Novel Constellation Design Tool: A Framework for Asymmetric Constellation Design

Kyle Casey, William Hunter
T2S Solutions
4960 Corporate Dr. NW Suite 125E Huntsville, AL; (703)-508-7490
KCasey@t2s-solutions.com

ABSTRACT

This paper addresses the problem of optimization of earth orbit space vehicle constellations balancing access over specified earth regions subject to constraints on the constituent members' orbital elements. Whereas symmetric constellations such as Walker Delta Parameter Constellations are often a convenient starting point for system trade studies, the desire to balance access over multiple earth latitude bands requires exploration of asymmetric constellations. This paper proposes a simple, yet effective method to rapidly test asymmetric constellation designs incorporating arbitrary constraints on the space vehicles' orbital elements. A user of this novel Constellation Design Tool (CDT) provides inputs to the code, including details about desired vehicle altitudes, range of inclinations, number of space vehicles per orbital plane, and number of total space vehicles within the constellation, as well as the desired set of latitude, longitude and altitude points to which space vehicle access is to be tested. The first stage of the CDT executes a Monte Carlo simulation using pseudorandom generations of constellation designs, providing the user with the Keplerian elements of the top-performing constellations subject to a user-defined figure of merit. Subsequently, the second stage of the CDT, drawing inspiration from a particle swarm optimization method, makes incremental changes to the orbital elements, testing the reported performance of the constellation against all other variations of the base constellation. After completion of a specified number of iterations, the top-performing constellation's orbital elements are loaded into STK with the replicated simulation environment to further analyze the constellation and provide performance data to the user.

INTRODUCTION

The design of a constellation of space vehicles (SVs) for a mission has many possible methods. Coverage over any specific latitude range is directly correlated to the inclination of the SVs within the constellation. Suppose the objective mission is to provide coverage for a specific area within a latitude band. In that case, the constellation design becomes trivial, simply requiring spacing of orbital paths and vehicles to provide the maximum coverage over the area at the desired altitude (in other words, a symmetric design). When the desired coverage area includes multiple areas over different latitude bands, the ability to select a single inclination that best provides coverage is no longer a viable option. To determine an optimal constellation design for the latter situation, a novel solution is required to analyze all the possibilities.

AGI Systems Tool Kit (STK) is the industry standard for orbital design and visualization. The STK software allows for the ability to design SV missions in orbit around a central body, whether Earth or other bodies. Space missions designed within STK allow for detailed coverage analysis, link margin analysis (LMA), SV access to other vehicles and stations, and more. Even with the functional analysis capabilities in STK, the software cannot rapidly test asymmetric constellations for optimized coverage over set regions, leading to lengthened design time when strictly working within the STK guided user interface (GUI). To avoid this lengthened design time within STK, alternative software can be utilized to speed up the time of the design process.

Traditional Constellation Design Method

When attempting to design a satellite constellation, the general thought is the equal spacing of orbital planes and satellites, along with a set inclination, provides both a good result and simple analysis. The most common design (in fact, the only way to design a constellation automatically within STK) is the Walker-Delta design. The design equally spaces the total number of SVs within the number of orbital planes specified by the user. This design has become very popular with space mission designers since it is relatively easy to provide a complete analysis of the characteristics of interest[1]. Another popular derivative of the Walker-Delta constellation is a constellation referred to as "streets of coverage" which consists of equally spaced satellites in orbit planes to provide continuous coverage along the ground track, or "street". Looking at some of the large, new LEO constellations to date, we see these patterns used exclusively: OneWeb uses a "streets of coverage" constellation pattern, Telesat utilizes two sets of

Walker-Delta patterns at different inclinations, and SpaceX has a similar hybrid pattern to Telesat, only with many more sets of Walker-Delta patterns and many more satellites[2].

The use of this symmetrical style of design isn't restricted to the cases listed above, the popularity can be seen among many designed SV constellations. Drawbacks to this design include the assumption that the constellation is being provided for coverage on a global scale, as well as the strict symmetric design. According to Wertz[1], the major problem for designing constellations is that although symmetric constellations provide easy analysis, an accurate constellation study is not symmetric.

Proposed Solution

To design a constellation to optimize coverage to specific areas on the globe while minimizing the required number of SVs, a symmetric Walker-Delta pattern does not provide adequate design freedom. To design a tool that optimizes the above specifications, consideration must be taken for the differing latitudes of possible areas of interest that the constellation might need to cover. To allow complete freedom of constellation optimization, an asymmetric design tool is required that allows for differing inclinations and unequal spacing of orbital planes. Removing any bias from the constellation design that would cause a trend toward a symmetric Walker-Delta design, the proposed tool utilizes MATLAB's pseudorandom input to develop constellation designs that fall within a user's specified inputs.

CONSTELLATION DESIGN TOOL

To find a constellation for the situation listed above, a novel tool to design and analyze asymmetric constellations was required. Utilizing MATLAB and the accompanying Aerospace Toolbox, an algorithm was developed to set up constellations whose access was computed to a set of ground point locations. Constellations were scored in 3 categories to determine how well they performed. Once the first MATLAB program was finished, a second MATLAB program performed localized searching on the parameters to see if the constellation design could be improved. Once the optimized Keplerian elements were returned, a MATLAB program loaded the constellation into STK, allowing further analysis and confirmation of the MATLAB custom algorithm calculated scores.

The specific framework laid out within this paper details the process of both designing and analyzing an asymmetric constellation with the following specifications: all SVs are at the same altitude in a circular orbit, orbits are determined from Keplerian elements, SVs have a simple conical sensor onboard that represents the entire field of view. See Figure 1 for a visual overview of the algorithm presented and the overarching framework.

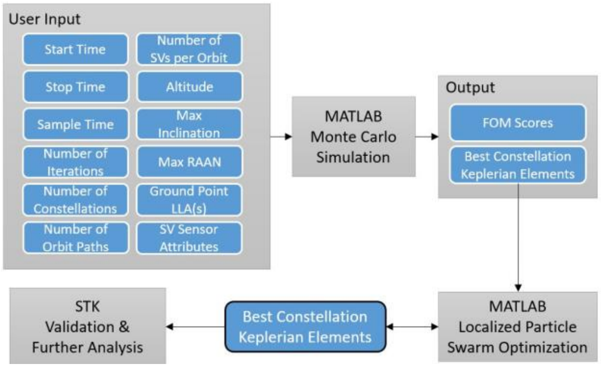

Figure 1: Constellation Design Tool Framework

Preliminary Requirements

To begin the search for an optimal constellation, initial user input must be provided to help constrain the problem which is being solved. The user must input the duration of the simulation, the desired sample time, the altitude of the SVs, the maximum inclination allowed, the number of orbital planes, and the number of vehicles per plane. In addition to the required orbital information, the user must also input the desired ground point locations to test access. The program can handle as many locations as desired, so it is up to the user to specify the desired level of granularity in the ground locations. Although the user is required to input these before running the first program, these are easily adjustable and can be set up to run a different set of requirements for each independent iteration of the program. By setting these initial inputs, the problem can be adequately constrained, and a solution set can be explored.

The user must input the desired number of iterations the program will run and the number of constellations per iteration. To properly explore the vast solution set of possible constellations, a higher number of constellations tested will help ensure a closer solution to the global maxima. Although more constellations provide a more complete solution set, the user must acknowledge the computational power required per simulation. A constellation with many SVs and ground points will take significantly longer to run than a small constellation computing to a single ground point. The user must allow for the processing time required to test their desired number of constellations. In the design of the specific tool specified in this paper, the Parallel Computing Toolbox from MATLAB was utilized to parallelize the simulation across all processor cores on a high-performance workstation. This design decision dramatically decreased the execution time of the custom MATLAB algorithm.

Monte Carlo Simulation

Once the user has appropriately constrained the problem, the algorithm then conducts a Monte Carlo run of the set parameters to rapidly explore the possibilities found within the problem's solution set. The algorithm created uses the user input of altitude, number of orbital planes, vehicles per orbit plane, max allowable inclination, and randomized right ascension of the ascending node (RAAN) to create randomized constellations representing local solutions within the solution set. Although the true anomalies of the spacecraft play a prominent role in the coverage of an orbital plane, the initial version of this algorithm uses equally spaced true anomalies for each orbital plane, set by dividing 360 by the number of vehicles in each plane. If desired, this variable can be randomized; however, computational time will significantly increase as the spacing of satellites in the orbital plane can become very uneven.

By running a simulation creating many constellations within MATLAB, all of which are tested for access to the user specified ground point locations, a comparison of constellations can be performed. The access provides the required quantitative data to perform calculations to determine which constellation design provides the best coverage to the ground locations. The constellation's access to ground points is stored in access arrays (1 for access, 0 for no access) for each vehicle to each ground station. These arrays allow a calculation to be completed for the coverage percentage, max gap time, and time-average gap for each constellation. Once these values are calculated, they are each scaled to a maximum value of 1 for the total simulation time. The formula to calculate the figure of merit (FOM) is the sum of each individual component: coverage percentage, max gap time, and time-average gap, where both gap values are reported in negative values to decrease the reported FOM.

After each iteration of the algorithm, the FOMs of all constellations are compared. If a FOM has the highest value calculated up to that point, the constellation’s data is saved into a .mat file, and the algorithm continues to iterate. The range of possible scores for any constellation is -2 to 1, with 1 always representing a constellation with access to all ground points and -2 representing a constellation with no access to the ground points during the simulation time. To display FOM values, the values calculated on the -2 to 1 scale are normalized so the values can be more easily compared from 0 to 1.

Localized Particle Swarm Optimization

With the approximate local optimal solution found using the Monte Carlo simulation, a localized particle swarm optimization, drawing inspiration from the Particle Swarm Optimization (PSO) method, is then utilized to help improve upon the presented optimized constellation. Since the driving factors of the constellation design are the individual plane's inclination and RAAN, the search randomly selects a variable, either the inclination or RAAN, in a randomly selected plane. Then, the selected variable is adjusted by an incremental step, either positive or negative, which again is randomly selected. The new constellation is then tested using the same procedure as above. Refer to Figure 2 for a visual outline of the process. To enable the algorithm to close in on an optimized constellation, it tracks the variable's rate of change as it is adjusted. It adjusts the variable's percent change lower as the rate of change decreases to ensure the optimal value does not get skipped over. The use of "particles" in this search allows the constellation to be adjusted with a new variable based on the number of particles set, which refers to the individual number of constellation variations being optimized. It is essential to understand that each particle is simply an isolated localized search of the constellation since the solution set is not inherently continuous, which is the case in a standard PSO[3].

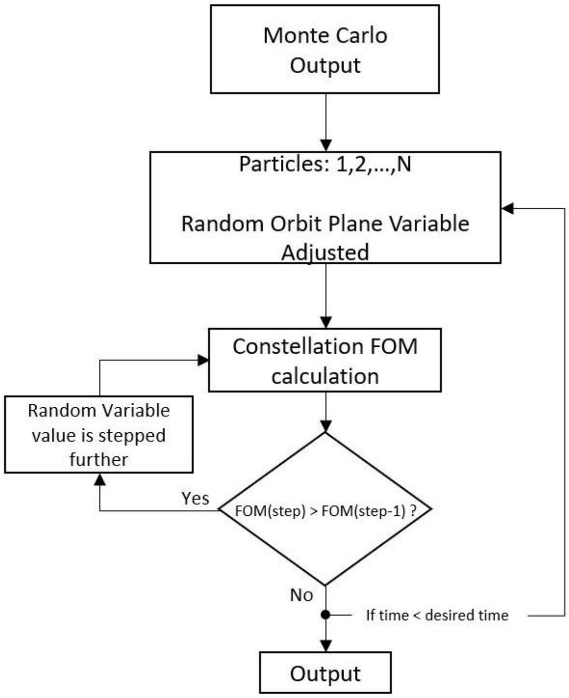
Monte Carlo Output
Particles: 1,2,…,N
Random Orbit Plane Variable Adjusted
Constellation FOM calculation
Random Variable value is stepped further
Yes
FOM(step) > FOM(step-1) ?
No
If time < desired time
Output

Figure 2: Localized Particle Swarm Optimization Outline

Since computational time in this algorithm can become very long as a variable closes in on an optimal value, the user must set both the time and count variable upfront. The time variable represents a "time step" within the search. At a time equal to 1, the constellation has each "particle” initialized with a specific variable changed. As each particle gets tested, if the score is improved, the count increases by 1, and the variable is adjusted again in the specific direction it was moving. A maximum count value can be set, and once all the particles hit either the max count or are no longer improving, the time iteration ends. Each updated constellation has another random variable changed, and the next time step iteration begins.

STK Simulation

Once the final optimized constellation is provided from the MATLAB algorithms, the corresponding Keplerian elements from each SV in the constellation are loaded into the STK environment using the STK Connect library within a MATLAB program. Using STK, the same ground locations are used to create a corresponding boundary area, providing an entire area to compute access to instead of specific ground points. The scripted STK environment provides for the confirmation of the computed MATLAB simulation FOM scores with higher fidelity modeling and analysis of the overall performance of the constellation.

SAMPLE PROBLEM RESULTS

The example study conducted within this paper looks at a network of satellites to best cover the 12 busiest space launch locations in the world. This scenario can be used to monitor rocket launches around the world, utilizing a small number of low Earth orbit (LEO) SVs. This constellation would allow public knowledge of when any country with a major space presence is attempting to launch into orbit. Using the algorithm discussed in section 2, the user input to the code was selected to create a problem in which there would not be enough SVs to provide full coverage, requiring an optimization to position the SVs to best cover the selected locations. The criteria for choosing the launch sites to be observed were the following: multiple launches per year (historically), scheduled launches for the 2022 calendar year, and must cover launches from all countries with major space programs. To frame the problem, the locations of each launch site considered are listed below in Table 1 with the name and respective geodetic latitude and longitude.

Table 1: List of All Launch Sites Used in Example Problem, Ground Points Provided in Geodetic Coordinates

Launch Site	Ground Points
Vandenberg AFB	**34.7483, -120.51817**
Kennedy Space Center/ Cape Canaveral	**28.45077, -80.52662**
Guiana Space Centre	**5.21238, -52.77388**
Plesetsk Cosmodrome	**62.98831, 40.96393**
Baikonur Cosmodrome	**45.90456, 63.32520**
Satish Dhawan Space Centre	**13.72396, 80.16396**
Tanegashima Space Center	**30.39855, 130.96905**
Wenchang Satellite Launch Center	**19.62864, 110.96013**
Xichang Satellite Launch Center	**28.24576, 102.02794**
Jiquan Satellite Launch Center	**40.96499, 100.28382**
Taiyuan Satellite Launch Center	**38.84880, 111.60152**
Rocket Lab Launch Complex	**-39.26081, 177.86576**

SVs per Constellation

For this example problem, the design for the constellation was bound between 20 and 30 SVs per constellation, all with a conical sensor with a field of view of 110 degrees representing the field-of-view of the observation payload. There were 5 separate designs considered:

- 4 orbital planes of 5 SVs
- 5 orbital planes of 4 SVs
- 5 orbital planes of 5 SVs
- 5 orbital planes of 6 SVs
- 6 orbital planes of 5 SVs

To quickly sample the design space of these constellation designs, the algorithm was set to test 6,000 randomized constellations per design, for a total of 30,000 constellations simulated. The goal of testing multiple designs across a range of SV numbers is to easily portray the expected differences between each design to provide evidence for which constellation design is selected for further optimization. The results can be seen below in Figure 3.

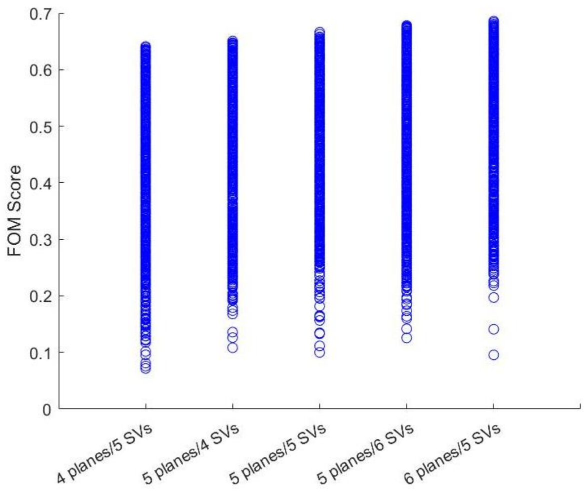

Figure 3: FOM Scores from Conducted Monte Carlo Simulation

The results of the Monte Carlo simulation show that each constellation configuration has maximum FOMs of 0.6410, 0.6506, 0.6663, 0.6780, and 0.6857 with respect to their order presented in Figure 3. These values, along with the scatter plot in Figure 3, show there are only slight differences in the maximum values found during the Monte Carlo simulation. Looking at the 6,000 sample constellations tested per configuration, there is a difference of only one one-hundredth of a FOM value between each configuration's top score, a negligible difference. Figure 3 also provides a visual for why the number of simulations conducted when randomizing constellation elements requires many constellations to be tested. The spread of FOM values for each constellation design varies greatly over the 6,000 constellations tested, which is why the user must make sure to provide a large enough sample of constellations to explore the vast solution set that is possible.

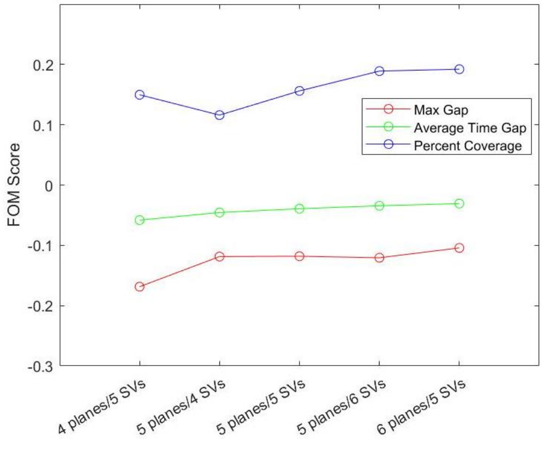

Figure 4: Max FOM Component Scores

In Figure 4 above, the best performing constellations, in terms of FOM, have their individual components broken down and plotted individually. We see that as the configurations progress to more SVs and/or more orbital planes, all components of the total FOM are trending towards a better score. An important conclusion from Figure 4 is that although the configurations with 30 SVs provide an increase in coverage, the relative change in both maximum gap time and average time gap does not show a significant increase over the configurations with fewer SVs. Due to this, a reasonable conclusion can be the larger constellations are not necessary as the smaller configurations are covering the locations at a similar rate as the larger configurations.

Further Optimization

After the completion of the Monte Carlo simulation run, the configuration with 5 planes and 4 SVs in each was run through the localized particle swarm optimization algorithm. The goal of this run was to improve upon the FOM score that was calculated for the constellation during the Monte Carlo run. Initially, the FOM score provided by the Monte Carlo simulation was approximately 0.65. Following the run of the orbital elements for this constellation through the localized PSO search, the score was improved to 0.70, an increase of almost 8 percent. To better understand the performance of the constellation and provide accurate results, the Keplerian elements of all satellites within the constellation were loaded into STK from MATLAB using the STK Connect library. A custom program recreated the simulation environment from the MATLAB simulation. The primary differences within the STK environment are the improved visualization and reporting abilities, as well as the creation of areas of interest for the specified launch locations. Instead of a singular ground point at each location in Table 1 for computing access, the STK simulation provides area target regions over each launch location with a ground point granularity of 0.5º. This allows a larger area to be included in calculations instead of only specified points. Testing SV access to a large area around the initial points computed by the MATLAB algorithm provides us with a much better understanding of the constellation's performance. The boundary area for each of the launch sites can be seen below in Figure 5 outlined in the red circles, with the SVs orbit path and conical sensor field-of-view in blue.

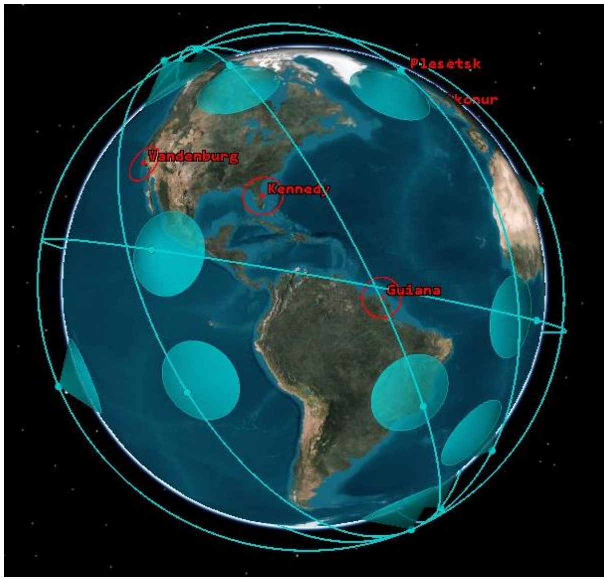

Figure 5: Visual of Constellation

Above in Figure 5, the constellation output from the algorithm as well as some of the launch site boundary areas are seen. The STK simulation was used to determine a quantitative measurement of the capability of the constellation to cover all the launch complexes to detect potential launches. In Table 2, the calculated response times for the constellation to each launch site over a 24-hour period is presented.

Table 2: Response Time to Each Launch Site

Region Name	Num Accesses	Min. (sec)	Max. (sec)	Average (sec)
Rocket Lab Launch Complex	53	1846	2525	2148
Guiana Space Centre	76	1288	1816	1545
Satish Dhawan Space Centre	67	1374	2065	1600
Wenchang Satellite Launch Center	61	1646	2598	1963
Xichang Satellite Launch Center	47	2107	2854	2436
Kennedy Space Center/ Cape Canaveral	48	2058	2642	2272
Tanegashima Space Center	45	2110	2558	2302
Vandenberg AFB	49	2038	2648	2302
Taiyuan Satellite Launch enter	53	1845	2712	2224
Jiquan Satellite Launch Center	57	1813	2411	2128
Baikonur Cosmodrome	65	1698	2439	2051
Plesetsk Cosmodrome	92	1140	1569	1321

The response time and access data provided in Table 2 shows the constellation's ability to cover the launch sites in the current configuration. With the maximum response time of any site being under 1 hour, the constellation would be able to monitor any attempted launch operations at any of the launch sites listed. Not only are the response times quick enough to pick up launch operations at the sites, but the number of times each site is accessed in a 24-hour period also provides reassurance that activity at any launch site will be monitored.

FUTURE WORK

At this paper's core is a simple, yet effective framework for conducting an asymmetric constellation design study which tends to be overlooked in favor of a simpler, symmetrical style constellation and mission design. The reasons for overlooking asymmetric constellations are numerous: there is a niche problem set that possibly stands to benefit from an asymmetric design, and within these design problems, asymmetric constellations pose a possible significant increase in mission cost and complexity if decided upon.

When considering the further development of this algorithm a few glaringly obvious points jump out. The first point to build upon is the time required to run the algorithm to provide a reasonable solution set. This algorithm would benefit from being re-written in a native programming language with consideration to optimization of the algorithm for modern computing hardware and software in comparison to the current computing environment of MATLAB and the toolboxes utilized within. Optimizing for modern computing hardware and software would allow the user to iterate on the design more rapidly, or to add higher fidelity parameters to conduct a more realistic simulation. Secondly, the current algorithm, although running well for simple cases of the same altitude and circular orbit constellations, would benefit from work to extend the capabilities to a wider subset of constraints. Widening the possible constraints on the current algorithm would allow the user to conduct studies of custom orbit constellations or constellations of SVs positioned at separate altitudes. Finally, it is possible that SVs of a constellation may have multiple objectives or multiple types of sensors on board. Extending the capabilities of the SVs themselves within the algorithm would allow for more complicated mission objectives to be considered.

REFERENCES

1. Wertz, J. R., "Mission geometry: orbit and constellation design and management: spacecraft orbit and attitude systems" El Segundo, CA: Microcosm Press, 2009.

2. P. Zong and S. Kohani, "Design of LEO Constellations with Inter-satellite Connects Based on the Performance Evaluation of the Three Constellations SpaceX, OneWeb and Telesat," Korean Journal of Remote Sensing, vol. 37, no. 1, pp. 23–40, Feb. 2021.

3. Bonyadi, M.R., & Michalewicz, Z. "Particle Swarm Optimization for Single Objective Continuous Space Problems: A Review." *Evolutionary Computation, 25*, 1-54, 2017.