Development of Satellite Constellation Planning Algorithms

AERO 402 Final Individual Report

By Jonathan Jermstad

## Satellite Pointing Algorithm

### Background information

In satellite operations, two essential concepts related to a satellite's observation capabilities are the Field of View (FOV) and the Field of Regard (FOR). The FOV is the specific area that the satellite’s instruments can directly observe at any given moment, typically represented by a cone-shaped region extending from the satellite toward Earth. In contrast, the FOR encompasses the broader area that the satellite could potentially observe by adjusting its orientation. While the FOV is limited to the area currently being observed, the FOR depends on the satellite’s ability to reposition or slew to different targets.

Diagram, radar chart

Description automatically generated

Figure 1. FOV Diagram

To assess the angular displacement required for the satellite to point at a specific target within its FOR, we calculate the principal angle of rotation, derived from the unit vectors between the satellite and each target. This angle provides a direct measure of the satellite's required rotation, which is fundamental to defining the pointing requirements and slewing rate necessary to establish or maintain line of sight with each target.

However, a key challenge arises due to the satellite’s limited ability to slew between targets. Each reorientation requires torque applied through the satellite’s actuators, and these actuators have a maximum rate of movement they can achieve. This limit restricts how quickly the satellite can transition between targets, meaning that if a new target requires a larger or faster rotation than the actuators can supply, the satellite may not be able to reach the desired orientation in time. Consequently, any pointing algorithm must account for this slewing constraint to ensure that the satellite can reliably meet observation goals without exceeding its physical movement capabilities.

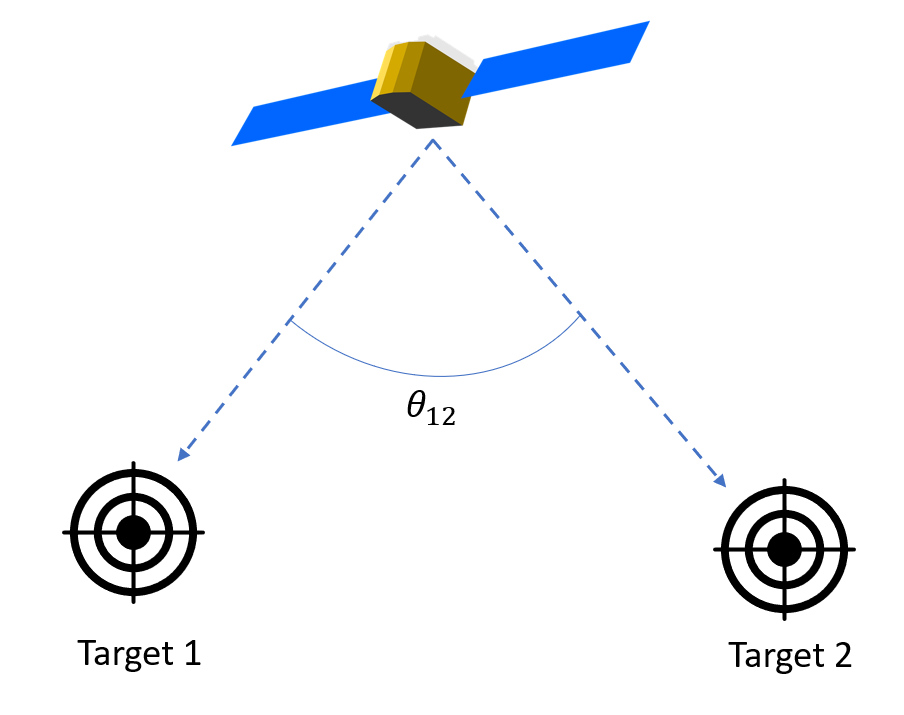


Figure 2. Principal Angle Example

Additionally, another key problem arose from the high computation cost associated with planning satellite target observations for the constellation optimizer. The complexity of scheduling observations for a large number of ground targets, especially when considering the various access intervals and angle bins (324 total), led to significantly prolonged computation times. As the number of targets increased, the planning process became more resource-intensive, requiring extensive time and computational power to generate feasible observation schedules. This inefficiency limited the scalability of the system, particularly in real-time scenarios where rapid adjustments to the satellite constellation's schedule were essential for optimizing coverage. The high computational cost also became a bottleneck when working with high-demand target scenarios, making it difficult to process multiple target requests efficiently within a reasonable time frame.

### Solution

#### Problem 1: Slewing Calculations

Addressing this challenge was one of my primary contributions this semester. I developed a pointing algorithm that enables precise selection of access times between each satellite and its ground targets. Each ground target is made up of 324 unique angle bins, and the goal is to ensure that the satellite constellation can efficiently point at each bin for every target and capture an image. To achieve this, the pointing algorithm prioritizes bin/target combinations based on the number of access points available for each combination. Those with fewer line-of-sight opportunities are addressed first. As the algorithm progresses, more frequent line-of-sight opportunities emerge, reducing the need for complex slewing maneuvers. This approach ensures efficient data collection from all bins while minimizing the time required for satellite repositioning.

As the algorithm iterates through various bin/target combinations, it processes data with multiple access times, satellite IDs, and unit vector values. From this data, the earliest available instance is selected and incorporated into each of the satellite ID specific plans. The availability of each access point is determined by the satellite’s slewing rate constraints. To calculate the slewing rate between two access points, we define a new term, θ­12, which represents the principal angle of rotation that we introduced earlier. This angle calculation can be seen in **Figure 3**, **Equation 1**.

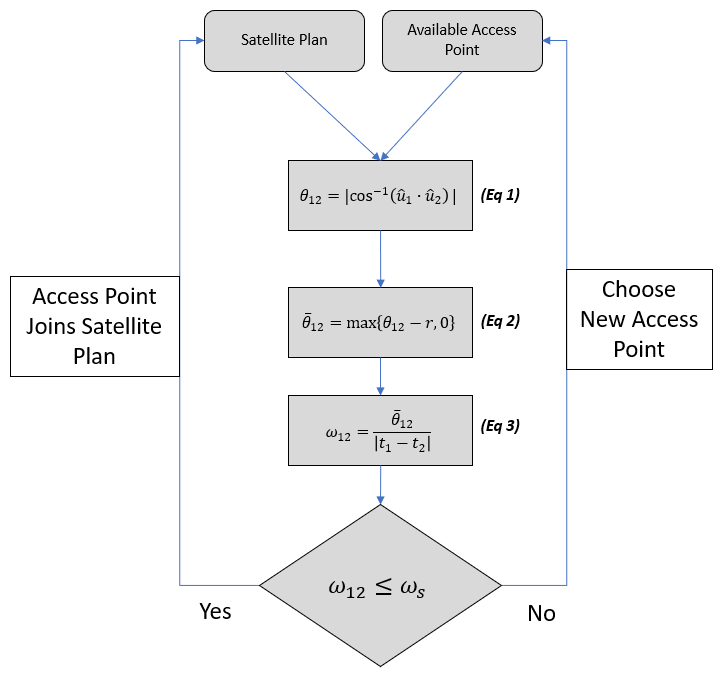


Figure . Planning Flow Chart

We decided to use a Center to Edge slewing scheme (shown in **Figure 4**), meaning that angle between each access point is calculated by subtracting the cone angle *(r)*. Where is adjusted to zero if the value is negative, meaning the point lies inside the FOV cone, as seen in **Figure 3**, **Equation 2**.

A graph of a function

Description automatically generated

Figure 4. Center to Edge Example

From there the slewing rate can be calculated by dividing by the difference in time between the points, as seen in **Figure 3**, **Equation 3**. If the calculated slewing rate is less than the slewing rate limit of the spacecraft, in our case (1 degree/s), then the point is added to satellite plan, and the bin associated with that access point is considered imaged. If not, then we chose another access point, until we find one that works. This process will compare each evaluated point against every chosen point, since we are iterating by bin scarcity instead of by time.

The slew rate calculation determines the speed at which the satellite must rotate to reorient its Field of View (FOV) between two access points. The satellite moves the center of its FOV cone from the initial access point (Point 1) to a point where the edge of the cone overlaps with the second access point (Point 2). This process evaluates each access point to determine if it is physically achievable.

#### Problem 2: Computation Time

Most of our targets are concentrated within a specific region of Earth, resulting in significant periods with no data collection. This pattern allows us to break down the planning problem into smaller, more manageable subproblems. To leverage this, we designed our planner to perform calculations both per satellite and within defined intervals, simplifying the computational load.

Each interval corresponds to an orbital revolution during which access occurs. We identified these intervals by analyzing the data for large time gaps (approximately 3000-second jumps between consecutive data points), which enabled us to classify data into interval bins and focus calculations only on relevant segments. **Figure 5** illustrates this interval subdivision.

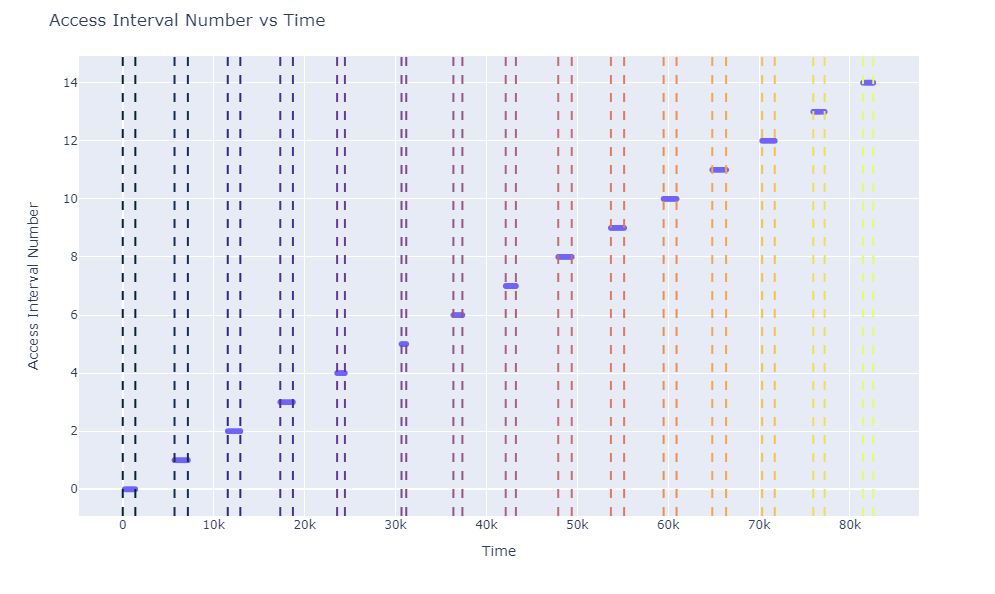


Figure 5. Access Intervals for One Satellite

### Results

The results of the updated pointing algorithm demonstrate its efficiency in capturing bin/target combinations in significantly less time. By subdividing access intervals and prioritizing those based on the scarcity of available line-of-sight opportunities, the algorithm greatly reduces computation time. In some instances, this reduction is from around 7 hours to just 5 minutes. The algorithm enables the satellite constellation to systematically target each of the 324 angle bins across all ground targets, ensuring optimal coverage.

To highlight the improvements, we compared this updated approach with a previous version that used a suboptimal planning method. The old method required satellites to pass through nadir between each target, resulting in significant coverage gaps and unobserved bins, particularly for targets needing precise positioning, as shown in **Figure 6**.

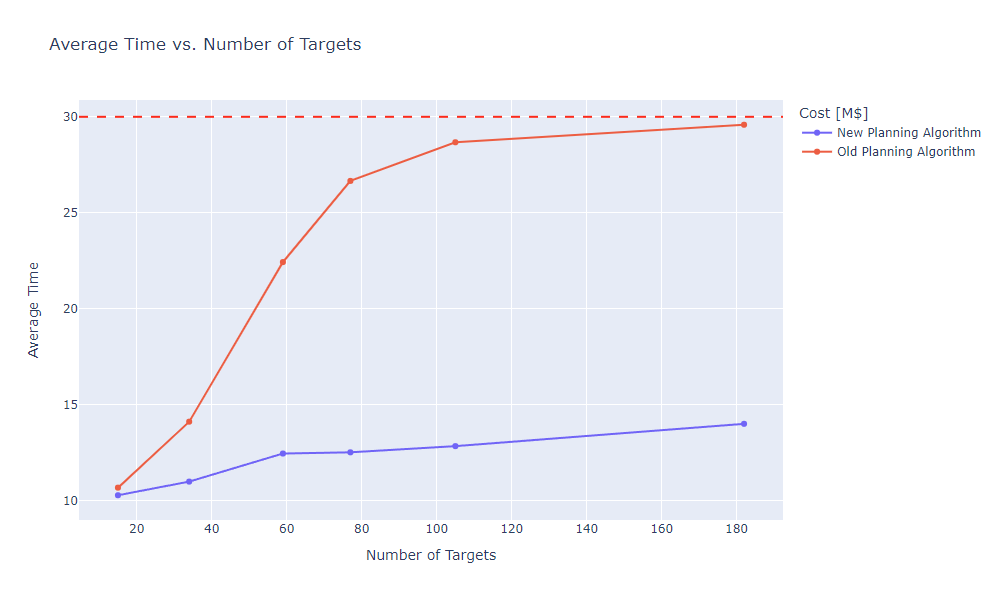


Figure 6. Planning Algorithm Comparison

The new planning algorithm offers a more scalable and reliable approach for scheduling satellite observations across varying target counts. It minimizes the average time per target, adapts more effectively to high-demand scenarios, and avoids the inefficiencies of the old nadir-passing strategy. This makes it a superior choice for maximizing target coverage while drastically reducing observation time.