

SSC20-I-07

An Automated Constellation Design & Mission Analysis Tool for Finding the Cheapest Mission Architecture

Martin Fugmann, Sabine Klinkner
 University of Stuttgart, Institute of Space Systems
 Pfaffenwaldring 29, 70569 Stuttgart, Germany; +49 711 685 69647
 fugmann@irs.uni-stuttgart.de

ABSTRACT

Identifying the optimal mission architecture for a space mission is critical for mission success, especially for large constellations. Here, optimizing the entire mission architecture for cost is necessary for the business case to work. This paper presents an automated system that combines constellation design and mission analysis functions in the context of a distributed engineering environment. It utilizes analytical methods, commercial simulation software and other specialized tools to identify multiple eligible constellations for the user-defined case, perform the associated mission analysis tasks, and provide input for additional tools like cost estimation software to eventually identify the optimal constellation. This allows assessing more options to fulfill the mission in less time, establishing the benefits of each constellation analyzed, and also allows non-expert users to quickly understand and evaluate consequences of design or requirement changes.

INTRODUCTION

All elements of a space mission architecture are closely linked, and changes in one element can have large impact on all other aspects. Thus, design trades need to be performed for the entire mission architecture rather than for each element sequentially, if an optimal architecture is to be developed. This is especially critical for satellite constellations, where for example small changes in payload requirements can lead to a drastic increase in the number of satellites necessary to fulfil the mission escalating cost and scope.

An automated system for constellation design is being developed for the Digital Concurrent Engineering Platform (DCEP) within the IRAS (Integrated Research Platform for Affordable Satellites) research program of the German Aerospace Center (DLR)¹. The DCEP enables engineers as well as automated tools to cooperate in a distributed engineering environment. The constellation design tool can be run within the DCEP and provides input data for the satellite design tool ESDC (Evolutionary System Design Converger)², a launchability analysis tool to find suitable launch vehicles, and appropriate cost estimation software, which are being integrated into the DCEP as well.

Within the DCEP, individual tools can be located on their own server accessible via SSH protocol. This means that partners can add tools they consider to be confidential to the DCEP, as the tool itself can stay on the partner's server, and the tool can only be accessed via the DCEP interface that is provided by the tool

owner. With the current setup, the main DCEP server is located at DLR, while IRS provides a server running the constellation design tool and ESDC (see Figure 1).

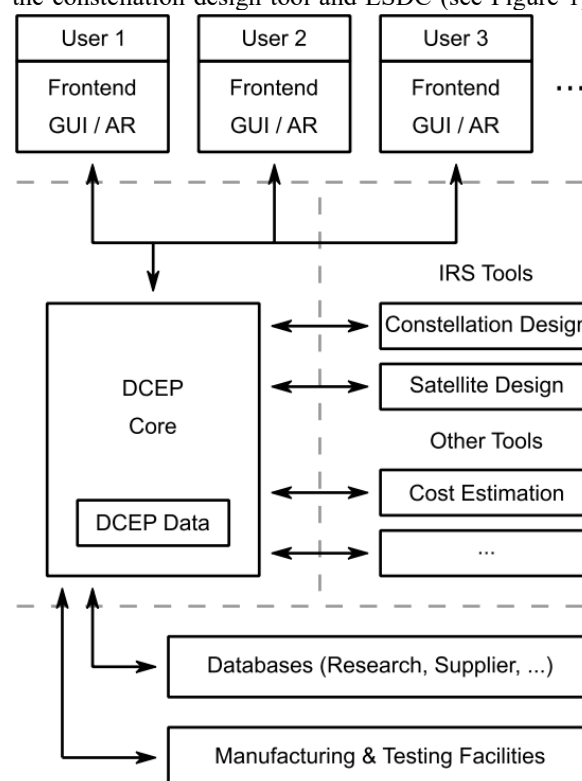


Figure 1: DCEP concept and architecture¹

The DCEP concept has two great benefits: First, it allows concurrent engineering in a similar way to concurrent engineering facilities without requiring physical presence. Second, the combination of different tools allows optimizing several key elements of the mission for minimum cost simultaneously. This aspect will be further explored in this paper.

The tool described is written in Python, but also uses several third-party tools (ESA-DRAMA³, ASTOS⁴). While the ESA-DRAMA tool offers a rudimentary Python API⁵ that was enhanced by additional functions, there is no direct interface for ASTOS. Calculations in ASTOS are done using a generic scenario which is automatically modified to the mission scenario under investigation.

Both input and output files are in XML format. XML is also used whenever possible to communicate with other tools. The input files are received from the DCEP central server, and results that can also include figures or other non-XML data are sent back to the DCEP central server as well. A summary of possible input and output values is given in Table 1 and Table 2 respectively. Not all input parameters are necessary. If multiple constellations were identified, the output can also be a list of possible constellations.

Table 1: Possible input parameters

Parameter	Comment
Constellation Type	Walker-Star, Walker-Delta, or both
No. of satellites	Not all of these parameters have to be defined. If enough parameters for a constellation are provided, the remaining ones will be calculated. If not, an iteration will be done using no. of orbital planes and / or no. of satellites per plane as iteration parameters
No. of orbital planes	
No. of satellites per plane	
Plane spacing	
Inclination	
Altitude	
Payload Field-of-View	
Payload min. ground elevation	
Max. Altitude	These define the iteration space if no altitude is defined
Min. Altitude	
Ground station location	
Mission start date& duration	
Max. residual orbital lifetime	Can be used to overwrite the 25-year lifetime limit of ECSS
Acceptable collision probability level	Can be used to overwrite the default criteria for collision avoidance maneuvers of 0.0001
Payload mass & power	Required to do preliminary satellite sizing
Drag coefficient	Default is 2.2
Reflectivity Coefficient	Default is 1.3

Parameter	Comment
Min. solar constant	Default is 1353 W/m ²
Cell efficiency	Begin-of-life solar cell efficiency
Degradation	Yearly solar cell degradation
Panel efficiency	Default is 71.53% (includes 85% area usage, 15% temperature loss, 1% cover loss)
Daytime power generation path efficiency	Default is 80%
Nighttime power generation path efficiency	Default is 60%
Satellite density	Default is 285 kg/m ³

Table 2: Output parameters

Parameter	Comment
Constellation Type	Walker-Star or Walker-Delta
No. of satellites	The parameters that were not defined in the input file are calculated and provided here
No. of orbital planes	
No. of satellites per plane	
Plane spacing	
Inclination	
Altitude	
Payload Field-of-View	
Payload min. ground elevation	
Satellite dimensions	Estimated acc. to SMAD
Subsystem powers & masses	
Orbital period	Calculated
Satellite avg. cross section	Calculated using CROC
Residual lifetime	Calculated using OSCAR
Annual collision probability	Calculated using ARES
Max. sun duration	Calculated or simulated using ASTOS
Solar panel area & max. power output	Estimated
High- and low thrust propulsion budget	Containing drag compensation, collision avoidance, deorbit, and margin
Ground station contact duration per day	Simulated using ASTOS
Max. time without ground station contact	
Plots	Filenames for generated plots, so DCEP central server can download them
Constellation scenario file	Zippered folder containing the ASTOS scenario representing the constellation

CONSTELLATION DESIGN PROCESS

The constellation design is based on the “streets of coverage” concept described in Wertz et al: Space Mission Engineering⁶. Currently it is limited to near-polar constellations with full coverage. This means that both Walker-Star and Walker-Delta constellations can be described using very few parameters: 5 for Walker-Delta, and 6 for Walker-Star. Between 2 and 4 parameters (depending on combination) are required to fully define a constellation, and the remaining parameters are calculated.

Constellation-defining parameters can be split into two groups: Constellation parameters (No. of satellites per plane, no. of orbital planes, spacing, street-of-coverage width), and satellite parameters (altitude, payload field-of-view, min. ground elevation). The two groups are linked by the earth-referenced payload field-of-view. This can always be calculated by applying spherical geometry relations from either the satellite or constellations parameter group as long as enough of them are known. For example, the satellite’s altitude, the payload’s field-of-view and the number of satellites per plane are sufficient to identify a Walker-Delta constellation. Minimum ground elevation can be calculated directly and allows to calculate the earth-referenced payload field-of-view, as shown in Figure 2. This in turn allows calculating the street-of-coverage width, as the number of satellites per plane is defined. Finally the plane spacing and thus the number of orbital planes can be determined to achieve the pattern shown in Figure 3.

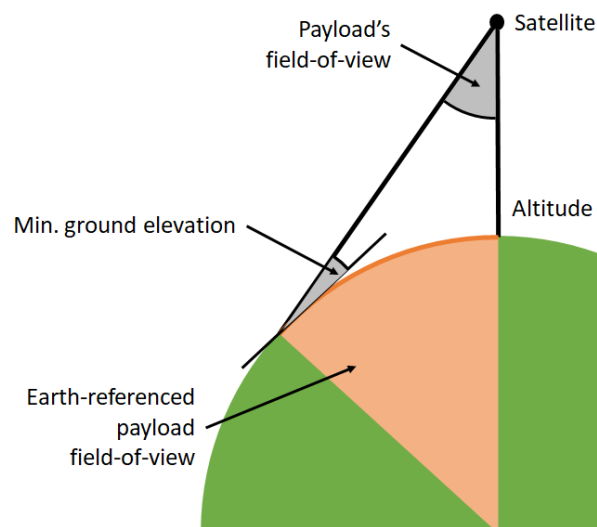


Figure 2: Geometry of satellite coverage

This approach also works without defining the number of satellites per plane: since the earth-referenced payload field-of-view can be determined first, the minimum number of satellites in a plane to achieve a

street-of-coverage can be calculated. The number of satellites per plane can then be gradually increased, and multiple valid constellations will be identified, which can then be evaluated further. In cases in which no direct calculation is possible, a least-squares method is used to identify the best solution (e.g. for distributing the orbital planes in a Walker-Star constellation).

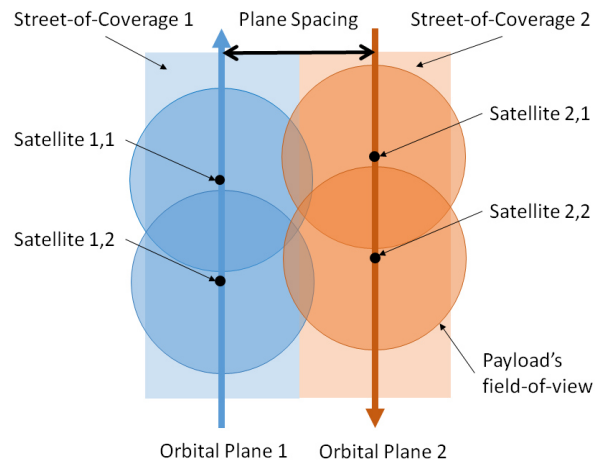


Figure 3: Geometry of Walker-Delta constellation coverage

Payload Scaling

Currently, the payload is considered “fixed”, which means its power and mass have to be defined first and are independent of other parameters, i.e. payload field-of-view. A better approach would be to let DCEP users define link budget requirements, e.g. the minimum isotropic signal level on the ground for a downlink application. After completing the link budget by calculating the losses (mostly free-space path loss and atmospheric dampening), the necessary RF output power can be determined, and payload mass and input power can be estimated based on a satellite hardware database. This approach is currently in development and an equivalent solution is planned for optical payloads.

MISSION ANALYSIS

For each suitable constellation identified by the process described above, an automated mission analysis is performed. The results of the mission analysis can be used by other tools like the ESDC for continuing the design process, and an ASTOS scenario file can also be generated to view the constellation and perform additional analyses.

First, preliminary mass and power budgets using handbook data⁶ are established based on the provided payload power and mass, and the satellite’s volume and thus outer dimensions are determined based on its mass

by using an average density of 285 kg/m^3 (by default)⁷. In the next step, the solar panel area is determined. To this end, the maximum eclipse duration is determined first. As typical LEO constellations are not sun-synchronous, the maximum possible eclipse duration will occur for all satellites in the constellation at some point. This is the case when the sun is located directly in the orbital plane. The maximum eclipse duration thus is only dependant on the eccentricity (which is very close to 0 in most cases) and the altitude, and is easily determined analytically. The solar panel area can then be calculated using typical or user-provided efficiency values for the solar cells, the panels, and the power processing unit.

Next, the propulsion budget is established considering station keeping, collision avoidance, and end-of-life disposal if necessary. First, the cross section of the satellite needs to be determined, as it is a required input for the following calculations. This is done using the CROC (Cross Section of Complex Bodies) tool which is part of the ESA-DRAMA toolkit. A generic scenario was set up that is then automatically modified with the previously determined satellite and solar panel dimensions. Examples can be seen in Figure 4. CROC then calculates the average cross section of the satellite. While the functionality was developed to simulate a randomly tumbling satellite, it is also used in this case as the effective cross section depends on the solar panel orientation, which is highly variable within the constellation. The randomly tumbling satellite calculation mode is thus considered a reasonable estimate. With the average cross section in place, the different propulsion demands can be determined.

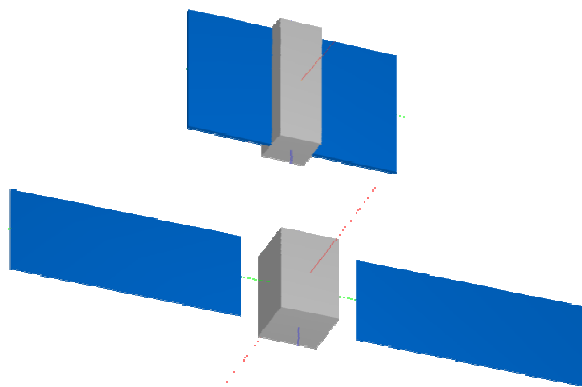


Figure 4: Generic satellite model used in CROC: 3U Cubesat (top), larger satellite (bottom). Satellites not to scale

For calculating the propulsion demand for station keeping, the atmospheric density along the flight path has to be known, which is highly dependent on the location, the time of day, and the solar cycle progression. Performing a full trajectory integration to

get the atmospheric density for one satellite might not be representative for the whole constellation, while doing so for all satellites in the constellation would take very long. For this reason, it was decided to calculate an average density that eliminates short-term and location-dependent variations. This is done by using a HEALPix grid (provided by the astropy-healpix python package⁸) to divide the atmosphere at the respective altitude into sections of equal area. The atmospheric density is determined for the central point of each section using the NRLMSISE-00 model, which is implemented by the msise00 python package⁹. This package automatically updates the predicted solar activity and geomagnetic activity indices with the latest predictions. Long-term variations and seasonal effects are considered, as the mission duration is split into segments of a specified length (e.g. one month), and the average density is calculated for each segment. A result of this calculation can be seen in Figure 5. The propulsion demand for station keeping can then be estimated by calculating the altitude loss per orbit⁶ for all segments, determining the Δv required to compensate for this altitude loss, and integrating over the whole mission duration.

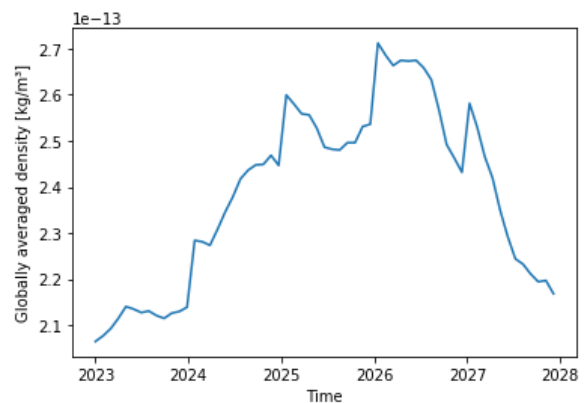


Figure 5: Globally averaged density in 400 km altitude over 5 years

Next, the propulsion demand for collision avoidance is calculated using the ARES (Assessment of Risk Event Statistics) tool of the ESA-DRAMA toolset. This program uses the MASTER (Meteoroid and Space Debris Terrestrial Environment Reference) database to estimate collision probability, avoidance maneuver frequency, and the propulsion demand per maneuver. ARES provides a python API package which requires orbital data & epoch, the Acceptable Collision Probability Level (ACPL), and the spacecraft radius. The ACPL has to be provided by DCEP users, otherwise a default value of $1:10,000$ is used. The spacecraft radius is calculated from the average cross section determined previously. With this input, the

annual collision probability, and the propulsion demand for collision avoidance, can be calculated.

The final item in the propulsion budget is a propulsive end-of-life deorbit in order to comply with the 25 year lifetime limit defined by UNOOSA¹⁰, or a lower user-defined limit. The tool OSCAR (Orbital SpaceCraft Active Removal) of the ESA-DRAMA toolset is used to calculate the residual orbital lifetime. It requires orbital data & epoch, the satellite's cross section, mass and drag coefficient, and the lifetime limit as input parameters. The tool then calculates how long it takes the satellite to deorbit naturally. It can also calculate the critical altitude the satellite has to reach to deorbit within the given limit, but this feature is currently not supported by the python API and not used in the constellation design tool.

If a propulsive deorbit is needed, the propulsion demand is calculated for both high-thrust and low-thrust propulsion systems. Both maneuvers are calculated so that the perigee reaches a specified altitude at which the satellite can be considered to be removed from orbit. For high-thrust systems, a single burn to lower the perigee is considered. For low-thrust systems, the continuous thrust equation is used.

To finalize the mission analysis, ground station contact times and maximum sun duration are calculated. The ground station contact is calculated using a customizable scenario in the simulation software ASTOS. The scenario is automatically modified to include the ground stations defined by the DCEP user, and the contact times are calculated using a visibility analysis.

For maximum sun duration calculation, conditions exist in which it can be calculated similar to the maximum eclipse duration described before. With the same assumptions as before, e.g. almost circular and non-sun-synchronous orbit, the maximum sun duration depends on the altitude and the angle between orbital plane and sun vector. It appears when the sun's elevation in the satellite-centred earth-referenced celestial sphere is maximized. The maximum angle is either the sum of the orbital inclination and the inclination of earth's rotational axis (23.5°), or 90° (which means the orbital plane is perpendicular to the sun vector), whichever is less. The maximum sun duration can then be calculated from the orbital period, which only depends on the altitude. However, this is not always applicable, as numerous situations exist in which the minimum eclipse duration is zero, which means that the maximum sun duration is longer than one orbital period. These cases cannot be covered by this analytical approach and are calculated using the ASTOS

simulation software, again using a customizable scenario and an eclipse analysis.

Launcher Selection

As launch is an important cost factor, it should also be considered in optimizing the constellation. Large constellations will most likely be launched by using the entirety of the vehicle's payload mass for as many constellation satellites as possible. The payload capacity, however, depends on altitude and inclination, and is often not publicly available for the constellation orbit as typical launcher manuals only contain payload masses for sun-synchronous orbits, and for a reference low-earth orbit. Deriving the actual payload capacity for a specific orbit often involves detailed simulations and optimization of the launcher's trajectory, which takes significant time.

To solve this problem, Astos solutions has developed a "Launchability Analysis Tool" (LAT) that uses a space launch system database as well as reference case simulations with their own simulation software ASTOS to provide quick results for suitable launchers and their payload capacity¹¹. While the tool was originally developed for use with a graphical user interface, a command-line controlled version is currently under development and will be included in the DCEP. This will allow identification of the cheapest launch vehicle for the constellation, also with several additional constraints like political restrictions. Launch of multiple satellites at once will be considered by adding a fraction of the satellites' mass to the payload mass requested, and by also considering the fairing space limits included in the database.

EXAMPLE RESULTS

In this section, a few results achievable with the current constellation design tool are presented. The figures were automatically generated using an XML interface that will also enable DCEP users to generate visualizations of their results the way they prefer.

Figure 6 shows all constellations identified for one scenario within a given altitude range, the payload's field-of-view was fixed. Figure 7 shows how the propulsion budget changes with respect to altitude. It can be seen that each component is dominant on different altitudes: in high orbits, active de-orbit requires the most propellant, while at low altitudes, the propulsion demand for station keeping rises. In between, the propulsion demand for collision avoidance is increased. Figure 8 shows the calculated residual lifetime, the calculations are automatically cut off at the given lifetime limit to save calculation time.

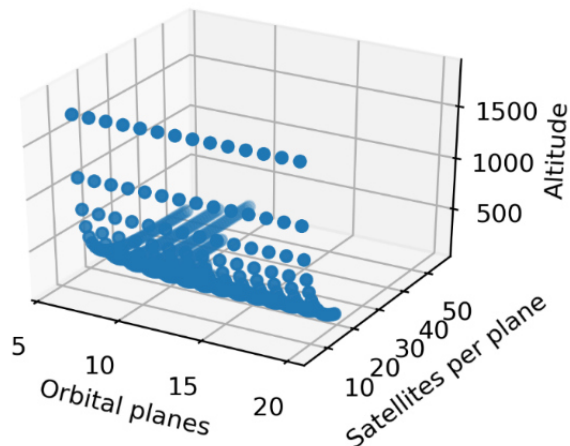


Figure 6: Example of identified constellations with respect to altitude

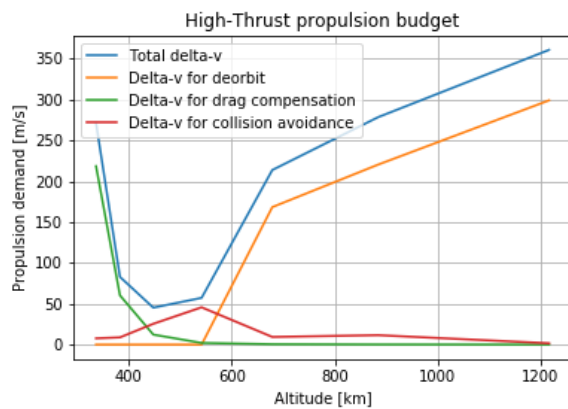


Figure 7: Example propulsion demands with respect to altitude

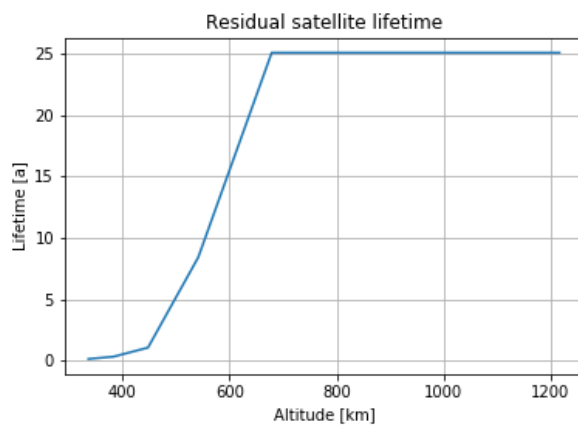


Figure 8: Calculated residual lifetime

Figure 9 and Figure 10 show more typical mission analysis results: Orbital period, maximum eclipse duration, average ground station contact per day and maximum time without contact.

Figure 11 shows that the reduction of sun duration in lower orbits is considered when establishing the satellite dimensions, as the solar panel size is increased.

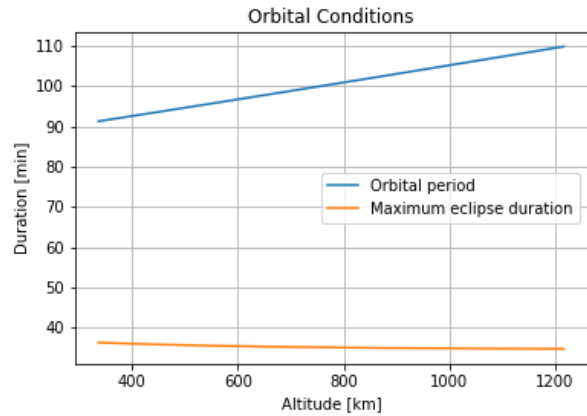


Figure 9: Orbital period & maximum eclipse duration calculated

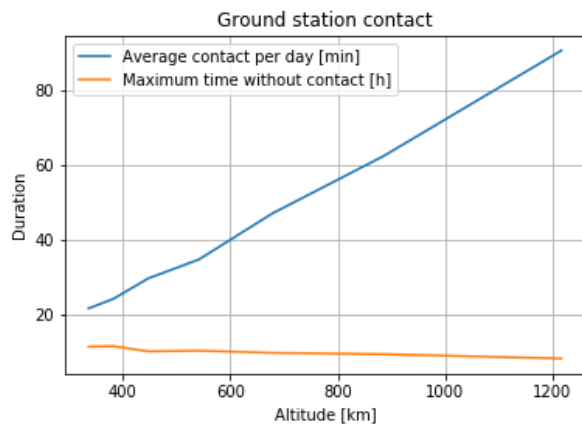


Figure 10: Simulated ground station contact times

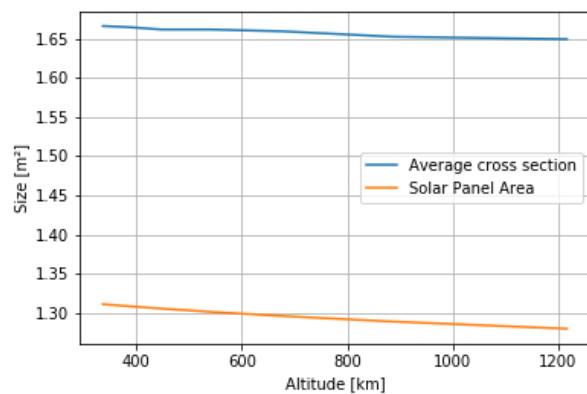


Figure 11: Calculated solar panel size & average cross section

Every constellation evaluated in the design tool can automatically be converted into an ASTOS scenario for further evaluation, as shown in Figure 12.

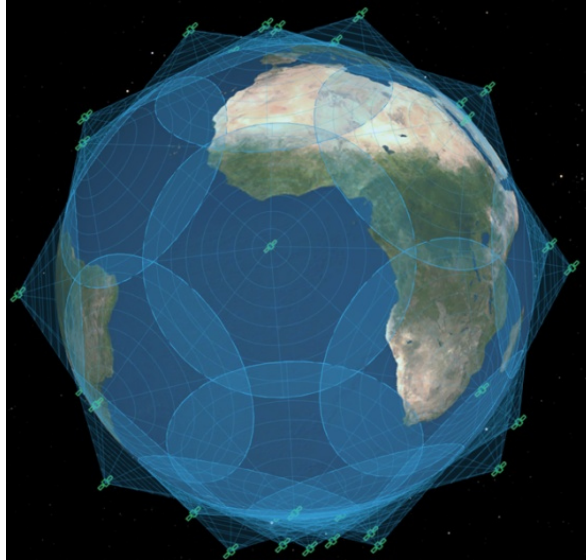


Figure 12: Example of automatically created scenario for use with ASTOS containing the constellation

CONCLUSION

Finding the optimal satellite constellation for a given mission can be a very tedious task. The constellation design tool presented within this paper automates the identification of potentially suitable constellations and several mission analysis tasks that enable comparing the constellations identified with respect to the requirements they impose on the satellite. The results can serve as input for additional design tools like the ESDC to automatically find optimized satellite designs for each constellation. Within the DCEP, cost estimation software will be used to evaluate the results and identify the most cost-efficient overall solution. Integration of the Astos LAT will allow including the launch vehicle in this process as well. A new payload scaling tool will allow doing additional design trades with respect to the payload power and mass for communications payloads. By analyzing the key mission elements orbit, satellite bus, satellite payload, ground stations, and launch vehicle together, this design tool allows identifying the optimal mission architecture faster and, within the DCEP, also remotely and automatically.

ACKNOWLEDGEMENTS

This work is financed by the state ministry of Baden-Württemberg for Wirtschaft, Arbeit und Wohnungsbau under the reference numbers 3-4332.62-DLR/49

(IRAS), 3-4332.62-DLR/56 (IRAS II), and 3-433.62-DLR/59 (IRAS III).

The IRAS project research partners are: DLR-BT (German Aerospace Center – Institute of structures and design), Fraunhofer IPA (Institute for Manufacturing Engineering and Automation), and Stuttgart University IRS (Institute of Space Systems). The project also includes industry partners, such as Astos Solutions GmbH.

REFERENCES

1. Stäbler, T., Echsel, M., Ehresmann, M., Fugmann, M., Galla, D., Gottschalk, N., Hümbert, S., Lengowski, M., Müller, I., Sakraker, I., Skalden, J., Springer, P., Stindl, T. and H. Voggenreiter, “Integrated Research Platform for Affordable Satellites”, Proceedings of the 70th International Aeronautical Congress, Washington D.C., October 2019.
2. Ehresmann, M., Fugmann, M., Skalden, J., Galla, D., Montag, C., Herdrich, G., Fasoulas, S. and S. Klinkner, “IRAS: Progress in Development of the Digital Concurrent Engineering Platform, Software Tools and Innovative Technologies”, Proceedings of the 70th International Astronautical Congress, Washington D.C., October 2019.
3. ESA, “DRAMA – Space Debris User Portal”, <https://sdup.esoc.esa.int/drama/>, 2020 (accessed June 5th, 2020).
4. Astos Solutions GmbH, “Analysis, Simulation and Trajectory Optimization Software for Space Applications”, <https://www.astos.de/products/astos>, 2020 (accessed June 5th, 2020).
5. ESA Space Debris Office, “Welcome to the documentation of DRAMA's python package”, https://sdup.esoc.esa.int/drama/python_package_docs/, 2018 (accessed June 5th, 2020).
6. Wertz, J.R., Everett, D.F. and J.J. Puschell, “Space Mission Engineering: The New SMAD”, Microcosm, Inc., Hawthorne, California, 2011.
7. Springmann, P.N. and O.L. de Weck, “Parametric Scaling Model for Nongeosynchronous Communications Satellites,” Journal of Spacecraft and Rockets, Vol. 3, No. 41, pp. 472-477, May-June 2004.
8. Deil, C., Robitaille, T. and D. Lang, “What is HEALPix?”, <https://astropy-healpix.readthedocs.io/>, May 18th, 2020 (accessed June 5th, 2020).

9. M. Hirsch, “msise00 – PyPI”, <https://pypi.org/project/msise00/>, April 2020 (accessed June 5th, 2020).
10. United Nations Office for Outer Space Affairs, “European Code of Conduct for Space Debris Mitigation”, Vienna, 2004.
11. Cremaschi, F., Volks, S. and M. Jürgens, “Turbo-Boost for Launchability Analysis Tool (LAT)”, Proceedings of the 5th International Conference on Astrodynamics Tools and Techniques, Noordwijk, Netherlands, May 2012.