Mission Design Engineer - Assignment

Ajinkya Suranjan Salve

Question 1

Track the object with two different trackers, **space-based tracker** and **ground-based tracker** for one day window starting from the epoch of the tracker.

A detection (or crossing) event is defined as a time interval during which the object satisfies specific geometric and illumination constraints relative to the tracker.

Solution Approach and Algorithm

The solution employs a time-marching approach over a 24-hour window beginning at the tracker epoch. At each discrete time step, the relative geometry between the tracker and the target is evaluated.

Propagation Strategy

- **Target Object Propagation (SGP4)**: The target space object is propagated using the SGP4 model, as its orbit is defined by Two-Line Element (TLE) data generated from observational tracking and catalog maintenance. SGP4 is a semi-analytical perturbation model derived from Brouwer–Lyddane theory and is specifically formulated to account for the dominant perturbations acting on Earth-orbiting objects. The equations of motion implicitly include the effects of Earth's non-spherical gravity field (primarily the J_2 term with additional secular and periodic corrections), atmospheric drag, and Earth rotation.

  SGP4 is valid for near-Earth orbits with orbital periods less than approximately 225 minutes, corresponding to altitudes below roughly 5875 km for circular orbits. Since the target object resides in Low Earth Orbit (LEO), its orbital period lies well within this validity range. Consequently, SGP4 provides a physically consistent and computationally efficient means of propagating the target state while preserving agreement with catalog-maintained orbital elements.

- **Space-Based Tracker Propagation (Two-Body Keplerian):** The space-based tracker is propagated using a two-body Keplerian model, assuming a central gravitational field dominated by the Earth. This approach is appropriate because the tracker's orbital elements are explicitly defined, the analysis duration is limited to a single day, and high-precision perturbation data (e.g., drag coefficients, attitude-dependent forces) are not available. Over such a short time window, higher-order perturbations introduce negligible deviations relative to the sensor field-of-view and detection geometry. The Keplerian model therefore provides a computationally efficient and sufficiently accurate representation of the tracker's motion for comparative access and visibility analysis.

Programming Logic for Object Detection (GitHub code)

At each time step:

1. The relative position vector between tracker and target is computed.
2. Angular separation between the sensor boresight and the target direction is evaluated.
3. Range constraints are checked.
4. Illumination conditions are verified using an Earth-umbra shadow model.

A crossing event is recorded when the target enters the sensor FOV/FoR and ends when it exits. Event-level quantities such as duration and minimum range are accumulated.

Algorithm Flowchart

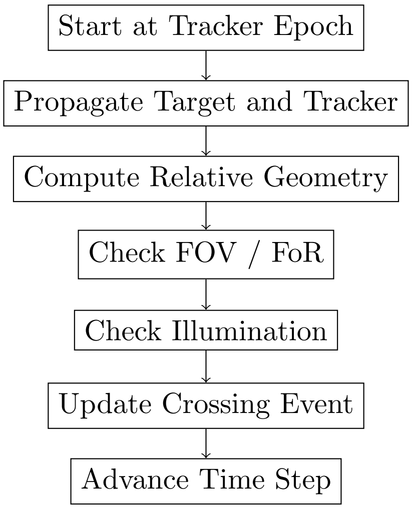


Figure 1: Algorithmic flow for crossing detection

Results

The Python implementation was executed for a full 24-hour window starting from the tracker epoch. The detected crossing events are summarized in Table 1.

Table 1: Detected crossing events for space-based and ground-based trackers

Start Time (UTC)	End Time (UTC)	Duration (s)	Min Range (km)	Sunlit	Detectable	Tracker Type	Station Night
2025-09-01 00:29:35	2025-09-01 00:29:45	10	96.22	TRUE	TRUE	Space-based	–
2025-09-01 00:29:55	2025-09-01 00:31:25	90	518.06	TRUE	TRUE	Ground-based	TRUE
2025-09-01 02:04:25	2025-09-01 02:04:35	10	617.83	TRUE	FALSE	Ground-based	FALSE
2025-09-01 05:11:45	2025-09-01 05:12:00	15	617.02	TRUE	FALSE	Ground-based	FALSE
2025-09-01 06:44:55	2025-09-01 06:46:30	95	517.55	TRUE	FALSE	Ground-based	FALSE

Discussion

The results presented in Table 1 indicate that multiple geometric crossing events were identified within the one-day analysis window for both space-based and ground-based trackers. Among these, a subset of events satisfied the additional visibility constraints, leading to successful detection opportunities.

For the space-based tracker, the detected crossing event occurred at a short relative range and under sunlit conditions, demonstrating the effectiveness of a velocity-pointing sensor architecture for close-proximity observations. The limited number of detections is attributed to the narrow field-of-view and the imposed range constraint of 1000 km, which significantly restrict the observable volume.

In the case of the ground-based tracker, several crossing events were observed; however, only one event satisfied both the target illumination and station night-time conditions required for detection. This highlights the strong dependence of ground-based observations on local lighting conditions, particularly for high-latitude stations where extended daylight or twilight periods are common.

Overall, the results illustrate the complementary nature of space-based and ground-based tracking systems. While space-based sensors provide consistent geometric access independent of ground lighting conditions, ground-based sensors are constrained by diurnal cycles but can still offer valuable detection opportunities when illumination conditions are favorable. The observed detection events are consistent with the imposed physical and operational constraints and validate the robustness of the implemented detection framework.

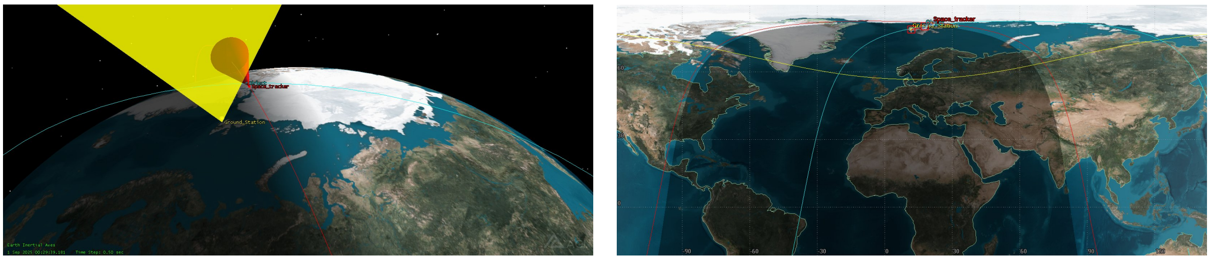
Validation with STK

To validate the correctness of the Python-based implementation, the complete tracking scenario was independently recreated in Systems Tool Kit (STK), which is widely used for high-fidelity space mission and sensor analysis. STK access, lighting, and visibility analyses were generated for both the space-based and ground-based trackers. The corresponding STK output reports can be accessed here:

A direct comparison between the Python-generated results and the STK reports shows strong agreement in both crossing and detection events. All detection intervals identified by the Python implementation fall within the corresponding STK access windows, and the sequence of events is consistent across both tools.

Minor discrepancies of a few seconds were observed in the reported start and end times of certain events. These differences are expected and arise primarily due to the discrete time-stepping used in the Python implementation, whereas STK employs adaptive event-finding algorithms with sub-second resolution. Additional contributing factors include differences in shadow boundary modeling and frame transformation precision.

Despite these small timing offsets, the overall detection behavior, event durations, and access trends remain consistent. This close agreement confirms the physical validity of the Python-based approach and demonstrates that it provides results comparable to a high-fidelity commercial tool for the analyzed scenario.



(a) 3D visualization of object, space-based tracker, and ground-based tracker

(b) 2D ground track visualization

Figure 2: STK visualization of space-based and ground-based tracking geometry illustrating sensor field-of-view, illumination conditions, and relative motion of the target object.