ASSET (AScent SafETy) – a new Flight Dynamics Service for the Safety of Ascent Trajectories and Injection Orbits

Giuseppe Di Campli Bayard de Volo(1), Ralph Kahle(1), Andrea Zollo(1), Lars Sprengelmeyer(1), Simon Gruber(1)

(1) German Aerospace Center (DLR), German Space Operations Centre (GSOC)
Münchener Str. 20, 82234 Weßling, Germany
Email: {giuseppe.dicamplibayarddevolo, ralph.kahle, andrea.zollo, lars.sprengelmeyer, simon.gruber}@dlr.de

The sharp increase of Earth-orbiting objects and, subsequently, of possible in-orbit collisions, poses several challenges not only in operating satellites, but also in launching new ones. Indeed, a close conjunction between a rocket stage (or launched payloads) and a resident object can happen in the early orbits and even during ascent, before the new satellites are inserted into the catalogue of objects and screened on a routine basis. To mitigate this risk, the Flight Dynamics team of DLR's German Space Operations Centre (GSOC) is developing a service for launch collision avoidance, named ASSET (AScent SafETy), capable of providing an assessment on the safety of launch trajectories in terms of probability of collision with resident objects. The software is composed by several computational boxes, in part directly inherited from the in-house, operational Collision Avoidance (COLA) software, in part newly developed for the specific application. This paper outlines the fundamental principles of ASSET and its software architecture. In addition, the first prototype is presented, starting from the implemented algorithms, numerical results and findings. Finally, the software limitations and points for improvement are described.

I. INTRODUCTION

With the creation of mega-constellations, like Starlink or OneWeb, the number of space objects and therefore close conjunctions in orbit has reached a level that requires new approaches to allow for a safe access to space [1]. In the past, the probability of a collision between a rocket during launch phase and a space object was so low that no space agency nor launch provider was actually tackling the problem. In some cases, and only starting in the past decade, screenings were performed between the launch trajectory and the *manned* space vehicles only, like in the case of the launches performed from Kourou: CNES developed a dedicated tool for this purpose, named ARCL (Collisions Risk Assessment at Launch), which has been operational since 2010 [2]. In more recent years, the risk of a close encounter with a space object during launch and first orbits is such that the main agencies equipped themselves with dedicated tools and procedures to screen for possible conjunctions ahead of the day of lift-off. This is the case of the “Launch Conjunction Assessment” of the 19th Space Defence Squadron (19SDS) in the US [3], which is a mandatory screening for all the launches performed from US launch bases that exceed 150 km in altitude [4][5]. In Europe, a similar analysis has also been reported in [6] for rocket launches from the UK.

A real-life operational example that shows the importance of this type of screenings is the launch of the Indian moon lander Chandrayaan-3 onboard the LVM-3 heavy-lift rocket in July 2023, which was postponed by 4 seconds due to a foreseen high risk of in-orbit collision.

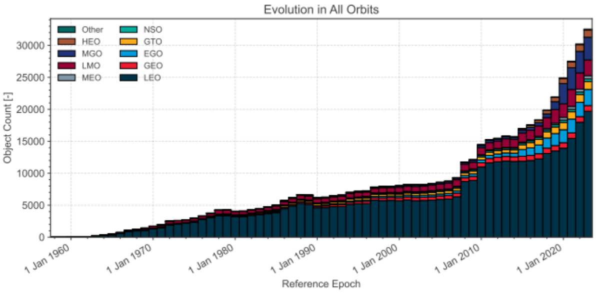


Figure 1 – Evolution of the number of Resident Objects [1]

The ASSET service, described in this paper, is the DLR’s response to this new need. The scope of ASSET, developed in the framework of DLR’s Responsive Space Capabilities program, is to provide the launch operator with an assessment on the safety of launch trajectories, during a given launch window, in terms of probability of collision. In the next sections, the current prototype of the software will be presented, from the idea behind the software architecture, to the computational boxes and algorithms. At the end of the paper, the results of some simulation scenarios will be shown, together with the current software limitations and points for improvement.

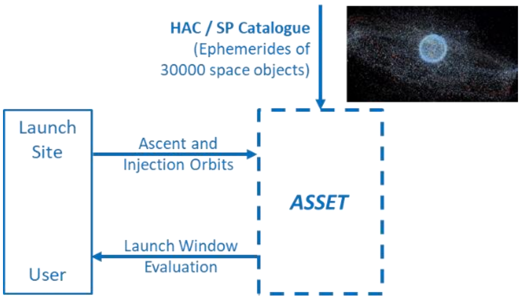


Figure 2 – ASSET service schema

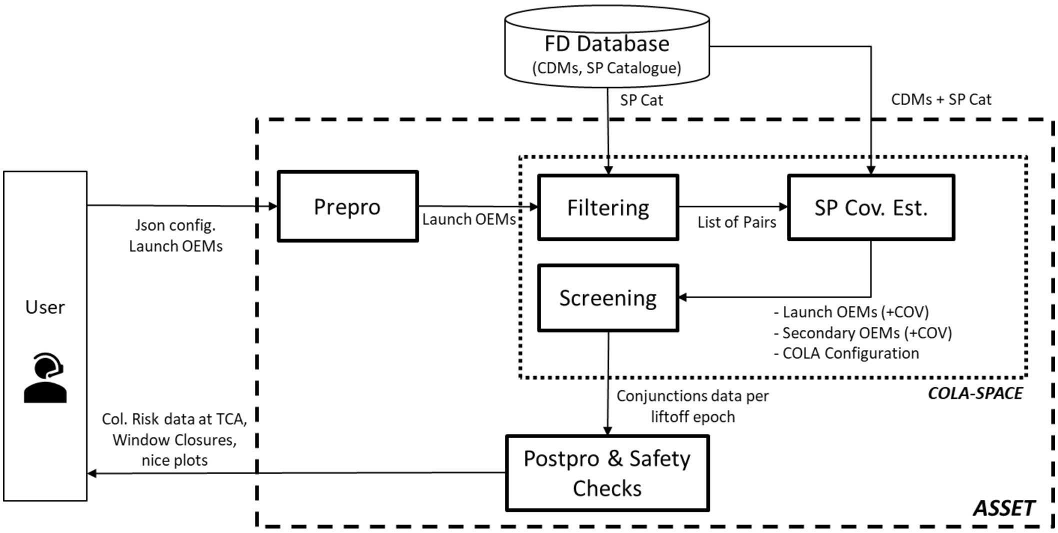


Figure 3 – ASSET Software Architecture

II. IDEA AND SOFTWARE ARCHITECTURE

The idea behind ASSET is to make use – as much as possible – of the operational collision avoidance software COLA, developed and operated by the flight dynamics team at GSOC for the satellite missions. Therefore, the functionalities of the COLA software have been expanded to create a self-standing software capable of different modes of screening, going from a “1-vs-all” to a “all-vs-all” screening. To do that, the Special Perturbation (SP) catalogue that GSOC receives from the 19SDS (19th Space Defence Squadron) via GSSAC (German Space Situational Awareness Center) is fetched and filtered to exclude all the objects that cannot cause close encounters. For those objects that pass the filter, an ephemeris is first created including the covariance record and then screened against the launch trajectory. This extended version of the COLA software has been named COLA-SPACE (SP catalogue Analysis for Conjunction Evaluation), as drawn in the simplified schema of *Figure 3*.

ASSET is created around COLA-SPACE, to act as an interface towards the final user, which can be for example a launch provider. For this purpose, it was necessary to develop a pre-processing toolbox, capable of ingesting trajectory data and a service setup file, with several modes of use and options that make the service as flexible as possible. At the end, a post-processing tool has been added to collect all the screening results and create the desired output, containing the probability of collision of each launch trajectory, the launch window closures, and so on. The output is returned to the user to provide support in the launch scheduling.

III. USER REQUEST AND TRAJECTORY PROCESSING

The interface to the user has been defined in a number of data files, including a setup file and an ephemeris file, containing the launch trajectory. The setup file is a text file in the “.json” format and contains all the options for the service call and the rocket/satellite data, in details:

- ASSET service setup options:
  - Minimum miss-distance for the screening [km]
  - Maximum probability of collision (PoC)
  - Duration of screening [days]
  - Number of deployed payloads
  - Option to screen a launch window or a single launch trajectory
  - Start epoch of the launch window
  - Stop epoch of the launch window
  - Minimum number of steps for the launch window screening
- Rocket / Payloads (P/L) data:
  - Ephemeris file name
  - Mass [kg]
  - Area/Dimensions [m^2]
  - Drag coefficient Cd
  - Solar radiation coefficient Cr
  - Object name and ID
  - Separation epoch (for P/L only)
  - Separation Delta-V with RTN components [m/s] (for P/L only)

In case the user selects the option to screen a single launch trajectory without P/Ls, only the ephemeris of the rocket is required. This ephemeris is propagated in the future for the number of days indicated in the setup file, the state vectors are transformed in EME2000 and the covariance matrices in RTN components.

If a certain number of P/Ls is to be included, during the propagation of the rocket orbit, the separation of each P/L is simulated applying the Delta-V of the separation at separation epoch, in the direction indicated again in the setup file in RTN components. In this way, a number of orbits are obtained starting from the rocket upper stage orbit. If, in addition, a launch window screening is requested by the user, the rocket and P/Ls trajectories are transformed in an Earth-fixed frame and rotated by simply shifting the lift-off epoch along the launch window. In this way, a whole group of launch trajectories is obtained, with the total number of trajectories being:

$$N_{trjs} = (N_{PL} + 1) * n_{step} \qquad (1)$$

where N_{trjs} is the total number of trajectories, N_{PL} is the number of injected payloads (+1 for the rocket upper stage, if not directly de-orbited) and n_{steps} is the number of time steps along the launch window. *Figure 4* shows an example of a group of trajectories (rocket upper stage only) spanning over the launch window. The naming convention of the objects is *<yymmddhhmmsssss-object_name>*, to indicate the lift-off epochs (in the image, spaced by 1 minute) and the object name.

Therefore, the time-filter evaluates the crossing times and removes pairings, that are out of sync. In contrast, the Sieve-filter consists of another subset of filters, five in total. While Hoots filters can be applied knowing only the orbital elements, Sieve makes use of the ephemeris of both objects: it performs a step-by-step epoch comparison by computing whether the two can possibly meet within one-time step, relying on conservative assumptions and simple motion laws like the escape velocity. If the epochs of two objects do not match, Lagrange interpolation can be applied in addition. The Sieve sub-filter sequence is described in [11]. Both filter variations can produce so called type I and type II errors, meaning that uncritical pairings are forwarded to the screening process (false positive), while hazardous pairings are unintentionally discarded (false negative).

Figure 4 – Pre-processed Launch Trajectories

Clearly the number of launch trajectories, which will be the "primaries" in the screening process, can be quite high: considering for example a launch window of 10 minutes, to be screened at every second, with 4 payloads inserted in orbit, the total number of primaries is 3,000. On the other hand, the number of potential secondary objects coming from the SP catalogue currently (as of April 2024) is circa 28,000.

IV. CATALOGUE FILTERING

To reduce the number of primary-secondary pairings, and thus to minimize the computationally intensive screening, filters are applied in a certain order. The first two are pure geometrical and independent of actual object locations, namely the apogee-perigee-filter and the path-filter, as described by Hoots [10]. They filter out pairs that cannot interfere with each other due to their orbital plane shapes and orientations in space. To define whether elliptical paths may lie within a critical distance, filter thresholds need to be set. For the third filter, which is optional, the user can choose between a time-filter or a Sieve-filter [11]. The former is also described by Hoots [10] and takes the actual satellite position into account. Although paths can be close, the two objects must simultaneously pass through the line of intersection to have a potential critical conjunction.

The number of errors strongly depends on the selected distance thresholds for each filter. The tighter the thresholds, the less type I, but the more type II errors. And vice versa for more conservative thresholds. The goal is to keep the type II error number as small as possible, to not miss on any conjunction. Therefore, precisely chosen thresholds are key. To sum up, the apogee-perigee filter and the path filter are low effort computations, and thus always included. Tests have shown that the latter's threshold should be set more conservative. The time-filter's disadvantage is its vulnerability to orbital decay. Objects in LEO are particularly affected by atmospheric drag, which can be difficult to estimate for unknown objects, but has a significant impact on the orbital period and thus on the crossing times. Sieve reduces the number of pairings to be assessed further the most, but comes with higher computational effort. Especially when the epochs of two ephemerides do not match initially and need to be adapted via Lagrange. For this reason, the overall filter method (written in Python) was kept as generic as possible, with the following parameters: filters to be used, filter thresholds, file directories and some third filter specific settings. This allows flexible adaptation to launch trajectory changes, long and short term, in case a re-filtering is necessary.

V. COVARIANCE ESTIMATION

As previously stated, the GSOC-FD System receives the SP catalogue as orbit ephemeris data. Although this dataset is valuable and could potentially facilitate the development of services like ASSET, it lacks the associated orbit uncertainty information. This limitation prevents a direct assessment on the risk of collision with the catalogued objects. As a workaround to this limitation, it has been proposed in [12] to connect a set of synthetic orbital error covariance matrices to a given SP ephemeris through a regression analysis of historical Conjunction Data Messages (CDMs) from past conjunctions involving GSOC satellites [13][14]. Being more specific, the orbital errors of all the encountered secondary objects are grouped into different orbital classes to mitigate the strong correlations of the covariance with parameters such as solar flux, object dimensions, perigee altitude, eccentricity, and orbit inclination. This classification process is designed to group together similar CDMs that share the dependencies mentioned above, and to estimate the expected 1-sigma position errors using optimized curve-fitting techniques. Through the assessment of curve fitting coefficients for a specific orbit class, a covariance matrix can be generated for any future prediction time. *Figure 5* provides an example of how the raw position uncertainties are fitted for a given orbital class. Specifically, for every CDM, the positional uncertainty is associated with its respective prediction time. The relationship between the covariance and the propagation time can then be determined using a non-linear least squares curve fitting method.

Whenever a synthetic covariance matrix needs to be associated to an SP ephemeris that belongs to an orbital class for which no data is available, a neighbouring class containing a sufficient number of CDMs is selected. The fitting coefficients of this last are then utilized. The selection of the neighbouring class is done in such a way that one should avoid moving along directions where the covariance exhibits significant variability.

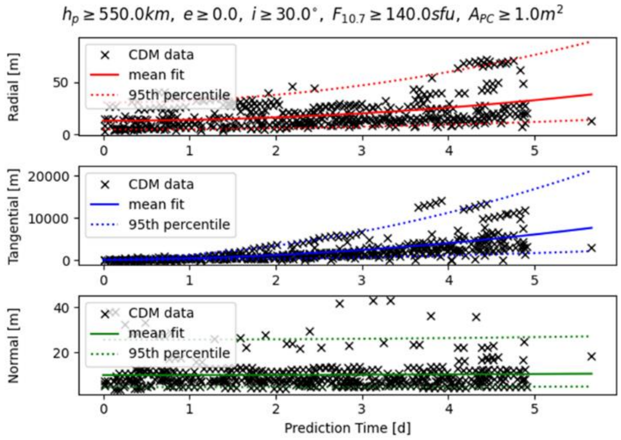


Figure 5 – Fit of 1-sigma position error in RTN for synthetic covariance generation

VI. SCREENING AND CONJUNCTION ESTIMATION

Once the filtering is performed and covariance records are associated to the secondary objects of the SP catalogue, the possible conjunction pairs need to be screened in more detail for possibly relevant events. Therefore, each primary object is screened against its set of secondaries and key parameters like miss distance, time of closest approach (TCA) and maximum and true probability of collision (PoC) calculated by the methods based on [7] and [8] are returned. For this, the in-house system COLA, which is also adopted for GSOC-operated satellite missions, is used.

The COLA software is an executable program written in Fortran, which takes file-based input information by the user. The main information, which has to be provided by the user consists of the following parameters:

- UTC start time of screening window
- Duration of screening window
- Total distance below which further conjunction assessment is performed
- Radial distance below which further conjunction assessment is performed
- Object ID and path to ephemeris file for all considered objects (primaries and secondaries)

Following the reading of the input parameters and ephemerides into internally defined software data structures, the actual screening is performed. Therefore, every primary object is screened against all relevant secondary objects in a “1vs1” manner. The screening starts by determining sets of Chebyshev coefficients for both ephemerides to obtain objects state vectors at any time during the screening window. This is followed by the calculation of the times of all nodes, where the orbital planes cross and determining a search period around each node. In this step, also nodes with a distance larger than a user defined threshold are discarded. Afterwards, local minima are calculated for every search span around a node. As a result, the TCA and the state vectors of both objects at TCA are obtained and further used to calculate other relevant parameters. Following the screening, COLA is also capable of generating several output files including Conjunction Data Messages, which will not be further evaluated, as the screening capability of COLA is the main interest for the service described in this paper. The overall screening process can also be seen in *Figure 5*.

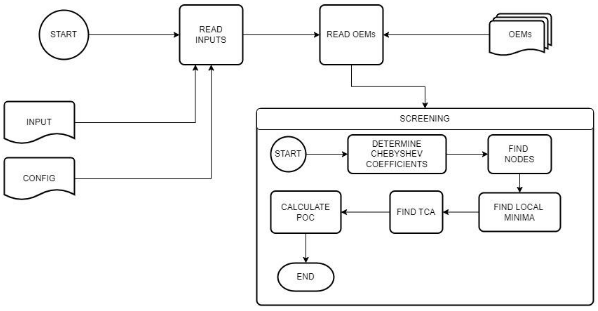


Figure 6 - Structure of COLA screening

Due to the file-based approach of COLA, it is well suited for the operational use inside GSOC, but rather inflexible when connected to another software or service, like ASSET. In order to improve this, it was attempted to wrap the existing Fortran code into a shared library and generate so called Python-wrapped files, which can be imported and called by other Python scripts. This is done via F2x, which GSOC Flight Dynamics already successfully used to wrap heritage Fortran modules and use them in modern Python applications, as described in [9].

In the scope of COLA-SPACE, the modules relevant for assessing conjunction events (matrix operations, PoC calculations) were wrapped and made callable from Python code. Additionally, a high-level interface was implemented in Python, which allows input/output handling. For verification, the file-based COLA executable was used, with inputs dynamically generated in Python.

VII. RESULTS

Going back to the process flow chart in *Figure 3*, the output of the software COLA-SPACE, once the screening is completed, is a list of risky conjunctions grouped for lift-off epoch. The list also contains, for each conjunction, a set of important parameters, namely:

- Object ID of primary object
- Object ID of secondary object
- TCA
- Maximum PoC
- Miss-distance

This list is returned to the user as file in "json" format. In addition, a summary plot (*Figure 7*) for the launch window is provided. Hereafter, an example is given for a specific test case.

The selected scenario takes into consideration a rocket launch into LEO, specifically at 500 km in sun-synchronous orbit (SSO). The screening is performed for 3 days after P/L separation and the user-selected maximum allowed PoC is 1e-5. The rocket upper stage separates two CubeSats 10 minutes apart, with opposite Delta-Vs. The user also provides a launch window of 1 hour, to be screened at every minute, which makes a total of 183 trajectories to be screened (61 per object, including the extremes of the window). It is assumed that the rocket upper stage does not perform a direct atmospheric re-entry, thus orbiting for the entire period of the screening.

The results of the ASSET run are shown in *Figure 7*. The upper bar chart represents, with colours, the level of risk of collision for each lift-off epoch: in red, the "alarm" zones, i.e. when the PoC exceeds the maximum value of 1e-5, in orange the "warning" zones with PoC of one order of magnitude lower than the threshold (in the example, 1e-6) and in green the "safe" zones for even lower PoC. This bar already gives a clear overview on the parts of the launch window to be avoided and the

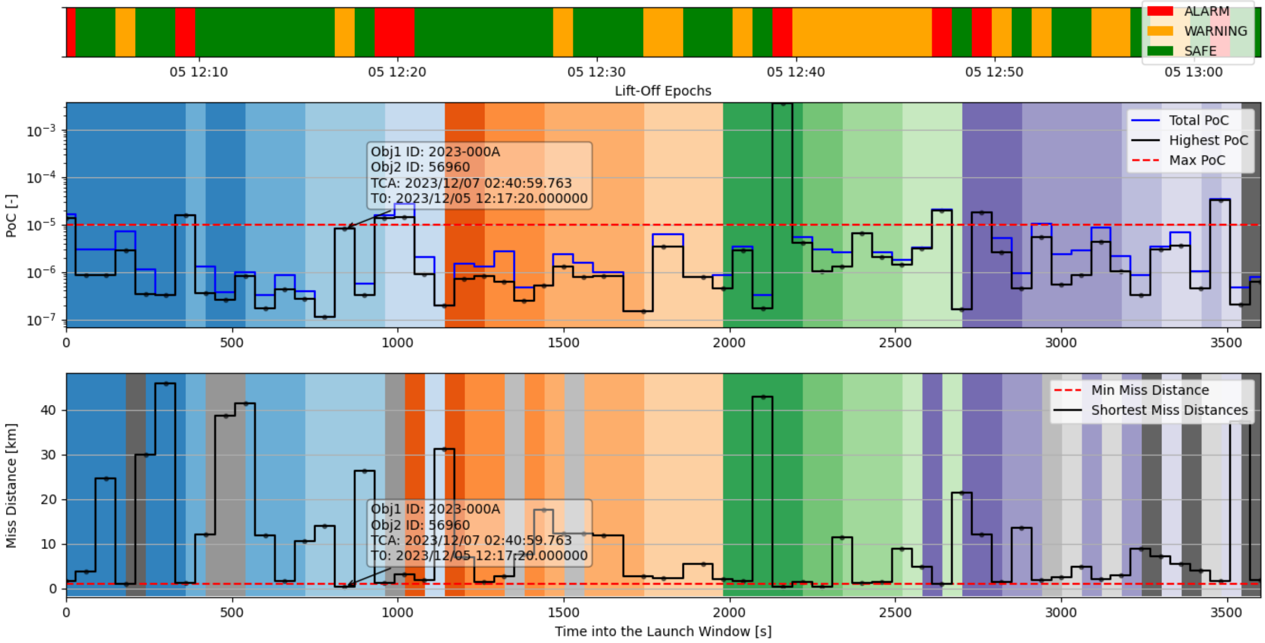


Figure 7 – ASSET output plot for a launch window of 1 hour

ones to be preferred to minimize collision risks. In the second plot, the PoC itself is shown, both for the riskiest conjunction (so highest PoC, black curve) and for the total, cumulated PoC along the screening period (blue curve). In other terms, the blue curve represents the sum of the PoC of all the conjunctions found for a specific lift-off epoch and for all the separated objects, so including the rocket and all the P/Ls. The red dashed line is the maximum PoC set by the user, meaning that every time that the black curve is higher than the red line, ASSET marks this part of the launch window as red in the bar plot at the top. In the lower plot, the absolute miss-distance is shown, together with the minimum acceptable miss-distance set by the user (dashed red line). It has to be highlighted that in this example, the miss-distance is not used for the definition of risky events, which are uniquely assessed based on the PoC. The background colours of the second and third plots define the secondary objects, i.e., each colour is assigned to a unique secondary object ID. For example, the two tooltip views in *Figure 7* show that the section coloured in "cyan" refers to the conjunction between the rocket upper stage (fictious ID 2023-000A) and the same secondary object (ID 56960), which is the riskiest in this 4-minutes interval. Also, it is interesting to see the (not so close) correlation between the maximum PoC and the lowest miss-distances, by comparing the background colours of the two plots.

Going into the numerical results, it's clear that many parts of the launch window are critical for the in-orbit conjunctions. This is mainly due to the presence of the Starlink mega constellation at around 550 km (and lower when satellites decay). Therefore, the preliminary results of the analysis performed with ASSET show that, when targeting these altitudes, launch providers should perform a pre-launch analysis to avoid the (quite) likely conjunctions until the injected P/Ls are inserted into the SP catalogue and screened on a routine basis.

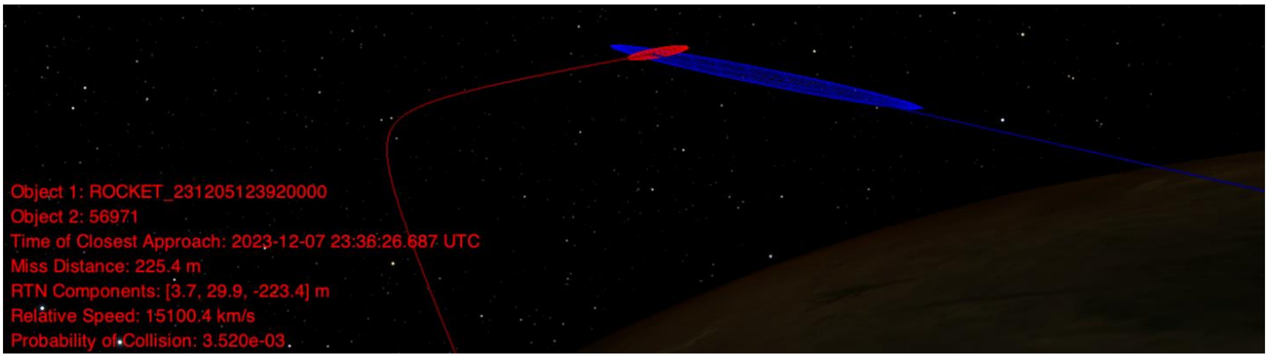
Note also that the PoC peaks at around 3.52e-3, a risk level usually considered unacceptable for operational missions, in a very close encounter between the rocket body and object 56971. Upon user request, the geometry of the riskiest encounter can also be visualised by ASSET, as shown in *Figure 8*.

VIII. PROTOTYPE LIMITATIONS AND FURTHER DEVELOPMENT

Since the early development of ASSET, it's been clear that the main challenge would have been keeping the computational effort reasonably low. Taking again the test scenario described in the previous section as example, the total computation time has been of 390 minutes, meaning circa 6 minutes per launch trajectory. Per se, the duration of the single screening is acceptable and could be easily inserted into the launch countdown, especially for those missions with single lift-off epochs or small launch windows, e.g., a LEO launch with fixed LTAN. However, it would not be suited to screen a very large launch window, spanning over 1 hour, especially if frequent re-screenings prior the lift-off are required, since in this case the screening time would exceed the remaining time to lift-off.

Another limitation of the current prototype of ASSET concerns the validity of the results: indeed, the risk assessment is performed (and it is valid) point by point at different lift-off epochs, but it is still subject of on-going research how this can be extended to the regions in between two screened points. In fact, due to the dynamics of the problem, especially the high relative speeds between objects, there is the possibility of missing a risky conjunction in between two screened trajectories completely, leading to a "false" safe region. A first attempt to solve this issue was done in estimating a minimum sampling time – which is a function of the orbital parameters and size of the covariance ellipsoid – that would give a certain degree of reliability in the risk assessment in between two screened points. The result is that this timestep is often 1 second or shorter, leading to a significant increase in the computational effort.

On the other hand, this limitation might not represent a problem in terms of real-life launch operations: in fact, launch missions do usually not need a continuous assessment of the launch window with seconds-accuracy since, in case of launch postponement, the new lift-off epoch is usually (tens of) minutes later. Also, in the case of lift-off epoch restrictions for specific mission requirements (e.g., LTAN), then most probably only few seconds around the optimal T0 have to be investigated, with no need for more extensive screening.



```
Object 1: ROCKET_231205123920000
Object 2: 56971
Time of Closest Approach: 2023-12-07 23:36:26.687 UTC
Miss Distance: 225.4 m
RTN Components: [3.7, 29.9, -223.4] m
Relative Speed: 15100.4 km/s
Probability of Collision: 3.520e-03
```

Figure 8 – ASSET plot for riskiest conjunction

Therefore, further developments will mainly focus on improving the computational time and validity of results, with the second being strongly related to the first. In particular, each computational module (*Figure 3*) will be optimized individually:

- The filtering module will be improved to limit the Type I and Type II errors, returning fewer secondary objects from the SP catalogue in a shorter time
- The covariance estimation module will be enhanced with a machine-learning algorithm, already under development, to improve the association of the CDM-derived covariance to the objects of the SP catalogue
- The screening module, inherited from the operational COLA Fortran code, will be optimized for the specific "some vs all" application in ASSET. In addition, the code will be restructured to make all modules accessible from Python
- A more comprehensive study on the validity of the results will be conducted to assess the best approach to provide a continuous/less sparse assessment of the launch window
- The errors in the P/Ls separation Delta-V will be considered in the propagation of the covariance matrix of the P/Ls, thus augmenting the reliability of the PoC calculation
- Finally, the user interface, now file-based, will be improved in cooperation with potential customers, by developing a GUI or web-interface for an easy and fast external access to the service.

Besides the on-going research and service improvements, the current ASSET prototype is already a powerful analysis tool to be used for pre-launch collision risk assessment. The first realistic scenarios analysed clearly show that the launch collision risk is quite considerable, especially for certain target orbits, making even more clear the need for such a service for present and future safe access to space.

IX. ACKNOWLEDGEMENTS

This research work has been supported by the German Federal Ministry of Defence through the technological research and development assignment "Responsive Space Capabilities".

X. REFERENCES

[1] ESA Space Debris Office, "ESA's Annual Space Environment Report", iss. 7.1, 12.09.2023

[2] B. Vidal, D.-A. Handschuh, C. Wang, "Collision Risk at Launch" Proceedings of the 5th IAASS Conference A Safer Space for Safer World, by Ouwehand, L. ISBN:978-92-9092-263-6. Noordwijk, Netherlands: European Space Agency, 2012, id.60

[3] 18th Space Control Squadron, Combined Space Operations Center (CSpOC), "Launch Conjunction Assessment Handbook", v.1.0, December 2018

[4] Beaver B. A., Hametz M. E., Ollivierre, J. C., Newman L. K., & Hejduk M. D. (March 2015). "Recommended Screening Practices for Launch Collision Avoidance". NASA/TM-2015-219270.

[5] Hejduk, M. D, Plakalovic D., Hametz M. E., Newman L. K., Ollivierre J. C., Beaver B.A., Thomson R.C. (2014/01/12). "Recommended Screening Practices for Launch Collision Avoidance". NASA/TM-2015-000000.

[6] Falco G., Toso F., Ratcliffe A., Sanchez-Ortiz N., "Launch Collision Avoidance Sensitivity Analysis", Proc. 8th European Conference on Space Debris (virtual), Darmstadt, Germany, 20–23 April 2021.

[7] Klinkrad, H., Alarcón, J., Sánchez, N. (2006). Operational Collision Avoidance with Regard to Catalog Objects. In: Space Debris. Springer Praxis Books. Springer, Berlin, Heidelberg . https://doi.org/10.1007/3-540-37674-7_8

[8] Alfano, S. A Numerical Implementation of Spherical Object Collision Probability. J of Astronaut Sci 53, 103–109 (2005). https://doi.org/10.1007/BF03546397

[9] Hackel, Stefan und Wasser, Yi und Hahn, Manuel und Meinel, Michael und Kahle, Ralph (2018) Flight Dynamics Microservices. 69th International Astronautical Congress, 1-5 Oct 2018, Bremen.

[10] Hoots, Crawford, Roehrich, An Analytic Method to Determine Future Close Approaches Between Satellites, Celestial Mechanics, Volume 33, Issue 2, pp.143-158 (1984)

[11] Rodriguez, Fadrique, Klinkrad, Collision Risk Assessment with a "Smart Sieve" Method, Joint ESA-NASA Space-Flight Safety Conference, ESA SP-486, 2002

[12] A. Zollo, G. Di Campli Bayard de Volo, M. Weigel, S. Aida, R. Kahle, J. F. San Juan Díaz, "Usage of historical Conjunction Data Messages to reverse-engineer orbit errors of Special Perturbation catalogued objects", 2nd International Orbital Debris Conference (IOC), Houston, Texas, 4-7 December 2023

[13] 18th & 19th Space Defence Squadrons, "Spaceflight Safety Handbook for Satellite Operators", version 1.7, April 2023.

[14] Consultative Committee for Space Data Systems (CCSDS), "CCSDS Recommended standard for Conjunction Data Messages", Recommended standard issue 1, June 2013.