[SSC20-WKVI-01]

Safe Construction in Space: Using Swarms of Small Satellites for In-Space Manufacturing

Rahul Rughani, David A. Barnhart
University of Southern California, Information Sciences Institute and Space Engineering Research Center
4676 Admiralty Way, Suite 1001, Marina del Rey, CA 90292; (310)-448-8503
rughani@usc.edu

ABSTRACT

With the emergence of a new space-to-space servicing sector, along with the return of manned missions beyond low earth orbit, there is an increased need for quick, efficient, and most of all, safe Rendezvous and Proximity Operations (RPO). An additional next big step forward may be true manufacturing in space, which could take advantage of swarms of small satellites cooperating in close proximity to each other, all subjected to the same laws of orbital mechanics. Currently, there is a lack of knowledge about how to safely operate a swarm of spacecraft in close quarters in a dynamically changing environment (i.e., a "space construction site"), without creating a high risk of collision and/or potential debris creation.

In order to formulate a stable, recurring, and efficient set of trajectories, a method was developed using genetic algorithms. This set of algorithms is able to solve for a set of relative motion trajectories for a swarm of N spacecraft, taking into account gravitational perturbations, to obtain trajectories that are recurring over a set amount of time. These algorithms also have the capability to dynamically alter the trajectories in order to take into account changes to the system, such as the addition of new spacecraft, or individual spacecraft failures.

INTRODUCTION

Given the recent advancements in satellite servicing technologies[1–7] and space robotics,[8] the collective capabilities of the space industry as a whole are moving towards in-space manufacturing. The ability to manufacture or assemble anything in space using robotics is a crucial technology for deep space exploration and the eventual colonization of Mars and beyond, as current platforms are limited to the volume and mass constraints of a single rocket fairing. Although the existing On-Orbit Servicing (OOS) methodology employs the use of large, monolithic robotic spacecraft, the method of OOS and in-space manufacturing proposed in this research study considers taking the leap from using one spacecraft to a swarm of dozens, potentially even hundreds of small robotic spacecraft. These *swarms* of spacecraft would operate analogous to a colony of bees, each individual member of the swarm performing its own dedicated task, culminating in the successful execution of a large and complex operation.[9,10]

To be able to control a large number of spacecraft cooperating in close proximity, each member of the swarm must be able to maneuver on its own, as well have the capability to sense the position and velocity of other nearby members of the swarm to communicate and avoid collisions. To streamline this process, for a given swarm function a set of optimized trajectories needs to be determined such that there are no predicted collisions between the spacecraft for a prescribed amount of time (e.g. at least a day), barring any malfunctions, where also the delta-v required to insert to the swarm is minimized.

This paper will showcase and describe the results of using a new genetic algorithm methodology to calculate safe trajectories, and how they can enable in-space manufacturing using multiple cooperating spacecraft. Additionally, investigations into how the scheme scales with the number of spacecraft involved will be discussed.

GENETIC ALGORITHMS

Genetic Algorithms (GAs) are a method of optimization, applicable to a wide variety of problems, that use a process similar to Darwinian evolution to *evolve* a set of random (or pseudo-random) initial conditions to find an acceptable solution, or even a globally optimal solution, to a problem.[11]

GAs can be applied to generate trajectories for a swarm of spacecraft, given a set of restrictions that the swarm will abide by.[12] These trajectories will be such that each member can perform their required individual actions, while minimizing the fuel required for maneuvering and also avoiding conjunctions, to a prescribed probability of collision, for a given amount of time. While near term it is expected that ground command uplink intervention would be needed to determine and execute remedial actions in the case of failure on orbit of a single element in the swarm, the long term goal is to develop autonomous algorithms to enable a swarm to accept and remediate failures of one or more of its members, in real time.

Computational Scheme

Computations for trajectory generation were performed in the local vertical local horizontal (LVLH) rendezvous coordinate frame, centered on a central spacecraft. In order to incorporate the J2 perturbations of the Earth's gravitational field, and reduce computation time, a two-stage solver was used, incorporating both the linearized Clohessy-Wiltshire (C-W) equations and the numerical integration of the equations of relative motion with perturbations.
This two-stage GA solver is comprised of the following:

1. *Stage One:* Use the GA solver to generate a set of trajectories satisfying the requirements of the swarm, using the linearized C-W equations to allow for fast iterations, albeit with a loss in accuracy from the linearization process.

2. *Stage Two:* Feed the solution from *Stage One* back into the GA solver, this time using the direct numerical integration of the equations of relative motion, with J2 perturbation added.

Kalman Filtering

In the *Stage Two* solver, a sensor-fusion Kalman Filter is employed to accurately compute the collision risk between each spacecraft, taking into account the errors in relative position knowledge between the spacecraft. In real-world operations, its impossible to know the position and velocity of a spacecraft with 100% precision. Thus, position and velocity are measured using onboard sensors, which have inherent error tolerances. These errors, from inputs such as GPS and relative Radar ranging, result in a covariance matrix attached to the state vector for each spacecraft in the swarm. These covariance matrices can be computed between each spacecraft in the swarm, meaning that if the covariance between spacecraft A and B is desired, sensor fusion between all other spacecraft and spacecraft B can be used to refine the state vector and covariance matrix of spacecraft B, minimizing the measurement error. By combining the measurements from multiple spacecraft, we can get more accurate measurements than by computing the covariances on each spacecraft separately.

A Kalman filter can be used to reduce the error in an

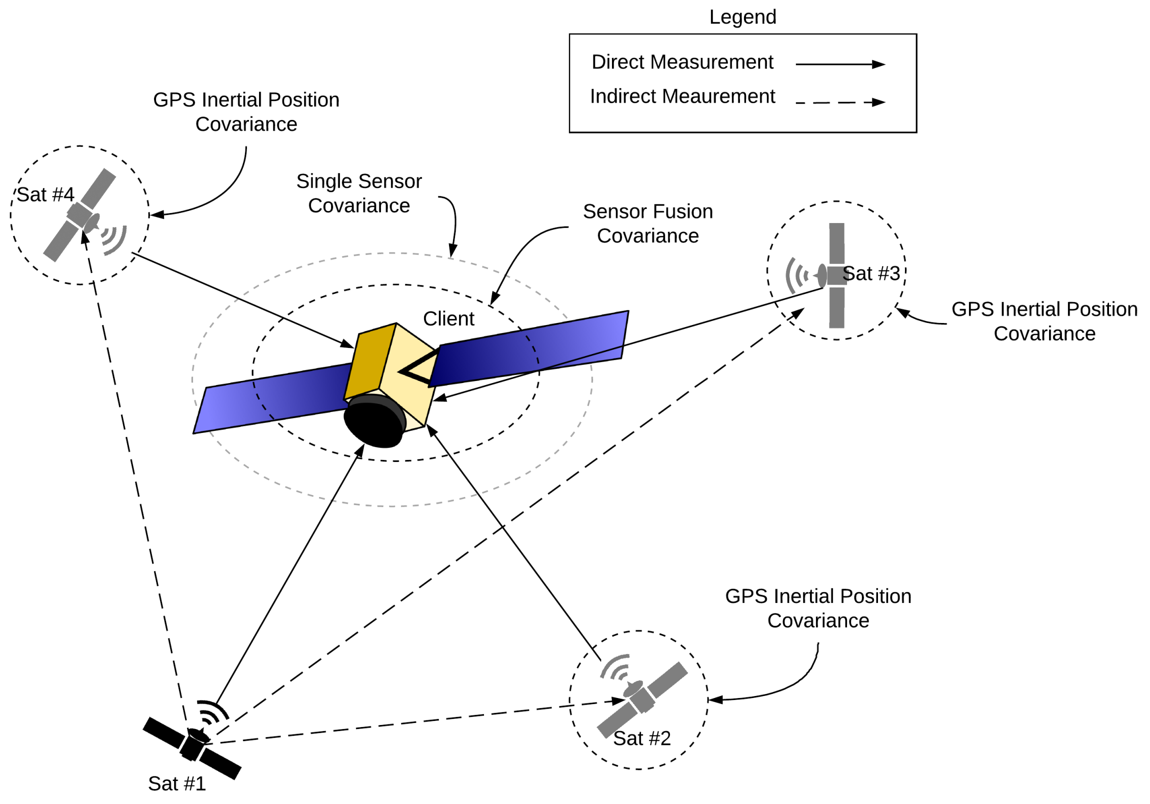

Figure 1: Sensor Fusion Diagram

estimated state by propagating a set of points through time, each corresponding to the boundaries of the covariance "bubble of uncertainty" that surrounds the spacecraft. As this is propagated, the covariance ellipsoid is refined by using measurements taken from a sensor at a known position, with a known precision. This is used successively over time to predict what states are more likely, and which are less likely, using a weighted scheme to determine within a 3-sigma gaussian distribution, where the spacecraft lies. The Kalman filtering method is useful for not only simulations, but for real-time operations, since the computational cost of the algorithm is very low, and can be run in real-time onboard a satellite.

In order to take into account the shared data of the swarm, which is the combination of the radar ranging sensors on each spacecraft, a sensor-fusion Kalman Filter is used. This is a modification of the standard Kalman filter described above, which uses multiple measurement update cycles to incorporate the shared data of the swarm to further refine the covariance ellipsoid for each spacecraft. Figure 1 above shows an example of the sensor fusion process, where the covariance of the position between *Sat #1* and the *Client* spacecraft can be improved by fusing the data from all the swarm spacecraft, even taking into account the GPS position errors defining the locations of each swarm spacecraft with respect to *Sat #1*.

TRAJECTORY GENERATION

One of the features that makes these GA-generated trajectories unique is that they are free flight trajectories that require no external inputs for their duration (within limits). This is because the major orbital perturbations are taken into account in the orbit propagation when generating the trajectories, such that if a time period of 10 days is specified, then the trajectory will not require any correction maneuvers for 10 days, and will return to the same starting position at that time, within a prescribed tolerance. Thus, a correction will only be needed after 10 days to repeat the same trajectory, using considerably less fuel than traditional station-keeping methods. However, these fuel savings come at a cost, as the distance and orientation to the target will not remain constant throughout the 10 day period, and thus is only useful if this is acceptable to the mission parameters. For most inspection and OOS tasks, this is allowable and thus allows fuel savings for smaller satellites that already have limited fuel reserves to begin with.

In an ideal case, there would be no limits on how far we can propagate these trajectories to ensure a collision-free solution, however this is in practice limited to 10-20 days, since gravitational perturbations of order higher than the J2 perturbations are not considered in this simulation. It would be possible to extend this limit be accounting for higher order orbital perturbations in future work.

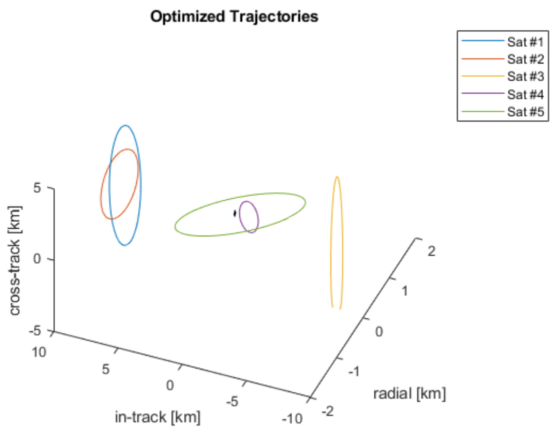

Figure 2: Swarm of 5 Spacecraft

Figure 2 shows an example of a set of 5 spacecraft in a swarm, centered around a central spacecraft. This figure shows the orbits propagated over a 24h period. In the figure, two spacecraft (Sats #4 & #5) are located within 4 km of the target spacecraft for close-up inspection and imaging, while another spacecraft (Sat #3) is located at 10 km to act as a communications relay. The final two spacecraft (Sats #1 & #2) are parked further out be used for future servicing operations after the inspection operations are complete.

Figure 3 shows a propagation of the same trajectories over a period of 10 days.

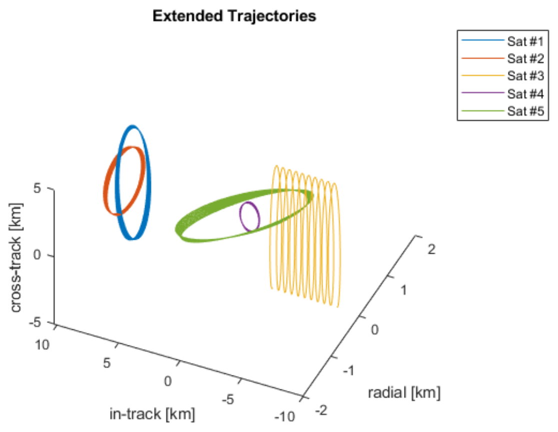

Figure 3: Extrapolated Trajectories over 10 days - Swarm of 5 Spacecraft

The goal of using a system like this to generate a set of trajectories for a swarm of spacecraft is to allow modular and customizeable designs for a swarm, to enable it to carry out whatever its goals may be. Thus, each spacecraft can have its own requirements and own sub-mission, yet they can all share data among themselves and their trajectories are setup to prevent collisions under free-flight with low delta-v.

IN-SPACE MANUFACTURING

Although to-date, the use of in-space manufacturing has been infrequent and limited to very large structures, like the International Space Station (ISS) assembled in orbit by astronauts and tele-operated robotics, the future of mankind's expansion into the cosmos will require the commonplace use of in-space manufacturing and assembly to exceed the dimensional constraints imposed by launch vehicle fairings. By assembling and constructing structures in-orbit, spacecraft can evolve to become more modular, with parts that are reusable and transferable, somewhat like using a set of standard LEGO® blocks in space.

Combining the method of swarm trajectory generation with the goal of in-space manufacturing/assembly, we can find new and unique ways to perform orbital assembly that cannot be done with a single, monolithic spacecraft. Using swarms of spacecraft which are operating autonomously in concert, an analog of a construction site on Earth can be setup, where there is a staging area for work on converting raw materials or prefabricated parts into sub-assemblies, which are then transported over to the site of the structure being aggregated.

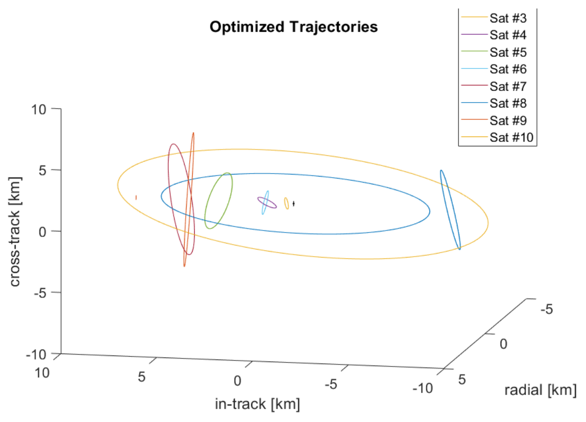

Figure 4: Swarm for In-Space Manufacturing

Figure 4 shows a swarm designed for in-space manufacturing, where there is a supply depot about 5 km ahead (in-track) of the aggregated object, while other spacecraft are approaching closer, within 0.5 km of the aggregated object.

Using GAs to generate trajectories for a swarm of spacecraft performing in-space manufacturing, we can ensure a safe "construction" orbital scenario, since one of the foundations of the swarm GA method is that all the trajectories are passively stable for a period of days (at least). Additionally, the swarm GA method allows dynamic re-configuration of the swarm to accept new spacecraft into the swarm, and to grow as the aggregated object changes in size and mass.[12]

SCALING WITH NUMBER OF SPACECRAFT

Although the algorithms in use for orbit maintenance and collision avoidance schemes can be run in real-time on the individual small satellites using Kalman filtering, the Genetic Algorithms that are used to generate the initial swarm trajectories, or to compute a new set of altered swarm trajectories, are not yet optimized to run in real-time aboard the spacecraft. The conjunction de-confliction process is the most time-consuming and it scales on the order of n-squared, where n is the number of spacecraft in the swarm. Future work will strive to reduce this computational burden, but in the meantime these computations must be done on the ground, and verified by ground support engineers, before being uploaded and implemented on the spacecraft.

The amount of time to compute a solution varies not only by the number of spacecraft in the swarm, but the pseudo-random initial conditions used to generate the GA population. In order to analyze variations in computational intensity of the problem with respect to the swarm size alone, a set of Monte-Carlo simulations were run for each swarm size. This allowed these variations introduced by the initial conditions of the solution to be averaged out.

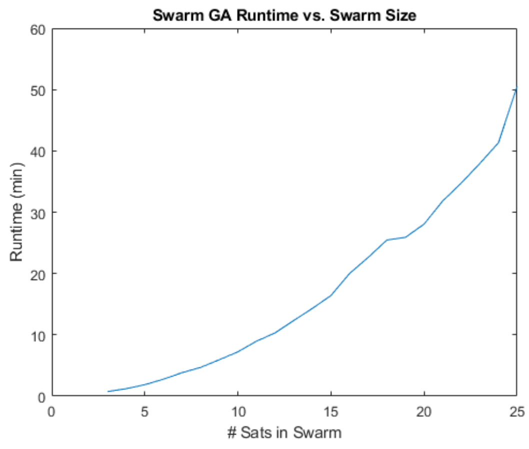

Figure 5: Runtime vs Swarm Size - Averaged over 100 runs

Although the computational time increases as a function of n-squared, future work will strive to reduce this by using machine learning techniques to avoid running similar types of problems over and over again, and instead store these typical solutions in a database for referencing. Additionally, the unique features of a ”swarm” of satellites each with their own processor may offer a real-time parallel processing distributed option for computing in real-time GA solutions.

CONCLUDING REMARKS

Using swarms of spacecraft assisted by Genetic Algorithm based trajectories, we can come up with safe and efficient trajectories for swarm rendezvous and proximity operations, thereby enabling in-space manufacturing and assembly on a large and distributed scale.

Simulations of these techniques have shown that it is possible to create such trajectories, as well as to maintain them using known and tested techniques to reduce real-time errors in flight such as Kalman filters for multi-sensor fusion.

While not possible at the moment, real-time generation of trajectories for N-number of satellites in a swarm may be possible with further optimization and machine learning in software, as well as taking advantage of the swarms natural number of multiple processors operating in parallel. The GA algorithm specifically envisions and enables multiple cooperative space objects to enable much larger scale in-space assembly and manufacturing of objects and platforms, safely and cooperatively. The GA algorithm trajectory may enable the possibility that a swarm could be launched to the orbit of Mars e.g., which would enable an autonomous swarm of spacecraft to ”build” a very large RF aperture in orbit thus allowing long distance communications portal for communication and relay before humans ever arrive.

REFERENCES

1. Caleb Henry. Northrop Grumman’s MEV-1 servicer docks with Intelsat satellite. *SpaceNews*, Feb 2020. `https://spacenews.com/northrop-grummans-mev-1-servicer-docks-with-intelsat-satellite/`.

2. Debra Werner. Orbital ATK’s giant leap into satellite servicing begins with baby steps. `https://spacenews.com/orbital-atks-giant-leap-into-satellite-servicing-begins-with-baby-steps/`, 2018. [Online; accessed 6-September-2018].

3. Neil Dipprey and Scott Rotenberger. Orbital Express Propellant Resupply Servicing. In *39th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, page 4898, 2003.

4. Aman Chandra, Himangshu Kalita, Roberto Furfaro, and Jekan Thangavelautham. End to End Satellite Servicing and Space Debris Management. *arXiv preprint arXiv:1901.11121*, 2019.

5. Benjamin B Reed, Robert C Smith, Bo J Naasz, Joseph F Pellegrino, and Charles E Bacon. The restore-L servicing mission. In *AIAA SPACE 2016*, page 5478. 2016.

6. Jason L Forshaw, Guglielmo S Aglietti, Nimal Navarathinam, Haval Kadhem, Thierry Salmon, Aurélien Pisseloup, Eric Joffre, Thomas Chabot, Ingo Retat, Robert Axthelm, et al. Removedebris: An in-orbit active debris removal demonstration mission. *Acta Astronautica*, 127:448–463, 2016.

7. Brook R Sullivan, Joseph Parrish, and Gordon Roesler. Upgrading In-service Spacecraft with On-orbit Attachable Capabilities. In *2018 AIAA SPACE and Astronautics Forum and Exposition*, page 5223, 2018.

8. J. Sasiadek. Space robotics - present and past challenges. *2014 19th International Conference on Methods and Models in Automation and Robotics, MMAR 2014*, pages 926–929, 11 2014.

9. David Saldana, Bruno Gabrich, Guanrui Li, Mark Yim, and Vijay Kumar. Modquad: The flying modular structure that self-assembles in midair. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 691–698. IEEE, 2018.

10. Neeraj Gandhi, David Saldana, Vijay Kumar, and Linh Thi Xuan Phan. Self-reconfiguration in response to faults in modular aerial systems. *IEEE Robotics and Automation Letters*, 5(2):2522–2529, 2020.

11. David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.

12. Rahul Rughani and David A Barnhart. Using Genetic Algorithms for Safe Swarm Trajectory Optimization. In *30th AIAA/AAS Space Flight Mechanics Meeting. Orlando Fl, USA*, 6-10 January 2020.