LLNL-CONF-497728

Maneuver Optimization through Simulated Annealing

W. de Vries

September 7, 2011

Advance Maui Optical and Space Surveillance Technologies
Conference
Wailea, Maui, HI, United States
September 13, 2011 through September 16, 2011

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

Maneuver Optimization through Simulated Annealing

Willem H. de Vries

Lawrence Livermore National Laboratory

ABSTRACT

We developed an efficient method for satellite maneuver optimization. It is based on a Monte Carlo (MC) approach in combination with Simulated Annealing. The former component enables us to consider all imaginable trajectories possible given the current satellite position and its available thrust, while the latter approach ensures that we reliably find the best global optimization solution. Furthermore, this optimization setup is eminently scalable. It runs efficiently on the current multi-core generation of desktop computers, but is equally at home on massively parallel high performance computers (HPC). The baseline method for desktops uses a modified two-body propagator that includes the lunar gravitational force, and corrects for nodal and apsidal precession. For the HPC environment, on the other hand, we can include all the necessary components for a full force-model propagation: higher gravitational moments, atmospheric drag, solar radiation pressure, etc.

A typical optimization scenario involves an initial orbit and a destination orbit / trajectory, a time period under consideration, and an available amount of thrust. After selecting a particular optimization (e.g., least amount of fuel, shortest maneuver), the program will determine when and in what direction to burn by what amount. Since we are considering all possible trajectories, we are not constrained to any particular transfer method (e.g., Hohmann transfers). Indeed, in some cases gravitational slingshots around the Earth turn out to be the best result. The paper will describe our approach in detail, its complement of optimizations for single- and multi-burn sequences, and some in-depth examples. In particular, we highlight an example where it is used to analyze a sequence of maneuvers after the fact, as well as showcase its utility as a planning and analysis tool for future maneuvers.

1. INTRODUCTION

The Lawrence Livermore National Laboratory (LLNL) is developing a comprehensive simulation environment for Space Situational Awareness (SSA) [1]. Many of its capabilities have been reported on at this and other conferences [2,3,4]. The Cosmos-Iridium satellite collision of February 2009 provided a good opportunity to perform full end-to-end testing and verification of all aspects of the simulation framework present at the time [5]. It became clear, however, that some capabilities were missing. In particular, we did not have adequate tools to do debris-cloud threat assessments against other orbiting objects, and how such a threatened target would have to maneuver to reduce the risk of a collision. We therefore started to develop code that could generate virtual debris-clouds based on just a few key parameters (e.g., mass, overlap, and relative velocity), all calibrated against the rather costly hydro-code simulations [6]. These debris-clouds are generated in their own coordinate frame, and can therefore easily be associated with any particular set of objects after a proper coordinate-frame transfer. In cases where a small piece of debris hits a larger satellite (i.e., an asymmetric collision whereby the impactor is completely destroyed), just a single cloud is generated. These single clouds, with appropriate scaling, can also be applied to the rather common "spontaneous" satellite break-up cases [7].

After we developed this virtual debris-cloud capability, it was a straightforward process to link it to the conjunction analysis module for risk assessments. The evolution of these clouds, as they were propagated forward in time, allowed us to generate observing schedules in the simulation that focused on the rapid characterization of the debris-pieces orbital elements. Initially, with many debris pieces close together, this is a non-trivial exercise as the actual case of the Cosmos-Iridium collision has shown.

A virtual cloud of debris particles is not so much different than a set of virtual trajectories of a maneuvering object. Both cases concern new trajectories that are different from the unperturbed orbit. If there are no constraints on the maneuver other than a certain upper limit to the delta-V (i.e., burn in any direction, with any speed up to the maximum) the virtual trajectory “cloud” is indistinguishable from a virtual debris cloud with a similarly small velocity range. Instead of calculating collision risks, we are now using the trajectory cloud for chain-of-custody purposes. After a particular object has been lost due to a maneuver, all possible locations to where it could have moved assuming a limit to the burn-budget (and time period over which the maneuver was executed) constitute a volume of space that is reachable to the maneuvering object. This “reachable volume” can then be searched effectively to regain chain-of-custody, employing the same methodology for debris cloud characterization. Another parallel between the two cases is that in the former it is debris posing a risk to a certain asset, while in the latter it is a maneuvering satellite that could potentially interfere with the asset.

In this paper we report on the maneuvering aspect. There are three levels of optimization that we have implemented. The first level is no optimization at all. In order to calculate the full reachable volume, we cannot make any assumptions on the burn directions. There are still physical constraints that can be put onto the problem, like a maximum delta-V and a specific time window (typically starting since the last good / unperturbed observation of the maneuvering object), but other than that, there is nothing to optimize. The second, medium level of optimization does place additional constraints on burn directions. It does so by having a particular target orbit. A typical scenario would be a LEO to GEO transfer. The medium optimization generates a trajectory volume that encompasses all possible transfers given a burn-budget and time period. The main difference with the first level option is that the former constitutes the full reachable volume, whereas the latter is a subset volume that just contains all the correct transfer trajectories. In other words, the second option generates a volume for which an intent is associated with the maneuver; the first option generates a volume without prior intent (or unknown intent). And finally, we have a full maneuver optimization. This requires both a known initial orbit and a known target orbit. The possible optimizations include both travel-time and burn-budget optimizations. More details on each stage, with appropriate examples are given in Section 3. We will start off describing the Monte Carlo approach in combination with Simulated Annealing in the next section.

2. OPTIMIZATION APPROACH

The most direct way to initiate a maneuver is to modify the state vector of the maneuvering object. An instantaneous impulse would be a single addition to the velocity component of the state vector in the appropriate direction, whereas a more continuous thrusting maneuver would continue to modify the state vector for the length of the burn. A force-model propagator can be used to determine the resulting trajectory, and depending on the level of required accuracy, this propagator can range from just two-body to one that incorporates many more forces. At the moment we have implemented two versions, a quick-and-fast modified two-body propagator that includes the effects of the Moon, and corrects for the nodal and apsidal regressions, and a fully featured force model approach that includes among other things the higher gravitational moments, atmospheric drag, and solar radiation pressure. The main difference between the two propagators, other than accuracy, is a factor of about 100 in execution speed.

The Monte Carlo aspect to the approach comes into play through the generation of many virtual thrust vectors, thrust magnitudes, and burn times. Typically, thrust directions are randomly, and uniformly, sampled in a 3D fashion. Their magnitudes can either be kept fixed, varied with a Gaussian σ around a mean value, or varied uniformly over a range. Burn-time constraints can be similarly constrained. Depending on the level of optimization, further action has to be taken. The first level (no optimization) just involves the propagation of the virtual trajectories. Checks against re-entry, or collision with the moon are made at this level. Then, at each time of interest, a simple position dump of the virtual particles defines the volume. Relative densities are directly related to the relative likelihood that the maneuvering object is in that location. The second and third level of optimization involves the removal of virtual trajectories that do not meet the additional target trajectory constraints.

The medium level of optimization, hereafter called the “Trajectory Volume” calculation, involves stepping through time and at each step generating a large Monte Carlo cloud of virtual trajectories. Only those trajectories that arrive at the target over the period of interest are kept. Since at each step each virtual trajectory is equally likely (i.e., there are no directional biases and the other burn constraints are constant), the set of resulting trajectories can be used to compile various interesting statistics (see next Section).

The full optimization goes beyond the first two levels in that it tries to determine the global minimum in the complex χ^2 space of the figure of merit. Since the number of free parameters is sufficiently large (2 thrust angles, a thrust magnitude, and a time of burn), combined with their particular distributions and the source and target orbital elements, the typical χ^2 surface has many local minima and shallow gradients. We therefore implemented a slowly cooling (with adjustable speeds) Simulated Annealing code [8, and references therein] that greatly reduces the danger of getting stuck in a false minimum (i.e., a wrong, incorrect optimization), albeit at a higher computational cost. With each cooling step, the allowable ranges of the free parameters are reduced (i.e., cooled); thrust vectors and allowable magnitudes are slowly settling on the overall best solution. Figure 1, left panel, shows an example of the cooling process. After a certain minimum temperature is reached, the assumption is made that this represents the optimum solution. It will always present a valid solution, but depending on the exact settings of the optimization and the number of MC trajectories and cooling rate, it is not necessarily the very best solution. The guarantee that the result is the overall best is very hard to give, and usually one settles on solutions that are almost certainly the best given the run-time limitations. Alternatively, since each annealing run is statistically independent, one can determine a sense of the robustness of the result based on the outcomes of a few short runs.

3. APPLICATIONS AND EXAMPLES

This section describes the three levels of the maneuver optimization module in more detail, and provides examples. In most cases we are using a single scenario. A MEO asset in a 33 degree inclined elliptical orbit tries to maneuver into its intended GEO slot, under a range of delta-V and time limitations. Figure 2, top panel shows a schematic of this maneuver.

3.1 Reachable Volume

A reachable volume spans all possible locations that a particular object can maneuver to given a certain burn-budget and time. Typically no constraints are placed on the burn-directions in order to encompass the full extent of the reachable space. As such we are not placing any “intent” or “purpose” on this maneuver, in contrast to the trajectory volume described in the next section. The prime purpose of this RV is for chain-of-custody applications. After an object has been lost due to a maneuver, its RV is in essence a 3D probability density distribution that can be used directly to optimize search-and-retrieval. Figure 1 illustrates this. The right two panels are 3D representations of the same RV at t=+30 minutes after the burn. The color-coding reflects the relative likelihood that the object can be found at that location (the integral over the whole volume is 1). By covering the high-likelihood regions first, a very efficient object recovery is guaranteed. It should be noted that a typical ground-based optical sensor only needs to cover the 2D projection of the volume on the plane of the sky (i.e., angles-only observations), thereby greatly increasing the recovery efficiency.

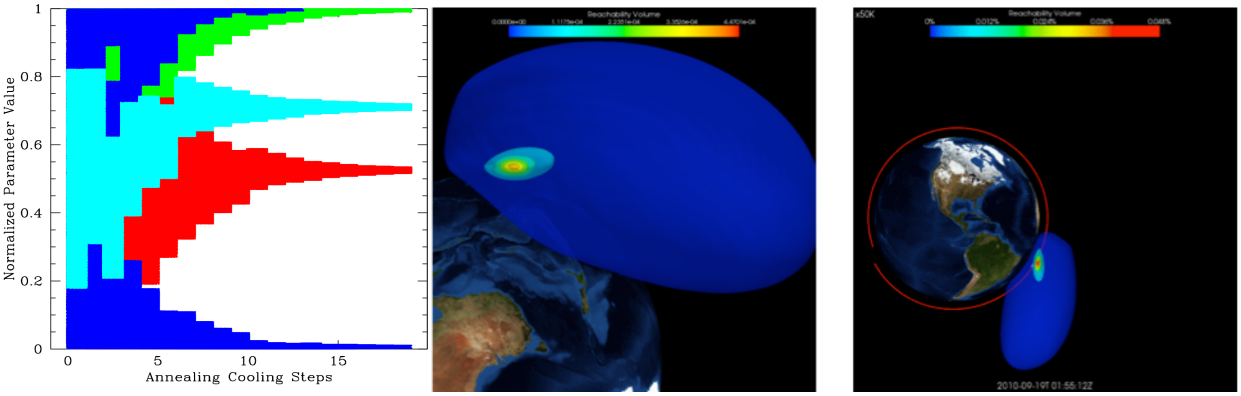


Figure 1. [Left Panel] Schematic illustration of the gradual reduction of allowable phase-space for the various free parameters under cooling in the annealing process. The range of the variables (burn magnitude, 2 burn direction angles, and burn-time) have been normalized. **[**Middle and Right Panel] Visualization of a reachable volume with a fixed t_burn and a uniform range of burn-magnitudes between 0 and 2 km/s, after 30 minutes. The color-coding reflect the relative likelihood that the maneuvering object can be found at that spatial location (running from low to high; blue to red respectively).

An additional application for the RV is assessing whether certain targets are within reach of the maneuvering object. This is an extension of the conjunction analysis module we reported on last year [4], in the sense that it will do a probabilistic conjunction against the RV instead of against another orbiting object. This is conceptually the same as running a conjunction analysis against a cloud of virtual debris particles.

The RV tends to grow rapidly with time depending on the allowed maximum delta-v. Especially in the LEO regime it does not take very long for the RV to wrap around the Earth (a few hours typically). Once this has happened, the usefulness of the RV diminishes rapidly as it starts filling more and more of the available space. More information is needed to reduce the growth-rate of the RV to keep it useful for longer periods of time. This could either be specific knowledge on thrust limits (i.e., the satellite is using an ion-thruster), or information about the intent of the maneuver (i.e., it is on its way to GEO). The latter case will be discussed next.

3.2 Trajectory Volume – Maneuver Statistics

Sometimes one has to contend with loss of custody for longer periods of time. Calculating an unconstrained RV for +8 hours for a LEO object produces a volume so large and complex that it provides little help in efforts to speed up the recovery of custody. If, on the other hand, we can make some assumptions on the intent of the maneuver, the size and shape of the RV can be reduced dramatically. A typical maneuver would raise a LEO / MEO object into its intended GEO slot. Figure 2, top panel, shows this in a schematic fashion. As explained before, only those trajectories that reach the intended target orbit within the time and burn-limit constraints are kept, all other trajectories (that would be present in an unconstrained RV) are discarded. This significantly extends the usefulness of the approach. Our example has an object maneuver from inclined MEO to GEO over a 3 day period, and at any given time during this period, the trajectory volume is small enough to be useful for target recovery purposes.

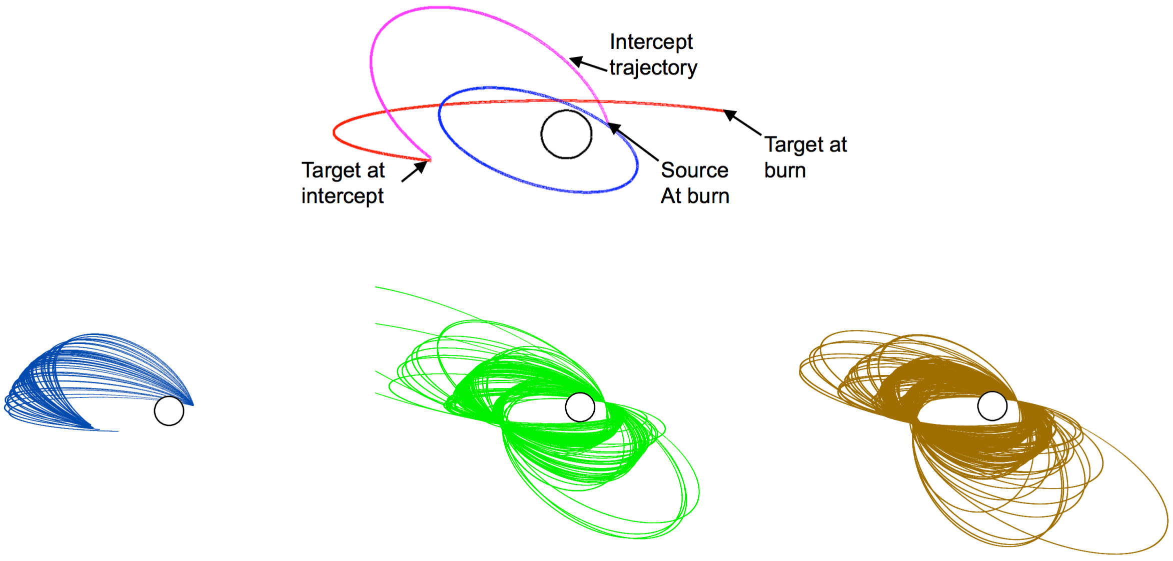


Figure 2. [Top Panel] A schematic representation of our example maneuver. The blue ellipse represents the source orbit, which is inclined at 33.5 degrees and has an eccentricity of 0.400. The red orbital segment represents the target orbit, with an inclination of 0.0 and an eccentricity of 0.0001. The open black circle represents the Earth to scale. [Bottom 3 Panels] A limited subset of allowable trajectories are shown, each with an initiating maneuver burn at t=0. The difference between the sets is the target arrival time. The set on the left arrives at the target within 24 hours, the second between 24 and 48 hours, and the right one between 48 and 72 hours. Note the distinct spatial bands / orbital regimes; these can be correlated to the structures seen in Figs 3 and 4.

The bottom part of Figure 2 illustrates the trajectory diversity depending on when the target orbit is reached. For orbits in the left panel, the target object is reached (for either a “rendezvous” or a flyby) within the first day, the middle panel has objects arrive between +1 and +2 days, and the right panel is between +2 and +3 days. To keep the figures clear we have only shown the maneuvers that were initiated immediately (i.e., at t=0; indicated by the “source at burn” arrow in the top plot). It is clear that there is a lot more structure to these data-sets than the unconstrained RV which tend to be smooth, nested surfaces of iso-probability, at least for the initial stages, see Fig 1). Indeed, since we are de-selecting trajectories that do not meet the added requirements after we have uniformly and randomly generated them[1], we can look at the statistical distributions to glean additional information.

A few representative distributions are discussed in this paper, but there are many more. The first is the diversity of available maneuver orbits as function of time. Given the source and target orbits, certain times are better for maneuvering than others. Orbital phases and limited burn-budgets may eliminate all but a few available options. This can be seen in Figure 3, left panel. It shows the number of distinct, valid trajectories of the 3 day period of interest. Some times are better than others by several orders of magnitude. It should be noted that the number of valid trajectories is just a function of the size of the simulation run and only important in the relative sense. After about 2.5 days no valid trajectories exist with delta-v < 3.5 km/s that arrive within +3 days. This arrival requirement is responsible for the slanted cut-off in the right hand plot, showing the rather curious pattern of travel-time versus maneuver initiation time. The color-coding reflects the magnitude of the initial burn (ranging from 0 to 3.5 km/s, red to blue respectively), adding a third dimension to the pattern. The trajectories plotted in Fig. 2 correspond to the various “families” of allowed trajectories at the t=0 point on the x-axis, going along the y-axis.

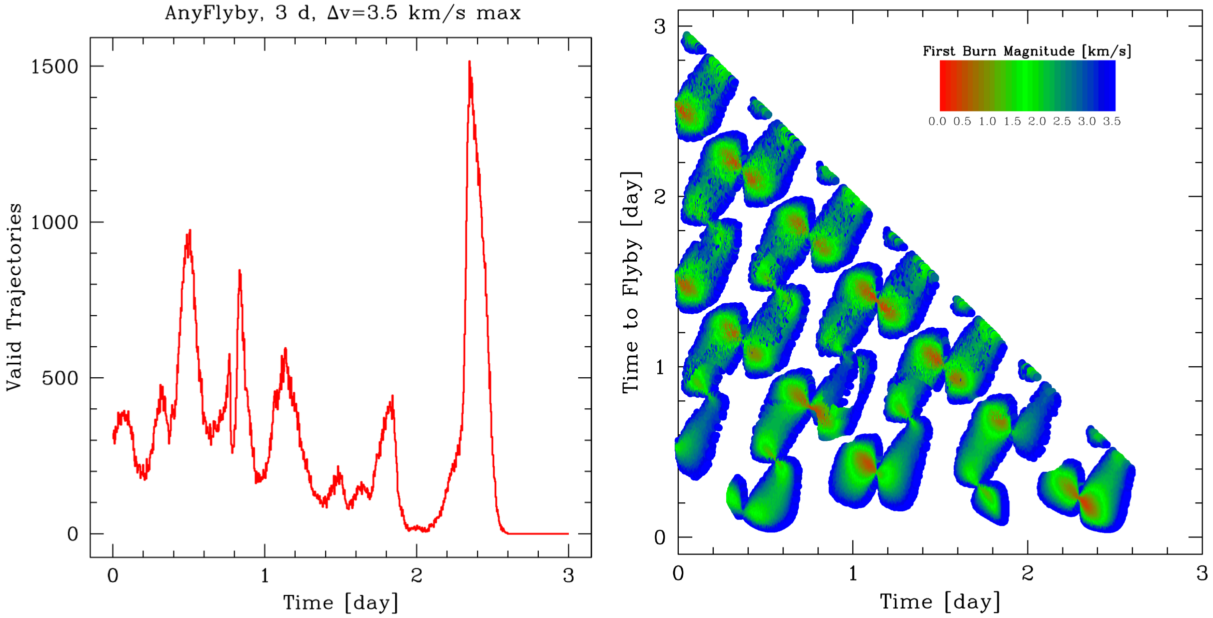


Figure 3. [Left Panel] A plot of the trajectory diversity as function of time. Certain times are open to a much larger part of phase-space. The constraints are listed at the top: a maximum delta-V of 3.5 km/s and a 3 day period in which the source should reach the target. Note that after 2.6 days no trajectories exist that satisfy these conditions. [Right Panel] Distribution of travel-time versus initiation of the maneuver (t_burn). The color-coding reflects the magnitude of the initial burn, ranging between 0 and 3.5 km/s. The slanted cut-off is due to the requirement that the flyby occurs within 3 days, for instance, after 2 days, only 1 day is left for travel.

[1] Again, for the unconstrained RV we use all these unbiased trajectories. The trajectory volume therefore represents a sub-set of the RV.

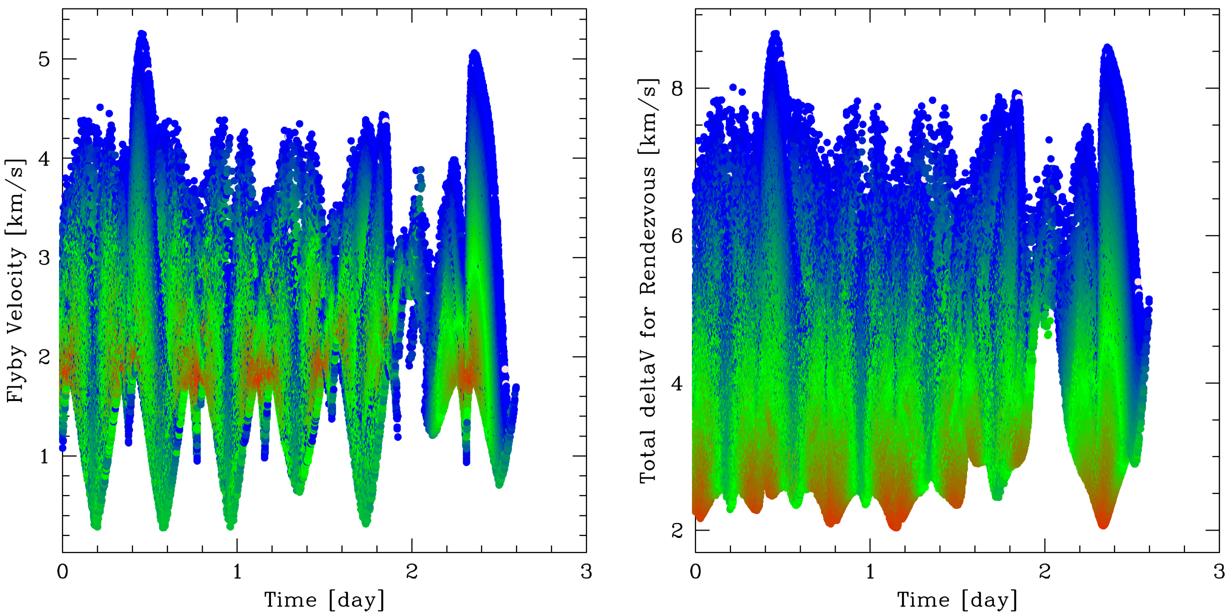


Figure 4. Corresponding distributions of the flyby velocity [Left Panel] and the overall delta-V for rendezvous [Right Panel] as function of time. The lower envelope of the right panel represents the lowest fuel expenditure for rendezvous at the corresponding maneuver execution time. The color-coding is identical to the one in Figure 3, right panel.

Another pair of time-varying quantities is plotted in Figure 4. On the left is the target flyby velocity as function of maneuver initiation time, and on the right is the required total delta-v (in km/s) for a successful rendezvous / docking with the target. Both plots are color-coded with the magnitude of the initial burn, analogously to Fig. 3, right panel. If one is interested in single-burn, very low delta-v flyby's (i.e., a near rendezvous), it is clear from the plot on the left that this happens only 4 times over the 3 day period and requires a ~2 km/s burn. The right-hand plot shows unambiguously, through its lower envelope, the times when the lowest delta-v rendezvous maneuvers (using two burns) should be executed.

The time evolution of these quantities with their detailed structure will provide a unique and immediate fingerprint of the maneuver solely based on the time and position of a successful reacquisition observation. Information on the remaining travel time, the relative flyby velocity, and whether a rendezvous is possible is immediately available.

3.3 Maneuver Optimization

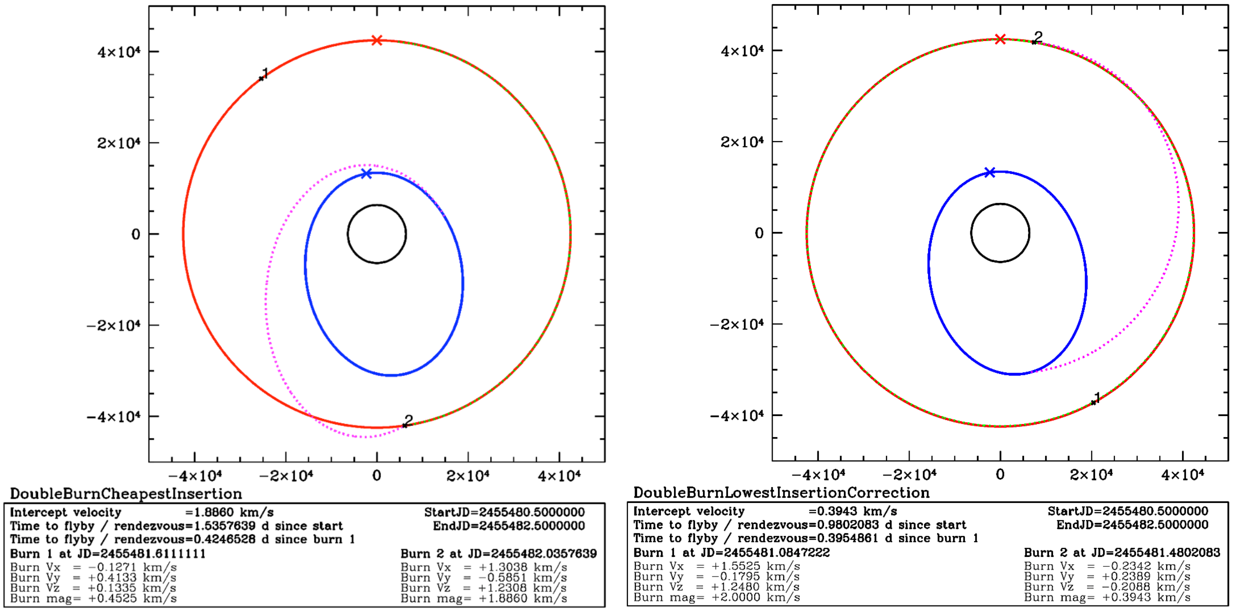
The final aspect of the maneuver analysis module is the detailed optimization capability. Unlike the previously discussed trajectory volume generation, this step really tries to find the best solution possible for a range of distinct optimizations. It is quite likely that the best solution for lowest total rendezvous fuel expenditure is found among the four lowest valleys in Fig 4, right panel, but the resolution of the medium stage is not enough to determine exactly which one. This is where the optimization step comes in. It is slowly biasing its distributions of the free parameters around these regions of interest[2], through the simulated annealing process.

Currently, ten distinct optimizations have been implemented. They are listed in Table 1, together with a short description.

[2] As such it does not provide usable statistics.

Table 1. Distinct Optimizations currently supported

Optimization	Description
1. SingleBurnQuickestFlybySinceNow	Shortest travel-time between now and flyby
2. SingleBurnQuickestFlybySinceBurn	Shortest maneuver / shortest time between burn and flyby
3. SingleBurnCheapestFlyby	Least amount of fuel spent for flyby
4. SingleBurnSlowestFlyby	Lowest delta-V at flyby
5. SingleBurnFastestFlyby	Largest delta-V at flyby
6. DoubleBurnQuickestInsertionSinceNow	Shortest overall time between now and rendezvous with target
7. DoubleBurnQuickestInsertionSinceBurn	Shortest maneuver time for rendezvous
8. DoubleBurnCheapestInsertion	Lowest amount of fuel (summed over two burns) for rendezvous
9. DoubleBurnLowestInsertionCorrection	Smallest possible second burn for rendezvous
10. DoubleBurnSlowestFlyby	Slowest flyby possible with two inadequate burns



DoubleBurnCheapestInsertion

```
Intercept velocity            =1.8860 km/s                    StartJD=2455480.5000000
Time to flyby / rendezvous=1.5357639 d since start              EndJD=2455482.5000000
Time to flyby / rendezvous=0.4246528 d since burn 1
Burn 1 at JD=2455481.6111111                          Burn 2 at JD=2455482.0357639
Burn Vx  = -0.1271 km/s                               Burn Vx  = +1.3038 km/s
Burn Vy  = +0.4133 km/s                               Burn Vy  = -0.5851 km/s
Burn Vz  = +0.1335 km/s                               Burn Vz  = +1.2308 km/s
Burn mag= +0.4525 km/s                                Burn mag= +1.8860 km/s
```

DoubleBurnLowestInsertionCorrection

```
Intercept velocity            =0.3943 km/s                    StartJD=2455480.5000000
Time to flyby / rendezvous=0.9802083 d since start              EndJD=2455482.5000000
Time to flyby / rendezvous=0.3954861 d since burn 1
Burn 1 at JD=2455481.0847222                          Burn 2 at JD=2455481.4802083
Burn Vx  = +1.5525 km/s                               Burn Vx  = -0.2342 km/s
Burn Vy  = -0.1795 km/s                               Burn Vy  = +0.2389 km/s
Burn Vz  = +1.2480 km/s                               Burn Vz  = -0.2088 km/s
Burn mag= +2.0000 km/s                                Burn mag= +0.3943 km/s
```

Figure 5. [Left Panel] Detailed maneuver optimization in which the overall fuel expenditure is kept at a minimum. The overall minimum burn sum is 2.3385 km/s for this maneuver. [Right Panel] The maneuver resulting in the lowest possible second burn correction (at 0.3943 km/s) for a total of 2.3943 km/s. The upper limit to the burns is 2.0 km/s, and the permissible time period is two days, otherwise the maneuver geometry is identical to Fig. 2, top panel. For both plots, the maneuver trajectory is shown in purple, the source and target orbits are blue and red, and the part after the successful rendezvous is color-coded in yellow. The numerals 1 and 2 mark the target position in its orbit after burn 1 and 2, respectively. The orbits are shown in the equatorial plane, so any elliptical trajectory at a non-zero inclination will appear distorted and more elliptical than it really is (cf. purple maneuver trajectory in left panel).

To illustrate the different outcomes for different optimizations, we show two cases in Figure 5. We are still using the source and target geometry shown in Fig. 2, top panel, but have limited the burn-budget to 2 km/s, and the period of interest to two days. It is clear that these two optimizations (cases 8 and 9 from Table 1) produce quite distinct trajectories.

4. SUMMARY

We have reported on our Maneuver Optimization module that is part of our HPC backed SSA testbed. It is based on an easily parallelizable Monte Carlo approach, in concert with a robust Simulated Annealing approach for the various optimization aspects. This framework not only allows for detailed trajectory optimizations, but it also can generate Reachable Volumes as well as Trajectory Volumes. The difference between the latter two volumes is the lack of intent for the Reachable Volume (i.e., the volume spans all possible locations that are within reach of the maneuvering object given its burn-budget and elapsed time), whereas the Trajectory Volume only contains those trajectories that reach a certain target orbit given a burn-budget and allowable time.

5. ACKNOWLEDGEMENTS

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

6. REFERENCES

1. Olivier, S. S., A Simulation and Modeling Framework for Space Situational Awareness, Proc. Adv. Maui Optical and Space Surveillance Technol. Conf., Wailea, Maui HI, 2008
2. Phillion, D.W., Pertica, A., et al., Large-Scale Simulation of a Process for Cataloguing Small Orbital Debris, Proc. Adv. Maui Optical and Space Surveillance Technol. Conf., Wailea, Maui HI, 2010
3. Henderson, J. R., Nikolaev, S., Phillion, D. W., de Vries, W. H., Pertica, A. J., and Olivier, S. S., Intelligent Sensor Tasking for Space Collision Mitigation, proceedings of the SPIE, Vol. 7691, 2010
4. De Vries, W.H., and Phillion, D. W., Monte Carlo Method for Collision Probability Calculations using 3D Satellite Models, Proc. Adv. Maui Optical and Space Surveillance Technol. Conf., Wailea, Maui HI, 2010
5. Olivier, S. S., Cook, K., Fasenfest, B., et al., High Performance Computer Modeling of the Cosmos-Iridium Collision, Proc. Adv. Maui Optical and Space Surveillance Technol. Conf., Wailea, Maui HI, 2009
6. Springer, H, Miller, W., Levatin, J, Pertica, A, and Olivier, S., Satellite Collision Modeling with Physics-Based Hydrocodes: Debris Generation Predictions of the Iridium-Cosmos Collision Event and Other Impact Events, Proc. Adv. Maui Optical and Space Surveillance Technol. Conf., Wailea, Maui HI, 2010
7. Johnson, N. L., et al., History of On-Orbit Satellite Fragmentations, 14th edition, Orbital Debris Program Office, 2008, NASA/TM-2008-214779
8. Press, W.H., Teukolsky, S. A., et al., Numerical Recipes, Cambridge University Press, 3rd edition, 2007, p. 549