Week 5 Report

Dhanush Balusa

June 18, 2025

Research

Space-Track.org's Collision Data

- https://www.space-track.org/documents/SFS_Handbook_For_Operators_V1.7.pdf
- Public conjunctions available on Space-Track are generated by the United States Space Command (USSPACECOM), specifically by the 19th Space Defense Squadron (19 SDS).
- This is done through daily screenings of the U.S. high-accuracy satellite catalog, which is maintained by the 18th Space Defense Squadron (18 SDS).
- When a potential close approach meets or exceeds established risk thresholds, a Conjunction Data Message (CDM) is automatically generated, reviewed, and published to Space-Track.

Other resources:

- https://www.space-track.org/documents/CSM_Guide.pdf
- https://www.nasa.gov/cara/frequently-asked-questions/#:~:text=18%20SDS%20operators%20support%20the,posed%20by%20the%20close%20approach.
- From test case #2: https://ntrs.nasa.gov/api/citations/20070007324/downloads/20070007324.pdf

Code Changes

Based on the feedback from last week's team meeting, I decided to make the following changes:

1. **My first task was to check if other close approaches were modeled accuratley.**
   To do this, there was a code change needed to ensure that the TLE is correctly fetched because for some reason the code was erroring with 3 lines of TLE data.

   When I first wrote the code, fetching the TLE was erroring because of the 3 lines. This change was made to ensure that the TLE data is correctly formatted and that the `EarthSatellite` object is created with the correct parameters (by adding the name).

   ```
   tle = [name] + tle_lines
   return EarthSatellite(tle[1], tle[2], tle[0])
   ```

   I had to change the `fetch_and_create_satellite` function to handle cases where the TLE data might not be available or is incomplete. This ensures that the code does not break when trying to create an `EarthSatellite` object with insufficient data. The code now checks if the TLE lines are less than 2, and if so, it constructs the TLE with the name and the available lines. This prevents errors when trying to create an `EarthSatellite` object with incomplete data. I believe Space-Track.org has changed their API to return TLE data in a different format, which is why the old code errored out.

   ```
   if len(tle_lines) < 2:
       tle = [name] + tle_lines
       return EarthSatellite(tle[0], tle[1])
   ```

   Using the updated code, I was able to successfully fetch the TLEs.

Using Space-Track.org, I was able to find conjunction data for the following test cases from `https://www.space-track.org/#/conjunctions`:

- Test Case #1: COSMOS 2294 & COSMOS 2288
- Test Case #2: FENGYUN 1C DEBRIS & NOAA 6
- Test Case #3: COSMOS 468 & DMSP 5D-2 F13 DEBRIS
- Test Case #4: COSMOS 1816 & M-4S Rocket Body

Test case #1 perfectly matched Space-Track with the Time of Closest Approach (TCA) of 2025-06-20 16:20:00 UTC. This test case is between 2 COSMOS satellites and fetches TLE data 4 days before the alert was created.
URL: `https://www.space-track.org/basicspacedata/query/class/cdm_public/cdm_id/1057414232/format/kvn/emptyresult/show`

```
CDM_ID                                   =1057414232
CREATED                                  =2025-06-18 00:25:20.000000
EMERGENCY_REPORTABLE                     =Y
TCA                                      =2025-06-20T16:20:00.347000
MIN_RNG                                  =2398
PC                                       =
SAT_1_ID                                 =23398
SAT_1_NAME                               =COSMOS 2294 (GLONASS)
SAT1_OBJECT_TYPE                         =PAYLOAD
SAT1_RCS                                 =LARGE
SAT_1_EXCL_VOL                           =5.00
SAT_2_ID                                 =23205
SAT_2_NAME                               =COSMOS 2288 (GLONASS)
SAT2_OBJECT_TYPE                         =PAYLOAD
SAT2_RCS                                 =LARGE
SAT_2_EXCL_VOL                           =5.00
```

Figure 1: Test Case #1: COSMOS 2294 & COSMOS 2288

```
Fetching TLE for COSMOS 2294 (GLONASS)...
Fetching TLE for COSMOS 2288 (GLONASS)...
COSMOS 2294 (GLONASS) Epoch: 2025-06-15T19:37:09Z
COSMOS 2288 (GLONASS) Epoch: 2025-06-14T18:51:18Z

  Near Miss Detected Between COSMOS 2294 and COSMOS 2288:
    - Closest Distance:      3.640 km
    - Approach Speed:        6.562 km/s
    - Approach Angle:        138.606 degrees
    - Time (UTC):            2025-06-20T16:20:00Z
```

Figure 2: Output of Test Case #1

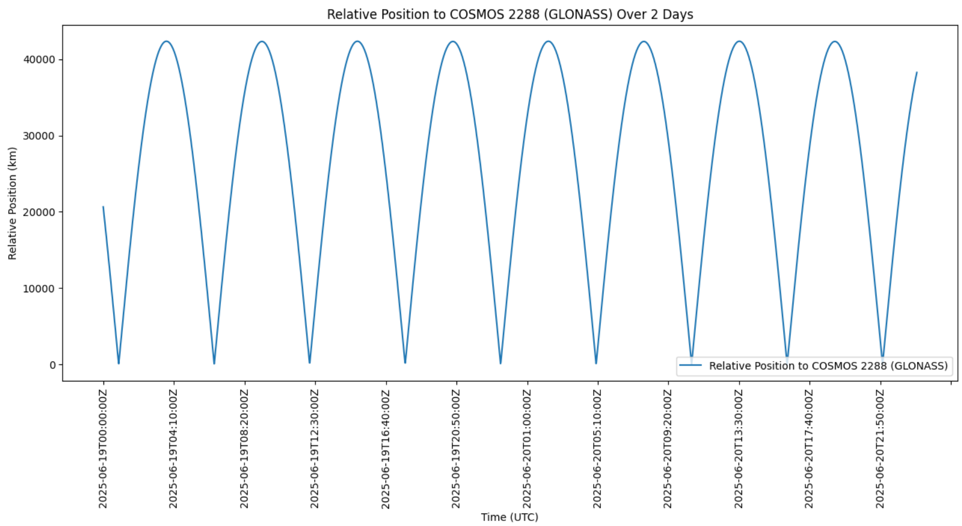

Figure 3: Graph of Test Case #1

Test case #2 perfectly matched Space-Track with the Time of Closest Approach (TCA) of 2025-06-19 19:02:17 UTC. This test case is between a debris from a Chinese anti-missle test and a NOAA satellite and fetches TLE data 1-2 days before the alert was created. Initially my calculated TCA was off by 1 hour, but once I increased the time steps, it worked perfectly.
URL: `https://www.space-track.org/basicspacedata/query/class/cdm_public/cdm_id/1057084854/format/kvn/emptyresult/show`

```
CDM_ID                                 =1057084854
CREATED                                =2025-06-17 14:52:12.000000
EMERGENCY_REPORTABLE                   =Y
TCA                                    =2025-06-19T19:02:16.881000
MIN_RNG                                =587
PC                                     =0.0006361094
SAT_1_ID                               =36182
SAT_1_NAME                             =FENGYUN 1C DEB
SAT1_OBJECT_TYPE                       =DEBRIS
SAT1_RCS                               =SMALL
SAT_1_EXCL_VOL                         =1.00
SAT_2_ID                               =11416
SAT_2_NAME                             =NOAA 6
SAT2_OBJECT_TYPE                       =PAYLOAD
SAT2_RCS                               =LARGE
SAT_2_EXCL_VOL                         =5.00
```

Figure 4: Test Case #2: FENGYUN 1C DEBRIS & NOAA 6

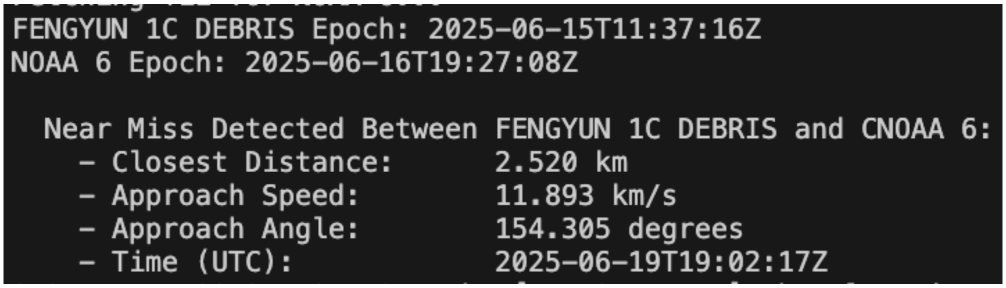

Figure 5: Output of Test Case #2

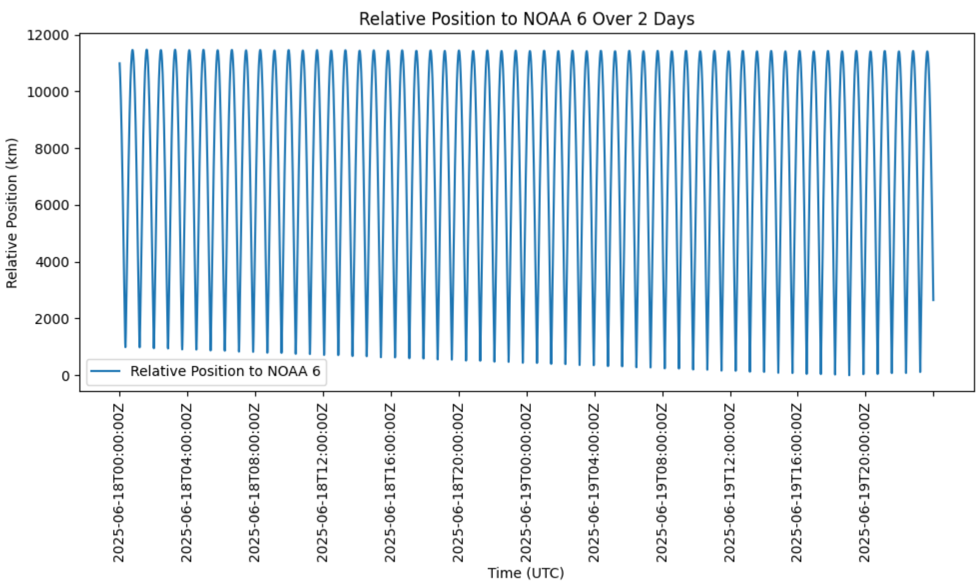

Figure 6: Graph of Test Case #2

Test case #3 wasn't even able to be fetched from Space-Track.org, as it was giving an error that the TLE data was not available for the COSMOS satellite and the debris. URL: `https://www.space-track.org/basicspacedata/query/class/cdm_public/cdm_id/1029160361/format/kvn/emptyresult/show`

```
CDM_ID                                  =1029160361
CREATED                                 =2025-05-19 07:31:30.000000
EMERGENCY_REPORTABLE                    =Y
TCA                                     =2025-05-21T04:15:02.632000
MIN_RNG                                 =815
PC                                      =0.0003446535
SAT_1_ID                                =5705
SAT_1_NAME                              =COSMOS 468
SAT1_OBJECT_TYPE                        =PAYLOAD
SAT1_RCS                                =LARGE
SAT_1_EXCL_VOL                          =5.00
SAT_2_ID                                =40612
SAT_2_NAME                              =DMSP 5D-2 F13 DEB
SAT2_OBJECT_TYPE                        =DEBRIS
SAT2_RCS                                =SMALL
SAT_2_EXCL_VOL                          =1.00
```

Figure 7: Test Case #3: COSMOS 468 - DMSP 5D-2 F13 DEBRIS

```
None Epoch: 1999-12-31T00:00:00Z
None Epoch: 1999-12-31T00:00:00Z

  Near Miss Detected Between COSMOS 468 and DMSP 5D-2 F13 DEBRIS:
    - Closest Distance:      nan km
    - Approach Speed:        nan km/s
    - Approach Angle:        nan degrees
    - Time (UTC):            2025-05-20T00:00:00Z
```

Figure 8: Output of Test Case #3

Test case #4 also wasn't able to be fetched from Space-Track.org, as it was giving an error that the TLE data was not available for the COSMOS satellite and M-4S Rocket Body.
URL: `https://www.space-track.org/basicspacedata/query/class/cdm_public/cdm_id/1056085317/format/kvn/emptyresult/show`

```
CDM_ID                                   =1056085317
CREATED                                  =2025-06-16 14:11:21.000000
EMERGENCY_REPORTABLE                     =Y
TCA                                      =2025-06-18T21:47:51.472000
MIN_RNG                                  =445
PC                                       =0.0019282
SAT_1_ID                                 =17359
SAT_1_NAME                               =COSMOS 1816
SAT1_OBJECT_TYPE                         =PAYLOAD
SAT1_RCS                                 =LARGE
SAT_1_EXCL_VOL                           =5.00
SAT_2_ID                                 =5126
SAT_2_NAME                               =M-4S R/B
SAT2_OBJECT_TYPE                         =ROCKET BODY
SAT2_RCS                                 =MEDIUM
SAT_2_EXCL_VOL                           =3.00
```

Figure 9: Test Case #4: COSMOS 1816 & M-4S Rocket Body

```
None Epoch: 1999-12-31T00:00:00Z
None Epoch: 1999-12-31T00:00:00Z

  Near Miss Detected Between COSMOS 1816 and M-4S Rocket/Body:
    - Closest Distance:      nan km
    - Approach Speed:        nan km/s
    - Approach Angle:        nan degrees
    - Time (UTC):            2025-06-18T00:00:00Z
```

Figure 10: Output of Test Case #4

I went back to the other test cases and found that the TLE data was not available for anything, even from previous weeks' code. This most likely means Space-Track.org has limited my requests to pull TLE data from their API.

2. **My second task was to check the approach angle math.**

   Previous week's approach angle code:
   The code calculates the **approach angle** between two satellites by measuring the angle between their *relative position vector* and the *velocity vector* with respect to the first satellite (Sat1).

   This is computed using the dot product formula:

$$\theta = \arccos\left(\frac{\vec{r} \cdot \vec{v}}{|\vec{r}||\vec{v}|}\right) \tag{1}$$

   where:

   - $\vec{r}$ is the relative position vector between the two satellites (from Sat2 to Sat1),

- $\vec{v}$ is the relative velocity vector (how fast and in what direction Sat1 is moving relative to Sat2),
- θ is the angle between the two vectors.

This approach angle provides insight into the geometry of the encounter:

- An angle near 0° indicates a *head-on approach*.
- An angle near 90° suggests a *tangential or glancing pass*.
- An angle near 180° implies the satellites are moving in *opposite directions*.

Proposed approach angle code:

The proposed code calculates the **approach angle** between two satellites by measuring the angle between the *velocity of Sat1* and the *velocity of Sat2*.

This is computed using the dot product formula:

$$\theta = \arccos\left(\frac{\vec{v_1} \cdot \vec{v_2}}{|\vec{v_1}||\vec{v_2}|}\right) \tag{2}$$

where:

- $\vec{v_1}$ is the velocity vector of Sat1,
- $\vec{v_2}$ is the velocity vector of Sat2,
- θ is the angle between the two vectors.

This approach angle provides insight into the geometry of the encounter:

- An angle near 0° indicates a *moving together*.
- An angle near 90° suggests a *right angle*.
- An angle near 180° implies the satellites are moving in *head-on collisin*.

This change was made to ensure that the approach angle is calculated based on the velocities of the satellites, which is more relevant for understanding their relative motion during a close approach.

Next Steps

My notes:

- Look into why test case 3 and 4 are not working.
- Test the rest of the cases from Space-Track.org throughout the week to see if I get access again.
- Test the proposed approach angle code.

Dr. Fan's feedback:

1. Read approach angle literature. Does not necessarily have to be related to space but can be related to other fields like cars.
2. Other research on space collision conditions consequence of a collision, break-up model, hypervelocity impact.
3. Conjunction analysis:
   - How do other people propagate orbit to predict collision?
   - What other parameters are important in conjunction analysis? Propagation techniques, something else other than TLE data?

End goals:

- Loop through all the TLEs in the LEO category and calculate the approach conditions for each satellite if it is within a certain distance (e.g., 100 km) of the ISS.
- Make my code a usable function so Catherine can just call it. Some input $\rightarrow$ Output (approach speed and angle).