Analysis of Persistent Detection Corridors for Cislunar Space Situational Awareness

Michael Klonowski
University of Colorado Boulder
Casey Heidrich
University of Colorado at Boulder

Naomi Owens-Fahrner
BAE Systems, Inc.

Marcus J. Holzinger
University of Colorado at Boulder

ABSTRACT

As humanity nears its long-awaited return to the Moon, this time for good, space situational awareness (SSA) architectures must be designed so that crewed missions through Cislunar space can be executed as safely as possible. Critically, crewed missions through Cislunar space to the Moon or Lagrange points may require near persistent detection or lines of communication by operators. However, distributed space-based SSA architectures consisting of observers with electro-optical sensors result in complex distributions of detectable regions throughout Cislunar space over time. Persistent detection will require that there exists feasible trajectories that traverse these regions. In this work, we introduce a novel method that captures the dynamics of detectable regions of Cislunar space, and uncovers corridors of persistent detection to be used in mission planning where high levels of coverage are paramount. Mission designers can use these corridors as references knowing that trajectories residing within said corridors maintain coverage minimums within the chosen architecture.

1. INTRODUCTION

As humanity advances toward a permanent presence on the Moon, epitomized by NASA's Artemis program [20] and a myriad of commercial and governmental missions, the extension of Space Situational Awareness (SSA) capabilities into Cislunar space has become crucial [14]. Cislunar space, defined as the vast region between Earth and the Moon, is a complex and dynamic environment where the safety of crewed missions hinges partially on the ability to maintain near-continuous detection and communication of spacecraft. The inherent challenges in this domain include the variability of observable regions, the influence of solar illumination on detection capabilities, and the need to safeguard high-value assets, particularly in scenarios where unexpected events could jeopardize mission success.

Current Earth-based SSA architectures are primarily designed for operations in Earth orbit and are ill-equipped for the unique demands of SSA in Cislunar space. As such, space-based architectures are necessary for conducting effective SSA activities in the new regime. However, the dynamic nature of Cislunar space, characterized by the movement of observers and the changing illumination conditions, results in highly variable coverage distributions within the field of regard of observers. Consequently, mission designers are confronted with the challenge of ensuring that critical assets remain detectable throughout their journey, especially during transits between Earth and the Moon or during operations at the Lagrange points. Decision makers are inevitably beholden to planning missions that take into account the capabilities and coverage behavior of whatever Cislunar SSA architecture is established.

In response to these challenges, we introduce the concept of detection corridors and a novel tool to aid in uncovering corridors that provide minimum levels of persistent coverage, termed *persistent detection corridors* (PDCs). 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.

This paper presents 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. 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.

2. BACKGROUND

2.1 Cislunar SSA

With NASA's Artemis missions planning for a permanent human presence on the Moon [20], and numerous other government and commercial focused missions to the Moon and within Cislunar space planned within the next couple of decades, there is an increasing effort being made to extend Space Situational Awareness (SSA) capabilities to the Cislunar regime [1]. As shown throughout the literature [17, 10, 14, 22, 2, 25, 24], space based distributed SSA architectures are necessary for providing robust coverage to the entirety of Cislunar space.

In previous work [17, 16] we have explored the massive design space of Cislunar architectures, and saw how attention must be placed on the relative positions and capabilities of observers within any given architecture so objectives of interest are maximized in the Pareto sense. Considering observers with electro-optical sensors, as a result of a given layout in Cislunar space, the distribution of detectable regions can vary significantly over time due to observer positions and the varying solar illumination. This variation introduces complexities in architecture performance and thus SSA activities over time that must be accounted for. An architecture's performance can not be solely evaluated in terms of total volume coverage or coverage of a single trans-lunar injection. Rather, an architecture's utility must be considered in the context of how *useful* its resultant coverage distribution is to mission designers, considering variations in individual mission objectives.

Last year [16], we explored how cost was distributed between SSA architectures and cooperative agents (CAs) executing a mission within Cislunar space. This work showed how architectures providing high levels of coverage to CAs during their transfers were generally higher in cost to enable lower total control effort by the CA. On the other hand, lower cost architectures providing similar levels of coverage passed on that cost to the CA, requiring higher levels of control effort. As an initial proof of concept, this work was the first that considered how stakeholders with often competing mission objectives may end up interacting with Cislunar SSA architectures.

2.2 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 our previous work, 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. 1. 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

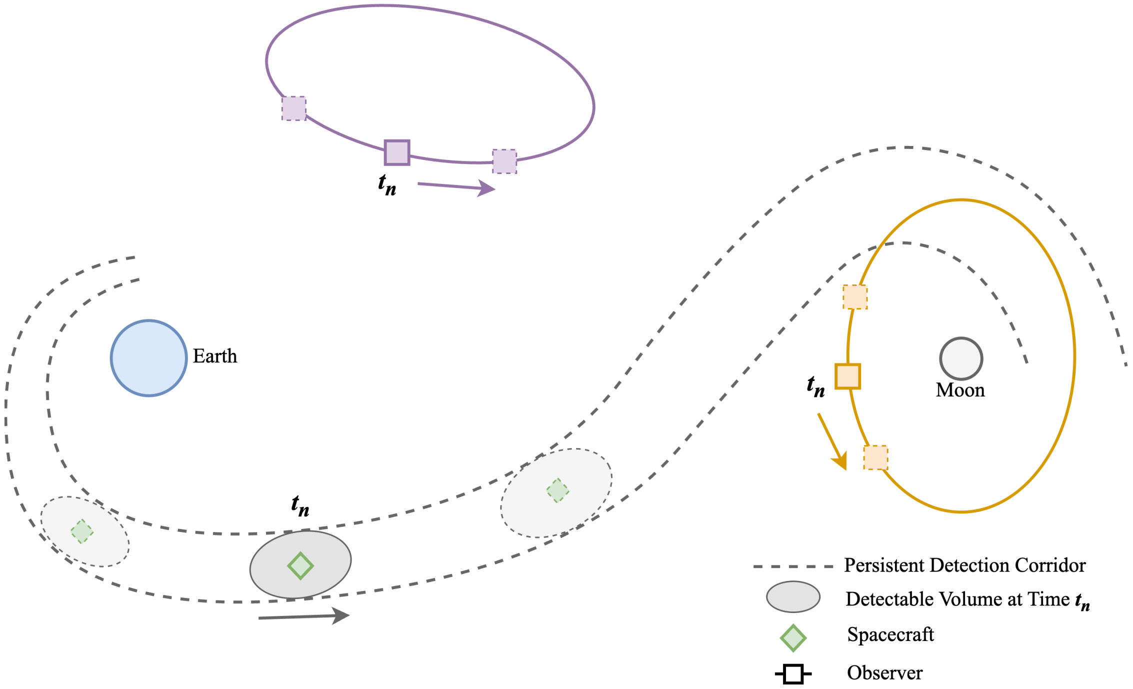

t_n
Earth
t_n
Moon
t_n
Persistent Detection Corridor
Detectable Volume at Time t_n
Spacecraft
Observer

Fig. 1: Visualization of a persistent detection corridor for an arbitrary Cislunar SSA architecture.

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.

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. 2 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:

$$\mathbf{C}(\hat{\mathbf{x}}) = \{\Gamma_{\hat{\mathbf{x}}}(t_i) : \quad t_i \in \{t_0, ..., t_f\}\} \tag{1}$$

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

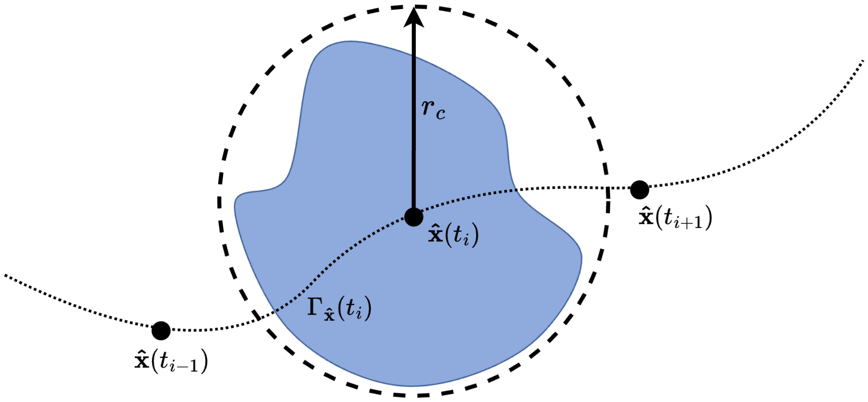

r_c
$\hat{\mathbf{x}}(t_i)$
$\hat{\mathbf{x}}(t_{i+1})$
$\hat{\mathbf{x}}(t_{i-1})$
$\Gamma_{\hat{\mathbf{x}}}(t_i)$

Fig. 2: Diagram of a local neighborhood of detectability about reference a trajectory at time t_i. The shaded region represents the approximate region of continuous detectability about the reference trajectory's position.

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.

3. METHODS

3.1 Cislunar SSA Architectures

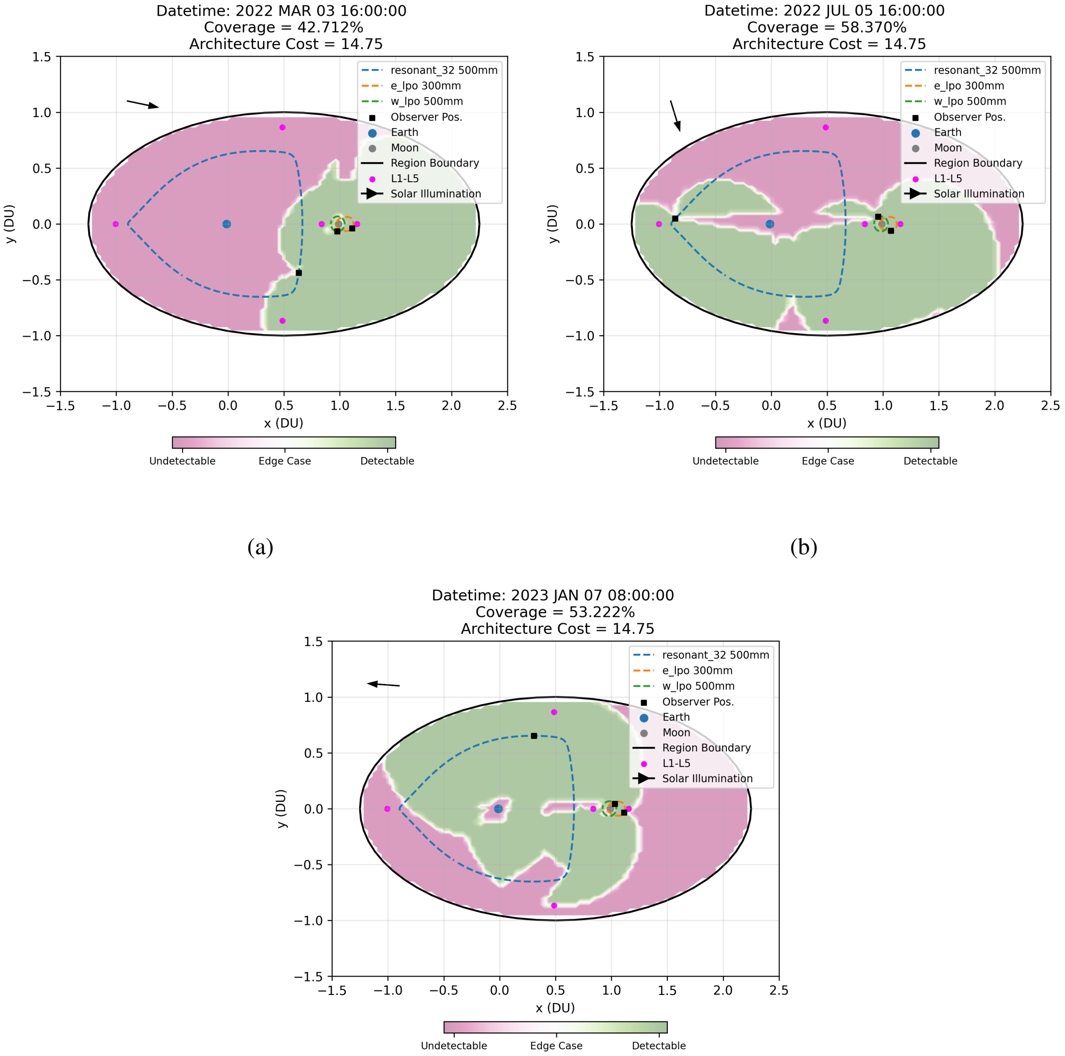
We can model a Cislunar SSA architecture as a collection of up to five observers with electro-optical (EO) sensors and varying telescope diameters. Through the modeling of EO sensors, we are able to obtain simulated measurements of photometric SNR of space objects within the field of regard of each observer. This enables use to use boolean labels indicating whether a target is detectable or not. Coverage metrics associated with the performance of an architecture can be obtained using methods from our previous work [17]. As in this work, we define an architecture as a set of observers $\{Obs_1, Obs_2, ...\}$, with each observer represented as a tuple:

$$Obs_i = [Family,\ Index,\ Phasing,\ Telescope\ Diameter] \tag{2}$$

where *Family* indicates the type of orbit (i.e. *L1 Lyapunov, Distant Prograde, GEO, etc.*), *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.

To obtain a distribution of regions of detectability, we create a grid of evenly distributed points in the $x-y$ plane in the CR3BP rotating frame, each point modeled as a 1 meter Lambertian sphere as in [22]. The process of calculating detectability takes place within a simulation defined at some initial epoch for some total time. At this initial epoch, the relative angular phasing of each observer is enforced and the observers' state histories are pre-propagated to be evaluated at later times during the simulation. Within a user specified time span, the detectability of each grid point is evaluated at each time step. The coverage per time step is then calculated as the number of detectable grid points divided by the total number of grid points sampled. Over the time span, the total average coverage is then the average of the coverage per time step. An example architecture and its resultant detection distribution from [17] is shown in

Fig. 3 for three epochs. The process that is described below aims to capture the dynamics of these varying detectable regions so that they can be leveraged by mission designers seeking to prioritize detectability of their critical assets.



(a)

(b)

(c)

Fig. 3: Variation in detectable regions for a selected architecture consisting of 3:2 Resonant 500mm, East Low Prograde 300mm, and West Low Prograde 500mm observers. Shown at three epochs over the simulation history. Artifacts and regions labeled as edge cases are a result of the discretization and interpolation of the sampled grid points.

3.2 Sample-Based Trajectory Planning

Sample-based trajectory 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 [18]. 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 [21, 4, 8]. 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 [5, 6] 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.

In this work, we implement the graph-based method from Bruchko & Bosanac with modifications to state sampling, edge weight calculation, and trajectory recovery that are informed by Cislunar SSA domain knowledge.

3.3 Graph Construction

As noted in [5, 6], construction of a graph approximating the dynamical environment of the CR3BP requires careful handling of parameters dictating the order, size, and resolution of the graph. Here, we outline the steps of graph construction as well as the parameters used to guide graph construction and the effects they have on the resultant paths and corridors recovered. The steps of this process are also outlined in the diagram in Fig. 4.

3.3.1 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 [6, 5], 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 are as uniformly distributed across the distribution of detectable points as possible. At each time step, we select up to n_c detectable points as centroids, ensuring that spatial uniformity is achieved. Additionally, we select a small number n_e of detectable points minimizing the distance to a desired start and goal region, here chosen to be the Earth and the Moon, respectively. These points are added as additional centroids to ensure that the graph captures the dynamics of detectable regions relevant to Earth-Moon transfers.

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 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, edges are added to the graph connecting backwards nodes to their centroid nodes, and centroid nodes to their forward nodes. These edges are directed, representing motion through centroids forward in time. Included with these edges is a default weight of 0, since traversal through the centroid represents natural motion in the CR3BP. The collection of nodes defined 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 with up to $2n_v$ edges. Fig. 5 shows an example of a centroid position and its corresponding forward and backward propagated nodes.

The identification of centroids and construction of local neighborhoods occurs at each time step during the time span of interest, resulting in up to $N \cdot 3n_v \cdot (n_c + n_e)$ nodes and $N \cdot 2n_v \cdot (n_c + n_e)$ edges in the graph for N time steps. The next step involves taking these disjoint local neighborhoods and creating connections between them via edges, with weights representing the dynamical feasible of traversing forward in time between neighborhoods.

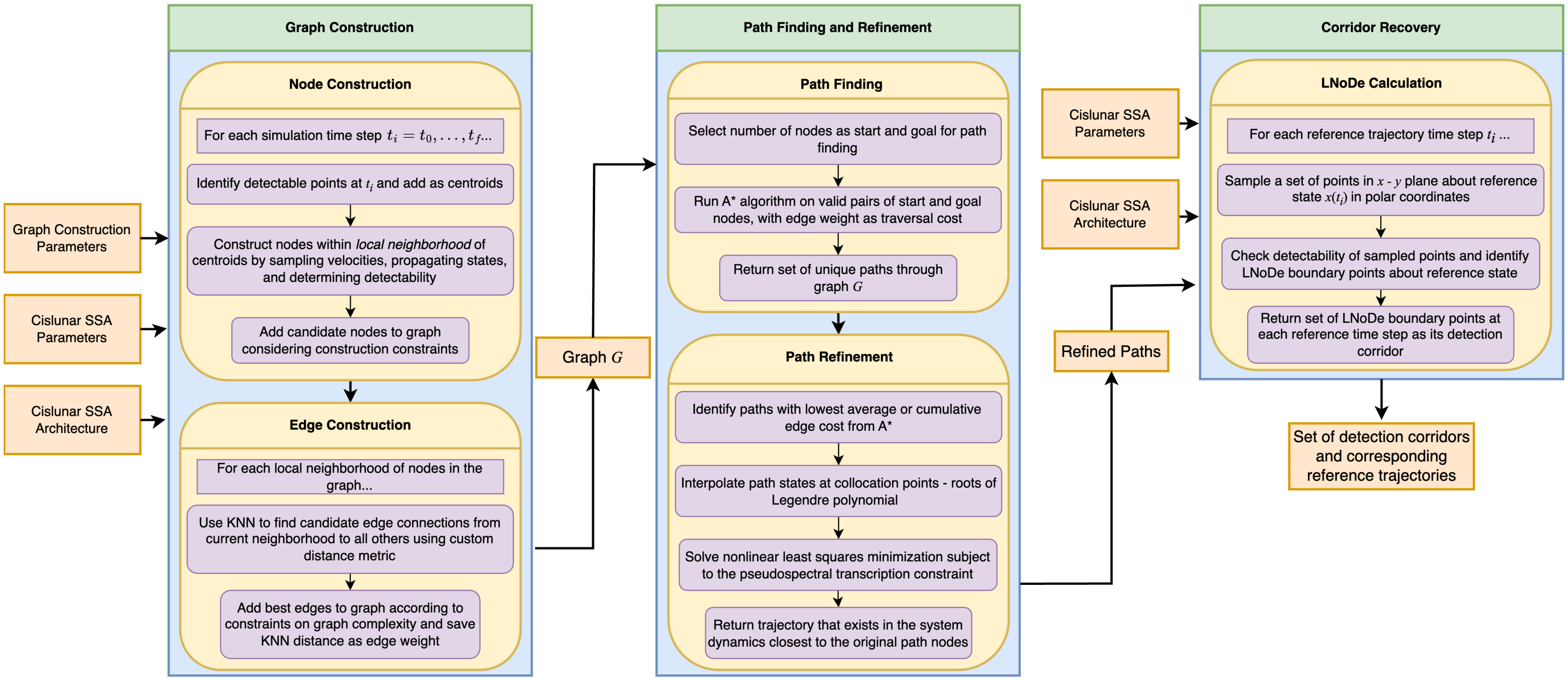

Graph Construction Parameters
Cislunar SSA Parameters
Cislunar SSA Architecture
Graph Construction
Node Construction
For each simulation time step $t_i = t_0, \dots, t_f$...
Identify detectable points at t_i and add as centroids
Construct nodes within *local neighborhood* of centroids by sampling velocities, propagating states, and determining detectability
Add candidate nodes to graph considering construction constraints
Edge Construction
For each local neighborhood of nodes in the graph...
Use KNN to find candidate edge connections from current neighborhood to all others using custom distance metric
Add best edges to graph according to constraints on graph complexity and save KNN distance as edge weight
Graph G
Path Finding and Refinement
Path Finding
Select number of nodes as start and goal for path finding
Run A* algorithm on valid pairs of start and goal nodes, with edge weight as traversal cost
Return set of unique paths through graph G
Path Refinement
Identify paths with lowest average or cumulative edge cost from A*
Interpolate path states at collocation points - roots of Legendre polynomial
Solve nonlinear least squares minimization subject to the pseudospectral transcription constraint
Return trajectory that exists in the system dynamics closest to the original path nodes
Refined Paths
Cislunar SSA Parameters
Cislunar SSA Architecture
Corridor Recovery
LNoDe Calculation
For each reference trajectory time step t_i ...
Sample a set of points in x - y plane about reference state $x(t_i)$ in polar coordinates
Check detectability of sampled points and identify LNoDe boundary points about reference state
Return set of LNoDe boundary points at each reference time step as its detection corridor
Set of detection corridors and corresponding reference trajectories

Fig. 4: Outline of detection corridor generation process.

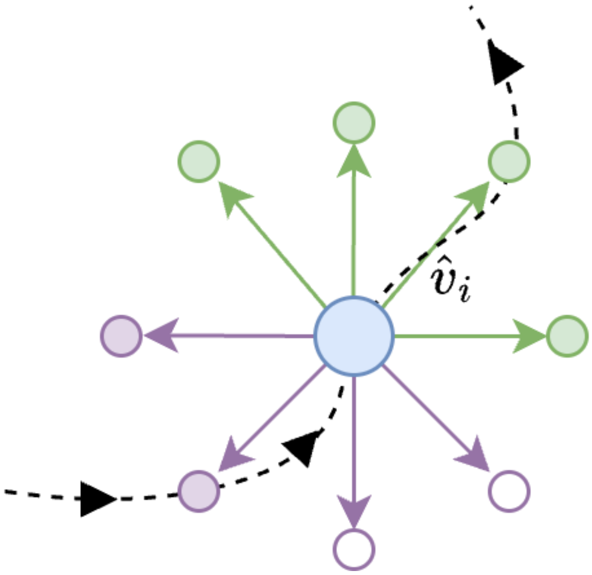


Fig. 5: A single centroid (blue) with a selection of velocity directions propagated forward (green) and backward (purple). Traversal through the neighborhood starting from a backward node to its centroid and finally through the corresponding forward node represents natural motion in the system dynamics. Shaded circles indicate detectability by an architecture, the states of which are added as nodes to the graph. Figure adopted from [6].

3.3.2 Global Edge Construction

Our graph now consists of nodes and edges representing natural motion in the local neighborhood of detectable grid points in the simulation. To create connections across neighborhoods, we use k-nearest neighbors (KNN) [19, 7] with a custom distance function to find the k best candidate nodes. A primary concern when adding edges between nodes is that the connection represents a dynamically feasible trajectory. 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 the edge. Here, we implement a novel distance metric inspired by methods in direct collocation [15] 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

$$\begin{aligned} \delta_{k,k+1} = (\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) - \frac{1}{2} h_k \left[\boldsymbol{f}(\boldsymbol{x}_{k+1}) + \boldsymbol{f}(\boldsymbol{x}_k) \right] \\ h_k = (t_{k+1} - t_k) \end{aligned} \tag{3}$$

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 custom distance function used in KNN for edge creation is then defined as

$$D_{i,j} = \|\delta_{i,j}\|_2 + \hat{d}_{i,j} + \hat{\alpha}_{i,j} \tag{4}$$

where $\hat{d}_{i,j}$ is the Euclidean distance between nodes i and j normalized by a maximum distance and $\hat{\alpha}_{i,j}$ is the angle between the velocity vectors between nodes i and j normalized by a maximum angle. Used within `sklearn`'s `NeareastNeighbors` module [19], this metric encourages edge connections between nodes that are near in position space, maintain continuity in direction of motion, and minimize error with respect to the environment's dynamics.

Iterating over all non-centroid neighborhood nodes in the graph, candidate nodes for edge creation are determined by finding the k-nearest neighbors of each node. The resultants candidates from KNN are then filtered so that 1) edges are not created between nodes that share a position, 2) edges are not created that connect backward propagated nodes to forward propagated nodes across neighborhoods, and 3) edges are not created that connect nodes defined at the same simulation time. The second filter condition ensures that any path through the graph will fully traverse through local neighborhoods where edge connections forward in time represent natural motion in the system. Once candidate edges between nodes are identified and properly filtered, they can be added to the graph with an edge weight determined by the distance metric in Eqn. (4). The process of constructing edges across neighborhoods is the msot computationally intense process throughout graph construction. Luckily, it is easily parallelizable with `Python`'s `multiprocessing` module.

Edge construction between nodes across local neighborhoods is subject to several constraints. First, there is a limit on the maximum number of edges that can originate from all nodes within a neighborhood. Second, we place a limit on the number of edges connecting a given neighborhood to any single other neighborhood, which promotes diverse connections across multiple neighborhoods, rather than concentrating all the best connections on just one. Lastly, each node has a limit on the total number of incoming and outgoing edges, apart from those associated with neighborhood connections. This ensures a more even distribution of edges among nodes, facilitating a broader and more representative approximation of the environment. The constraints chosen here significantly impact the graph's overall complexity and, consequently, the precision with which it captures the underlying dynamics.

3.4 Path Finding

A path in the context of graph theory is a subgraph P of graph G consisting of distinct nodes and edges that traverse through G given initial and final nodes η_0 and η_f [3]. A plethora of algorithms have been developed for uncovering paths in graphs, with A* [13], Dijkstra's [9], and Yen's [26] among those most widely used due to their guarantees of convergence given certain criteria are met. In this work, we use an implementation of A* from the `networkx` `Python` package [12].

3.4.1 A*

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 [13].

A* operates by exploring paths that minimize the function $f(\eta) = g(\eta) + h(\eta)$, where $g(\eta)$ is the cumulative edge cost from the start node to η, and $h(\eta)$ 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 [13].

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 found that a constant heuristic of 0 returned paths that most closely resembled natural motion within the dynamical system. In future work we will explore heuristics that better reflect the true cost to reach the goal node at any given node. The edge costs between nodes, represented by $g(\eta)$, are then determined by our custom distance function in Eqn. (4).

3.5 Path Refinement

Paths recovered from A* within the graph represent approximate feasible trajectories within detectable regions of Cislunar space. However, due to the discretization of the environment, these paths 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.

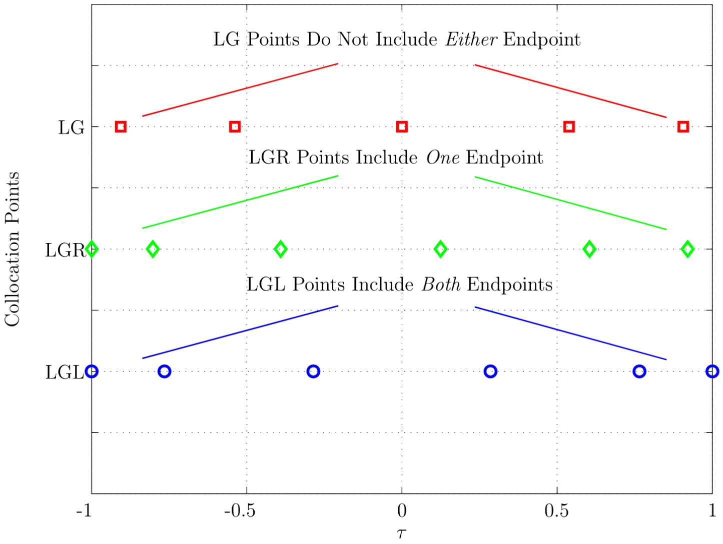


Fig. 6: Distribution of different types of Legendre-Gauss collocation points. Obtained from [11].

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* [15]. Fig. 6 displays a selection of Legendre polynomial roots, obtained from [11]. 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

$$\begin{aligned} \min_X & (X - X_C)^T (X - X_C) \\ s.t. \quad & DX - \boldsymbol{f}(X) = 0 \end{aligned} \tag{5}$$

where $\boldsymbol{f}(X)$ are the CR3BP dynamics evaluated at the collocation points and $DX = \dot{X}$ are the derivatives of the interpolated collocation points. The constraint in Eqn. (5) is referred to as the *pseudospectral transcription error* and similar to the trapezoidal transcription error in Eqn. (3), 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 [15]. The solving of Eqn. (5) is performed with `scipy`'s `minimize` function using the SLSQP algorithm [23]. 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.

3.6 Persistent 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

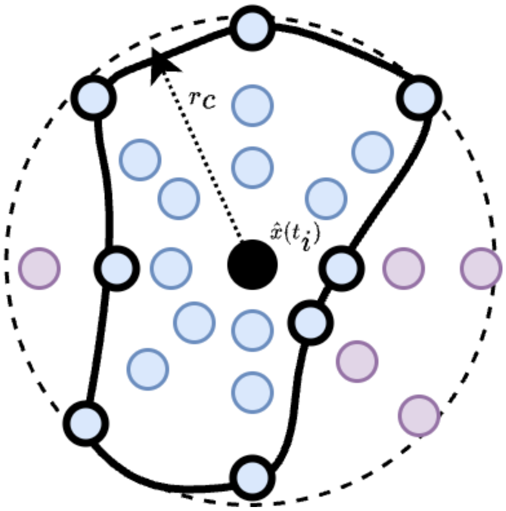


Fig. 7: Polar coordinate sampling about the reference trajectory from which LNoDe boundary points are estimated. Detectable points are colored blue with purple points undetectable. The points determined as the boundary are outlined in black with the interpolated boundary similarly marked connecting these points.

detectable by the architecture. Fig. 7 visualizes this method for determining the boundary points of an LNoDe at time t_i. As discussed previously, the set of LNoDes about reference trajectory constitutes its detection corridor, representing the detectability of the trajectory and its local neighborhood in position space over time.

3.6.1 Persistence Metrics

Three 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:

$$\rho_{\Gamma_{\hat{\mathbf{x}}}(t_i)} = \frac{A_{\Gamma_{\hat{\mathbf{x}}}(t_i)}}{A_{r_c}} \tag{6}$$

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 $\boldsymbol{C}(\hat{\mathbf{x}})$ 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

$$\rho_{\boldsymbol{C}(\hat{\mathbf{x}})} = \frac{1}{N}\sum_{t_i}^{t_f}\rho_{\Gamma_{\hat{\mathbf{x}}}(t_i)} \tag{7}$$

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.

Finally, persistence can be further described by the frequency and duration of gaps in coverage. We can look at the standard deviation of LNoDe persistence over the trajectory duration, giving us insight into the distribution of

Observer Orbit	Initial Phasing	Telescope Diameter	Period (TU)	Jacobi Energy
3:2 Resonant	0.333	500mm	11.284	3.047
East Low Prograde	0.555	300mm	2.226	3.171
West Low Prograde	0.444	500mm	1.075	3.196

Table 1: Member observers of a selected optimal architecture. The architecture was found to be Pareto-optimal in [17], with a cost of 14.75 and total average volume coverage of 73.81% over 30 simulation epochs.

detectable regions throughout a corridor:

$$\sigma_{C(\hat{x})} = \sqrt{\frac{1}{N}\sum_{t_0}^{t_f}(\rho_{\Gamma_{\hat{x}}(t_i)} - \rho_{C(\hat{x})})^2} \tag{8}$$

Corridors with high variation per the standard deviation indicate non-uniform persistence per LNoDe, an undesirable result if there is high likelihood of deviation from the reference trajectory at time steps where LNoDe persistence is low. On the other hand, low variation in persistence indicates a more uniform distribution of persistence, a desirable result if the total corridor coverage meets desired minimums. 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.

4. RESULTS

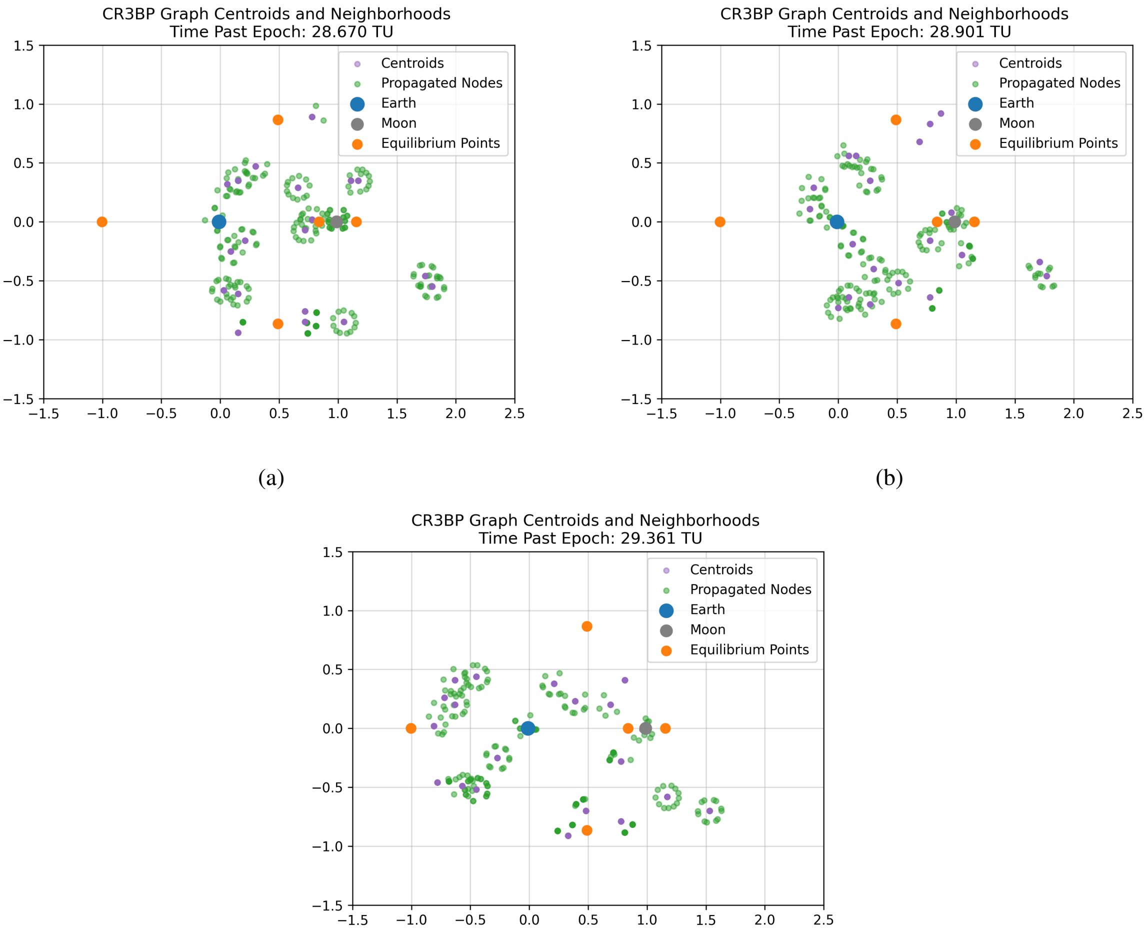
4.1 An Optimal Architecture

As found in previous work [17, 16], 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 initial work, we seek to uncover persistent detection corridors defined by natural reference trajectories. As such, we explore architectures resulting in high average coverage of volumes and low CA control efforts. These constraints result in high cost architectures with multiple observers, enabling large continuous regions of detectability throughout Cislunar, as shown in a selected architecture in Fig. 3.

For an initial proof of concept for this method, we use the architecture layout from Fig. 3. The observers and characteristics of the architecture observers are shown in Table 1.

4.2 Initial Graph

The initial simulation epoch is set to 01 Jan 2022 00:00:00 GMT, at which point initial observer phasings are set. Starting arbitrarily past the initial epoch at 04 May 2022 16:00:00 GMT, we begin graph construction by evaluating the detectability of evenly distributed grid points in the $x-y$ plane, over a time period of 30 days with a time step of 12 hours. As outlined above, at each time step, detectable points are sampled 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. Detectability of propagated states by the architecture is evaluated and detectable states are added as nodes, with edges representing natural motion through the corresponding centroid state. In Fig. 8, centroids and propagated nodes added to the graph are shown for three time steps within the simulation. After node construction, edges between nodes across neighborhoods are constructed so that the trapezoidal transcription error is minimized, and the KNN distance metric from Eqn. (4) evaluated between nodes is added as a weight to the relevant edge. Fig. 9 displays a 3D representation of this resultant graph, with time past simulation epoch represented in the z-axis.



(a)

(b)

(c)

Fig. 8: Centroids and neighborhood nodes added to the graph at distinct time steps.

4.3 Paths

With our graph constructed, the A* path finding algorithm is applied to uncover near feasible trajectories. We choose to use a heuristic that always returns 0, as this was found to produce better resultant paths with respect to dynamical feasibility. Edge weights added during graph construction are used by A* to estimate the cost of traversing along edges. To find paths representative of crewed missions from the Earth to the Moon, we identify a number of initial and goal nodes that minimize distance to the Earth and Moon respectively at unique time steps. We iterate over each initial node to find paths to every goal node. This results in a variety of paths representing transfers of varying durations, as well as paths with varying departure and arrival dates. The selection of initial and goal nodes is easily altered to account for specific mission requirements if so desired.

Figure 10 displays a number of paths found after iterating through 50 initial nodes and 50 goal 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 a goal node, the path had to traverse across a larger distance or period of time, during which detection is ambiguous. Another important note is that even if a path is found from an initial to goal node, it may not be close to feasible in the dynamical system, as it may simply be the path corresponding to the lowest cumulative edge cost. From these paths, it is helpful to filter out or sort paths by cumulative edge cost or transcription error.

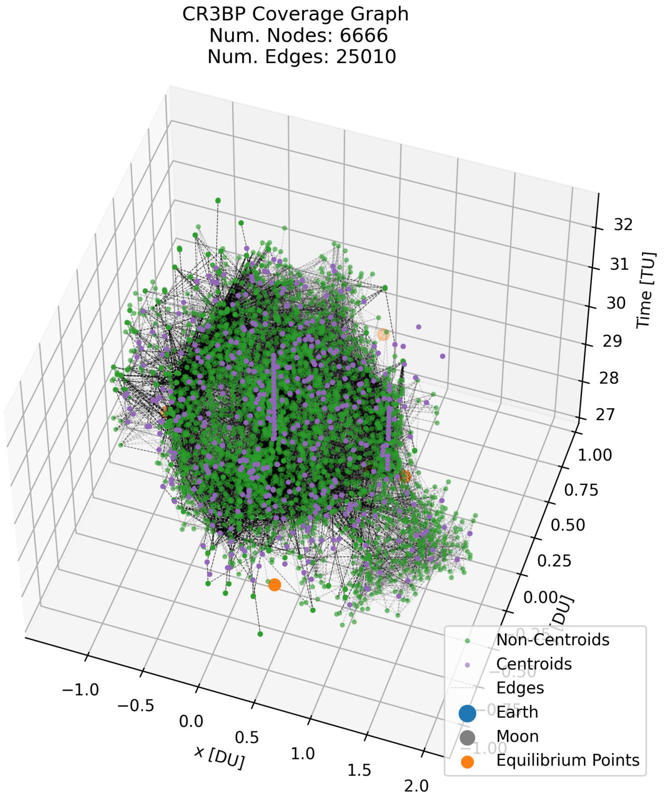


Fig. 9: Three-dimensional graph representing dynamical evolution of detectable regions of Cislunar space over time as a result of a selected architecture.

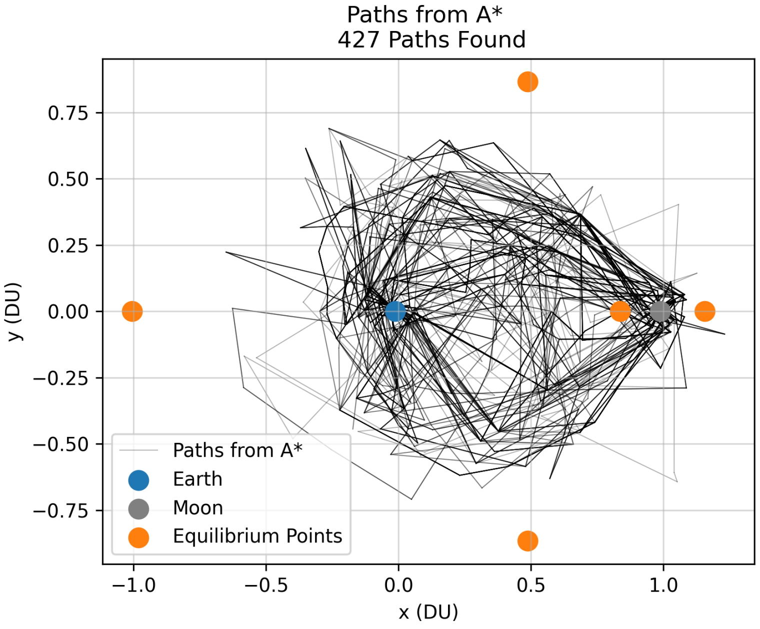
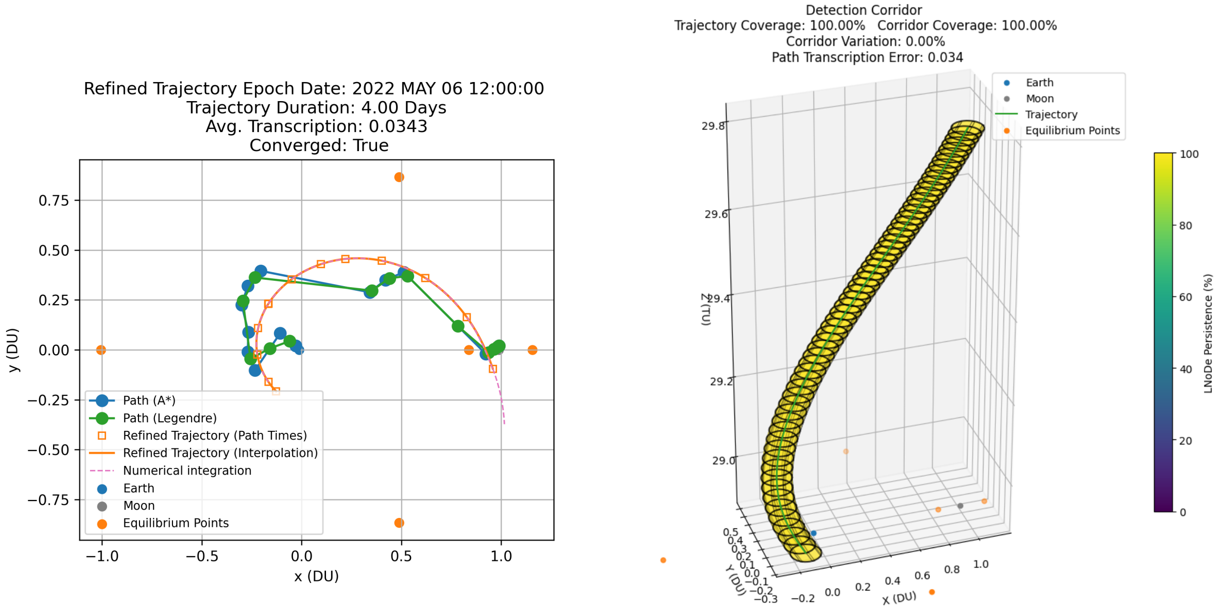


Fig. 10: Set of paths recovered using A* within the detectability graph. Bends along paths indicate the position of a graph node, with edges representing traversal between nodes. Edges are not necessarily representative of true motion within the CR3BP.

4.4 Corridors

We select a couple of paths from our graph with which to construct detection corridors about. The paths are sorted by cumulative edge weight and the methods outlined above are used to obtain nearby feasible trajectories within the CR3BP. We begin by selecting three unique paths on which to perform the interpolation at Legendre collocation points, nonlinear least squares optimization, and corridor generation. Figures 11 - 13 display the results of trajectory refinement and corridor generation. In each plot is shown the resultant interpolant of the original path, the solution to the least squares optimization, and finally the corresponding detection corridor displayed in a 3D plot, with LNoDes connected across time via interpolation of their boundary points, for clarity. Note that for ease of visualization, we implement a corridor radius of $r_c = 0.1DU$ for each of the results that follow.



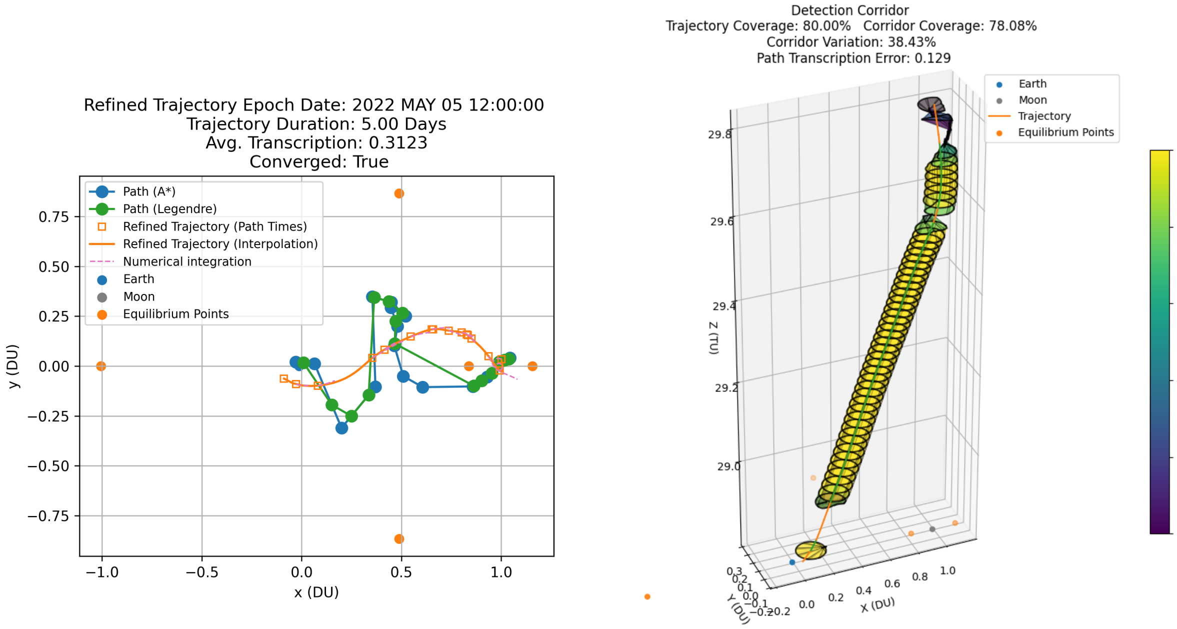
(a) Graph path refinement.

(b) Resultant corridor about reference trajectory.

Fig. 11: Refined trajectory and resultant detection corridor with high levels of persistence.

In Fig. 11, we recover a trajectory starting within the vicinity of the Earth (albeit quite outside of GEO), with motion opposite of a geostationary orbit reaching the vicinity of the Moon within four days. The benefit of this trajectory is clear when looking at the resultant corridor. Both the reference trajectory and the corridor itself achieve 100% coverage by the architecture, with the trivial result of no corridor variation. However, this is a case where even though a feasible trajectory was found through the optimization in Eqn. (5) (the pseudospectral transcription error constraint was satisfied), the reference trajectory here is not realistic given that its initial motion around the Earth is opposite to that of most near-Earth orbits. Regardless, this is an example of a true PDC, where coverage is maintained of and about a reference trajectory for the duration of its transfer.

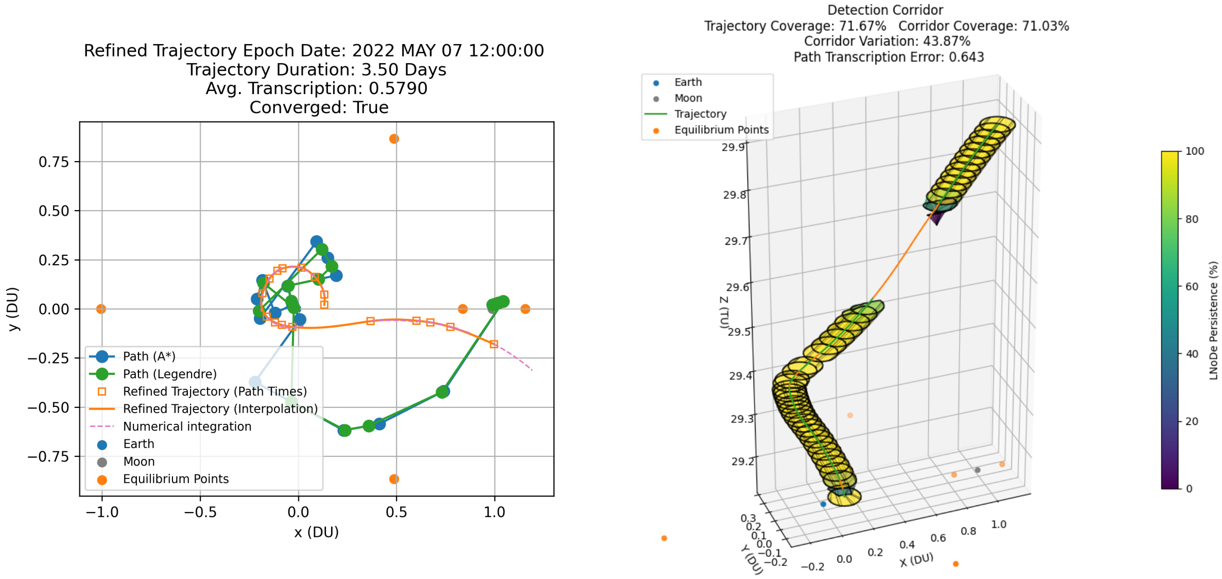
Fig. 12, when compared to the previous result, more closely resembles a trajectory that we may expect in a crewed mission to the Moon. This trajectory starts a day before the previous one, and has a duration of 5 days. Notably, the detection corridor about this reference trajectory results in more variation in persistence. There are clearly some large measurement gaps about the reference trajectory where no coverage is obtained, as well as LNoDes where significantly less coverage about the reference is provided. This second observation is noticeable through the changing shape and color of certain LNoDes along the reference. These gaps in coverage are captured in the corridor variation metric, indicating a standard deviation in coverage among all time steps of 38.43%, a rather significant but intuitive value given the gaps. However, even with a substantial variation in coverage, the total corridor coverage is maintained around 80%, indicating potential for a persistent detection corridor, depending on how individual mission requirements may apply to gaps in coverage.



(a) Graph path refinement.

(b) Resultant corridor about reference trajectory.

Fig. 12: Refined trajectory and resultant detection corridor with small gaps in coverage but moderately high levels of persistence.



(a) Graph path refinement.

(b) Resultant corridor about reference trajectory.

Fig. 13: Refined trajectory and resultant detection corridor with significant gap in coverage during the middle of the transfer.

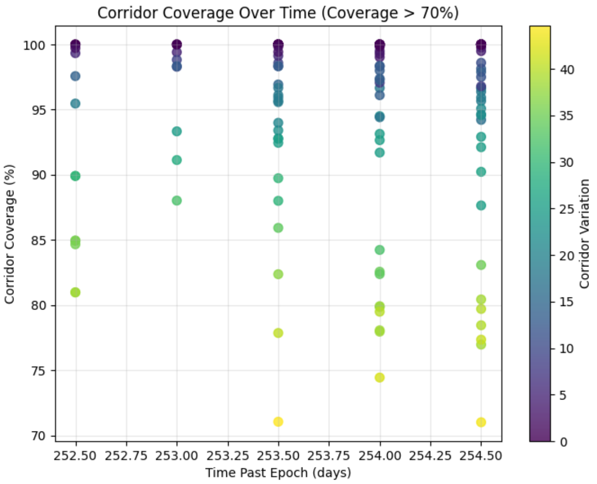


Fig. 14: Corridor coverage over time with marker color representing corridor variation.

Finally, Fig. 13 shows a reference trajectory starting two days after the previous with a total duration of 3.5 days. Similar to the previous trajectory, this transfer resembles what we may expect a "generic" Earth-Moon transfer to look like. It appears to start around Earth in an orbit near GEO, ending relatively close to the Moon with quite a bit of energy leftover. In this scenario, the nonlinear optimization converged to find a feasible path in the dynamics, but clearly the cost function from Eqn. (5) is rather large given how far away many of the path nodes are from the trajectory. As a result, within the middle part of the transfer, the detection corridor lacks any coverage, leading to a rather large gap. This is reflected in both the total corridor coverage and corridor variation metrics. Such a gap could potentially lead to significant loss of asset tracking or communication, and significantly increase risk to human life in the case of a crewed mission along this reference trajectory.

While we only show three selected corridors from this methodology, a great benefit to the automated generation of corridors allows for a wider analysis and selection of corridors that may be of relevance to mission designers. We can get a sense of the variations in corridor quality over trajectory epoch by looking at a scatter plot of resultant corridors. Fig. 14 shows a plot of corridor coverage plotted over days past the simulation epoch, with marker color representing corridor variation. As we might expect, corridor variation generally increases as corridor coverage decreases. This is not necessarily always guaranteed to be the case, as demonstrated in Figures 12 and 13 where coverage is similar but due to a large gap in detectability in Fig. 13 its corresponding variation is significantly higher. Meanwhile, the corridor in Fig. 12 ends up with lower variation in persistence due to a shorter detectability gap but ends up with a similar level of corridor coverage due to smaller variations in LNoDe persistence, as seen at the end of the transfer duration.

5. CONCLUSION

In this work we introduced a novel method for exploiting the capabilities of arbitrary Cislunar SSA architectures to ensure persistent coverage of crewed missions within Cislunar space. Through our tool, multiple feasible reference trajectories are uncovered which traverse primarily through detectable regions of space. Detection corridors along these reference trajectories can be evaluated per given mission requirements, and corridors that meet persistence minimums can be used for initial trajectory design. Enumeration of many such persistent detection corridors allows for a wide search of the trajectory design space where coverage of crewed and critical assets is crucial. At its core, this method is an initial exploration into a process for future mission design that prioritizes persistent coverage. Inevitably, a Cislunar SSA architecture will be constructed and missions influenced by the capabilities and resulting coverage of said architecture. By encapsulating the dynamics of detectable regions of Cislunar space within a graph, mission designers can make better decisions with respect to trajectory design while ensuring the safety of human life.

6. REFERENCES

[1] National Cislunar Science and Technology Strategy. *National Science and Technology Council*, Cislunar Technology Strategy Interagency Working Group, November 2022.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

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

[10] Carolin Frueh, Kathleen Howell, Kyle J DeMars, and Surabhi Bhadauria. Cislunar space situational awareness. In *31st AIAA/AAS Space Flight Mechanics Meeting*, pages 6–7, 2021.

[11] 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.

[12] 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.

[13] 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.

[14] M J Holzinger, C C Chow, and P Garretson. A Primer on Cislunar Space. 2021.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] Thomas Raymond Smith. *Using Motion Primitives to Rapidly Design Trajectories in Multi-Body Systems*. PhD

thesis, University of Colorado at Boulder, Boulder, CO, 2023.

[22] 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.

[23] 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.

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

[25] 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.

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