543842	5.4215	2.8204	5:4
	4	1.101012, 0.000000, -0.000000, 0.000000, -0.315109, 0.587259	5.6357	2.7577	6:5
L2 Lyapunov	0	0.996239, 0.000000, 0.000000, -0.000000, 1.709423, -0.000000	6.7745	2.9261	1:1
	1	1.023608, 0.000000, 0.000000, -0.000000, 0.795587, 0.000000	4.5080	3.0019	3:2
	2	1.012228, 0.000000, 0.000000, -0.000000, 0.986610, 0.000000	5.0709	2.9767	4:3
	3	1.039919, -0.000000, -0.000000, -0.000000, 0.626457, 0.000000	4.0537	3.0336	5:3
	4	1.007598, 0.000000, -0.000000, -0.000000, 1.103704, -0.000000	5.4095	2.9650	5:4
	5	1.005068, -0.000000, -0.000000, 0.000000, 1.185646, -0.000000	5.6354	2.9580	6:5
N. Butterfly	0	0.997458, 0.000000, 0.180887, 0.000000, -0.267825, 0.000000	6.8018	2.9836	1:1
	1	0.909486, 0.000000, 0.147987, 0.000000, -0.029831, 0.000000	3.4014	3.0880	2:1
	2	1.029694, 0.000000, 0.319069, 0.000000, -0.257194, 0.000000	10.2035	2.8829	2:3
	3	0.968194, 0.000000, 0.462906, 0.000000, -0.161387, -0.000000	11.3371	2.7862	3:4
	4	0.906875, 0.000000, 0.143163, -0.000000, -0.080985, 0.000000	4.0817	3.0878	5:3
	5	0.947594, 0.000000, 0.157279, -0.000000, -0.202470, 0.000000	5.4420	3.0381	5:4
	6	0.956277, 0.000000, 0.160945, 0.000000, -0.217203, 0.000000	5.6687	3.0280	6:5
Dist. Retro.	0	0.851461, 0.000000, 0.000000, 0.000000, 0.478229, 0.000000	2.2669	2.9622	3:1
	1	0.676037, 0.000000, 0.000000, -0.000000, 0.743602, 0.000000	5.1012	2.8529	4:3
	2	0.895774, 0.000000, 0.000000, 0.000000, 0.474414, 0.000000	1.3597	3.0173	5:1
	3	0.829698, 0.000000, 0.000000, -0.000000, 0.494298, 0.000000	2.7198	2.9446	5:2
	4	0.757239, 0.000000, 0.000000, -0.000000, 0.588358, 0.000000	4.0817	2.9005	5:3
	5	0.632343, 0.000000, 0.000000, 0.000000, 0.842931, 0.000000	5.4414	2.8232	5:4
	6	0.907375, 0.000000, 0.000000, -0.000000, 0.483388, 0.000000	1.1347	3.0402	6:1
	7	0.591235, 0.000000, 0.000000, 0.000000, 0.945278, 0.000000	5.6690	2.7916	6:5
	8	0.591418, 0.000000, 0.000000, 0.000000, 0.944802, 0.000000	5.6681	2.7918	6:5

Table A.1: CR3BP periodic orbit initial conditions (continued).

Orbit Family	**Orbit Index**	**Initial Conditions (DU)**	**Period (TU)**	**Jacobi Constant**	**Resonance (≈)**
E. Low Pro.	0	1.126435, 0.000000, 0.000000, -0.000000, 0.129877, 0.000000	3.4018	3.1626	2:1
	1	1.126444, 0.000000, 0.000000, -0.000000, 0.129825, 0.000000	3.4015	3.1626	2:1
	2	1.126454, 0.000000, 0.000000, -0.000000, 0.129772, 0.000000	3.4012	3.1626	2:1
	3	1.126463, 0.000000, 0.000000, -0.000000, 0.129719, 0.000000	3.4008	3.1626	2:1
	4	1.126473, 0.000000, 0.000000, -0.000000, 0.129666, 0.000000	3.4005	3.1626	2:1
	5	1.126482, 0.000000, 0.000000, -0.000000, 0.129614, 0.000000	3.4001	3.1626	2:1
	6	1.122229, 0.000000, 0.000000, -0.000000, 0.104933, 0.000000	2.2678	3.1709	3:1
	7	1.100709, 0.000000, 0.000000, -0.000000, 0.164162, 0.000000	1.7003	3.1753	4:1
	8	1.061527, 0.000000, 0.000000, -0.000000, 0.340148, 0.000000	1.3611	3.1811	5:1
	9	1.061584, 0.000000, 0.000000, -0.000000, 0.339720, 0.000000	1.3601	3.1812	5:1
	10	1.070939, 0.000000, 0.000000, -0.000000, 0.284993, 0.000000	1.3611	3.1823	5:1
	11	1.070848, 0.000000, 0.000000, -0.000000, 0.285457, 0.000000	1.3606	3.1823	5:1
	12	1.070757, 0.000000, 0.000000, -0.000000, 0.285922, 0.000000	1.3601	3.1823	5:1
	13	1.070667, 0.000000, 0.000000, -0.000000, 0.286387, 0.000000	1.3596	3.1823	5:1
	14	1.129685, 0.000000, 0.000000, -0.000000, 0.092717, 0.000000	2.7215	3.1692	5:2
L4 Planar	0	0.904300, 0.866025, 0.000000, 0.175400, -0.546700, 0.000000	6.4836	2.8329	1:1
W. Low Pro.	0	0.999349, 0.000000, 0.000000, -0.000000, 1.371593, 0.000000	2.2681	3.1839	3:1
	1	0.999363, 0.000000, 0.000000, -0.000000, 1.370637, 0.000000	2.2671	3.1839	3:1
	2	1.008986, 0.000000, 0.000000, -0.000000, 0.957299, 0.000000	1.7010	3.1861	4:1
	3	1.009017, 0.000000, 0.000000, -0.000000, 0.956431, 0.000000	1.6997	3.1861	4:1
	4	1.021275, 0.000000, 0.000000, 0.000000, 0.701783, 0.000000	1.3612	3.1893	5:1
	5	1.021325, 0.000000, 0.000000, 0.000000, 0.701006, 0.000000	1.3604	3.1893	5:1
	6	1.021375, 0.000000, 0.000000, 0.000000, 0.700229, 0.000000	1.3596	3.1894	5:1
	7	1.045433, 0.000000, 0.000000, 0.000000, 0.432190, 0.000000	1.1338	3.1963	6:1
	8	1.045471, 0.000000, 0.000000, -0.000000, 0.431786, 0.000000	1.1328	3.1964	6:1
L4 Long	0	0.468586, 0.887598, 0.000000, 0.013572, -0.005511, 0.000000	21.0757	2.9881	1:3
	1	-0.463998, 1.045147, 0.000000, 0.154044, 0.237071, -0.000000	26.1775	2.9764	1:4
3:2 Resonant	0	0.665646, 0.000000, 0.000000, 0.000000, 0.621468, 0.000000	11.3364	3.0472	3:4
L1 Halo	0	0.871245, -0.000000, 0.189871, -0.000000, 0.238390, 0.000000	2.2396	2.9978	3:1
	1	0.841174, -0.000000, 0.160524, -0.000000, 0.262475, 0.000000	2.6906	3.0258	5:2