66th International Astronautical Congress, Jerusalem, Israel. Copyright ©2015 by Eleven International Publishing. All rights reserved. One or more authors of this work are employees of the government of the United States of America, which may preclude the work from being subject to copyright in the United States, in which event no copyright is asserted in that country.

IAC-15-A6.6.8

LIGHTFORCE PHOTON-PRESSURE COLLISION AVOIDANCE: EFFICIENCY ANALYSIS IN THE CURRENT DEBRIS ENVIRONMENT AND LONG-TERM SIMULATION PERSPECTIVE

Fan Yang Yang
MEI / NASA Ames Research Center, United States
Bron Nelson
Computer Sciences Corporation / NASA Ames Research Center, United States
Roberto Carlino
STC / NASA Ames Research Center, United States
Andres Dono Perez
MEI / NASA Ames Research Center, United States
Nicolas Faber
SGT / NASA Ames Research Center, United States
Chris Henze
NASA Ames Research Center, United States
Arif Göktuğ Karacalıoğlu
STC / NASA Ames Research Center, United States
Conor O'Toole
University College Dublin / NASA Ames Research Center
Jason Swenson
LMCO Space OPNS / NASA Ames Research Center
Jan Stupl
SGT / NASA Ames Research Center, United States, jan.stupl@nasa.gov

This work provides an efficiency analysis of the LightForce space debris collision avoidance scheme in the current debris environment and describes a simulation approach to assess its impact on the long-term evolution of the space debris environment. LightForce aims to provide just-in-time collision avoidance by utilizing photon pressure from ground-based industrial lasers. These ground stations impart minimal accelerations to increase the miss distance for a predicted conjunction between two objects. In the first part of this paper we will present research that investigates the short-term effect of a few systems consisting of 10kW class lasers directed by 1.5 m diameter telescopes using adaptive optics. The results found such a network of ground stations to mitigate more than 85 percent of conjunctions and could lower the expected number of collisions in Low Earth Orbit (LEO) by an order of magnitude. While these are impressive numbers that indicate LightForce's utility in the short-term, the remaining 15 percent of possible collisions contain (among others) conjunctions between two massive objects that would add large amount of debris if they collide. Still, conjunctions between massive objects and smaller objects can be mitigated. Hence we choose to expand the capabilities of the simulation software to investigate the overall effect of a network of LightForce stations on the long-term debris evolution. In the second part of this paper, we will present the planed simulation approach for that effort.

For the efficiency analysis of collision avoidance in the current debris environment, we utilize a simulation approach that uses the entire Two Line Element (TLE) catalogue in LEO for a given day as initial input. These objects are propagated for one year and an all-on-all conjunction analysis is performed. For conjunctions that exceed a range threshold, we calculate the probability of collision and record those values. To assess efficiency, we compare a baseline (without collision avoidance) conjunction analysis with an analysis where LightForce is active. Using that approach, we take into account that collision avoidance maneuvers could have effects on third objects. Performing all-on-all conjunction analyses for extended period of time requires significant computer resources; hence we implemented this simulation utilizing a highly parallel approach on the NASA Pleiades supercomputer.

0. INTRODUCTION

This paper presents the current status of the research on the LightForce space debris collision avoidance concept. The paper is divided into two parts. The first part presents results on the short term utility of the concept in the current debris environment. The second part gives an overview on the current implementation of a long-term simulation approach.

Space debris is a growing problem for active spacecraft. A 2010 study calculated cost increases for several representative model satellite constellations due to debris impact in today's environment [1]. For the specific examples, replacement of satellites that are damaged or destroyed through collisions with debris would raise the cost of operating and maintaining by 4 to 14 percent, translating to hundreds of millions of dollars over the assumed operational period of 20 years. Long-term projections predict an increase in the debris population, leading to a further increase of risk and cost [2]. The increasing numbers are the result of new launches, spontaneous explosions, and fragmentations through collisions between space objects. The latter leads to a cascading effect that was originally predicted by Kessler [3] and was also confirmed in various other studies [4]. This Kessler cascade would increase the number of debris objects in orbit and increase the collision risk further, even if no more launches occur [2]. Hence, space debris already has both short-term and long-term implications.

Various methods have been proposed to improve this situation, or at least stabilize the number of debris objects in orbit. In order to classify these methods, the first distinction is between debris mitigation and remediation. In the space debris community, the term mitigation includes activities that fall under more responsible behavior, e.g. preventing accidental explosions and de-orbiting spacecraft at the end of their mission. Activities grouped under the term remediation deal with actively engaging existing debris objects. One class of remediation methods focuses on the active removal of a number of heavy debris objects from orbit [5]. As they are potential sources of additional debris, their active removal can help stabilize the number of debris objects, as shown in simulations. The second class of remediation activities are active, externally induced collision avoidance measures, where debris trajectories are influenced just in time to avoid collisions [6]. Just-in-time collision avoidance has the potential to be both of short-term and long-term use: on the one hand, such a capability can be used to protect active spacecraft; on the other hand, each avoided collision reduces the number of additional debris created.

LightForce is a method for just-in-time collision avoidance. The concept envisions reducing the risk of collisions by slightly changing the orbits of objects that are predicted to have a conjunction. Slight orbital perturbations are induced by photon pressure from ground-based, industrial strength lasers (fig. 1).

LightForce aims to reduce the risk of collisions by targeting conjunctions on warning. By tackling high risk conjunctions it addresses potential collisions directly. Taking part in stabilizing the debris environment (by preventing additional collision debris) is a second benefit. LightForce would use tracking data and orbit prediction to continuously compile and update a list of high risk conjunctions to engage. As illustrated in Fig.1, photon pressure from ground-based lasers would be used to alter the in-track velocity of space objects. Over time, that translates to an in-track displacement. In an operational setting, LightForce would engage objects involved in a conjunction and simultaneously (and continuously) update orbital data.

The first part of this paper summarizes the status of our research on LightForce's as a tool for just-in-time collision avoidance in today's environment. We provide an outline of the goals of the conducted simulations, the simulation approach, its implementation and a summary of results.

The second part of this paper describes our ongoing efforts to implement an open-scenario, long-term space debris simulation approach. The state-of-the-art simulation tool to assess the debris environment over decades is NASA LEGEND [7, 8]. Within a Monte Carlo framework, LEGEND propagates every object over a 10cm size cutoff, using large time steps of several days (the default is 5 days) [8]. In between these time steps, a statistical evaluation returns the probability of a collision event which then leads to a random decision whether to insert new fragments into the model. Utilizing multiple Monte Carlo runs, this approach provides an average projection of the future debris environment.

Investigating active debris removal (ADR) in this framework is comparably easier than for collision

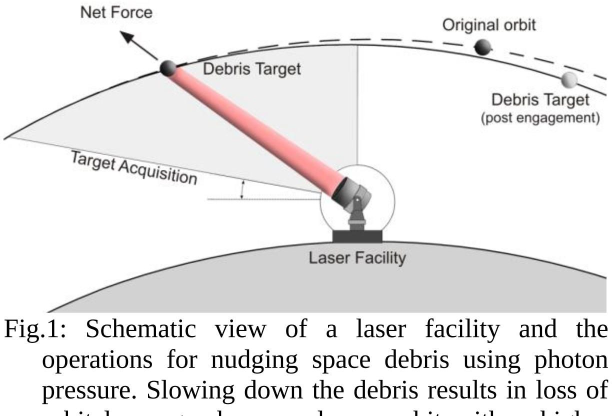


Fig.1: Schematic view of a laser facility and the operations for nudging space debris using photon pressure. Slowing down the debris results in loss of orbital energy, hence a lower orbit with a higher velocity. In general, both acceleration or deceleration can be useful to avoid a collision.

avoidance methods, because for ADR a given group of objects is removed from the environment at a given time. Investigating collision avoidance, however, is more complicated. For example, whether a specific LightForce collision avoidance engagement is successful or not will depend on the specifics of each close encounter between objects (a conjunction). Success will depend on the masses of the objects involved, their orbits, position of laser ground stations and ground station capabilities. The probability of success will depend on these and other inputs that cannot be easily derived and averaged and hence cannot easily be introduced into a tool like LEGEND that uses time-steps that are orders of magnitude larger than the actual engagement. Hence, we decided to work towards a long-term debris simulation tool that uses time steps on the order of seconds and is flexible enough to assess various debris remediation schemes and other events and scenarios of interest.

The second part of this paper outlines the goals for our long-term simulation approach, the chosen approach and the current status of implementation (including examples). We conclude the paper with a section on next steps and a summary.

1. PART I: LIGHTFORCE EFFICIENCY IN TODAY'S DEBRIS ENVIRONMENT

1.1. Goals of the investigation

This part of the paper summarizes the progress in our investigations that assess LightForce's utility as a collision avoidance method in today's space environment. Further details can be found in our past papers [9,10,11,12]. We have the following goals: 1) We want to assess LightForce's utility on a representative number of close encounters between space objects (conjunctions). 2) We want to make sure to provide a holistic investigation, also taking conjunctions into account that may be caused because LightForce engaged objects earlier. 3) We want to take into account that there is imperfect knowledge about future object positions and velocities. To meet these goals we use the following approach:

1.2. Simulation approach

Our analysis utilizes a simulation of the trajectories of all tracked space objects in Low Earth Orbit (LEO). We track all conjunctions and analyze the probability of collision for each of one, with and without LightForce being active.

The analysis follows three steps:

1) *Create a baseline simulation (without LightForce).* We utilize the publicly available two-line element (TLE) orbital data from the Joint Space Operations Center (JSpOC) to simulate the entire LEO environment and create a list of conjunctions that naturally occur during the simulated timeframe.
2) *Simulate an active LightForce system.* We propagate the same set of objects as in step 1, now including additional forces caused by an active LightForce system. All occurring conjunctions are recorded, also providing insight into secondary conjunctions.
3) *Assess the efficiency of the LightForce system.* Both the lists of conjunctions without LightForce and the list of conjunctions with a LightForce system are compared using tailored metrics.

In the following, we describe the three steps in more detail.

Step 1: Create a baseline (without LightForce)

As original input, we use the publicly available catalog of TLEs [13]. For each object the catalog provides a unique identifier and the orbital elements at a given epoch. Unfortunately, the catalogue does not directly provide an area-to-mass ratio. Also, single TLEs are error prone and have limited accuracy. To enable the best possible results (with reasonable efforts), we use least-squares fitting of TLE data as described by Levit and Marshall [14] to obtain an improved state vector, an area-to-mass ratio and a covariance uncertainty matrix. Details are described in [11].

The algorithm results in an object database consisting of state vectors, area-to-mass-ratios and object areas. Using the derived state vectors derived, we now propagate the orbits of the objects throughout the simulation time-frame and perform an all-on-all conjunction assessment. This gives us a sample of conjunctions based on real world data. If the probability of collision P_c for a given conjunction exceeds a threshold T_c, we save the data (object IDs, time of closest approach (TCA), P_c) in a list of high risk conjunctions. Pc is calculated using Patera's method[15].

Step 2: Simulate an active LightForce system

In the real world, a LightForce system will provide collision avoidance based on conjunction alerts. The accuracy of those alerts degrades over time due to inaccuracy of propagators. Hence, the data created in Step 1 should not be misinterpreted for an attempt to predict conjunctions a year in advance, but as a way to create a representative baseline, using real world inputs. For space operations, tracking is used to continuously update the orbital data and collision avoidance decisions are made short term. LightForce would be used in a similar fashion, reacting to incoming tracking data from various sources, including high accuracy laser ranging data provided by the LightForce stations themselves. Hence, we do not attempt to develop an optimal engagement strategy for the entire simulation duration, but optimize that engagement strategy for a shorter time frame (e.g. one week) and continuously update that strategy. Figure 2 illustrates that approach.

The first step is a “search run”, where we propagate the objects without LightForce for a given time frame (e.g. 9 days in advance, see fig 2). Occurring conjunctions are detected and an engagement strategy is developed using the following conditions, described in more detail in [11]:

A laser ground station will be tasked to illuminate an object, if all of the following apply:

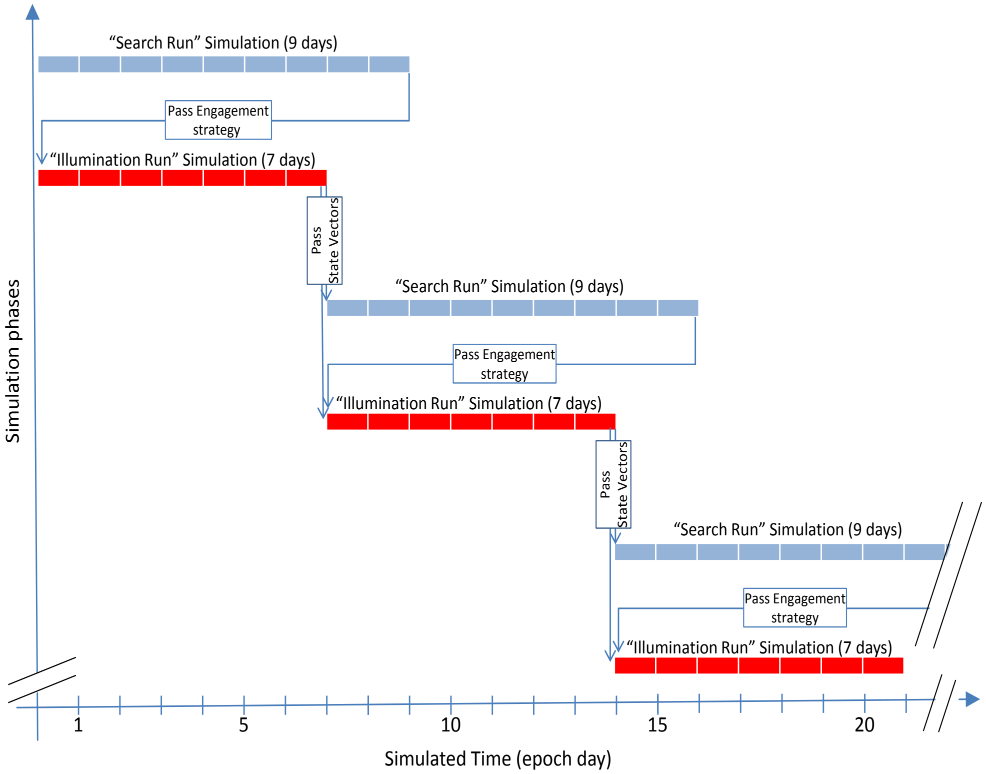
a) There is a line of sight between the object and the laser and the elevation angle is >10 degrees.

b) The time remaining to the time of closest approach for the specific conjunction is less than a set engagement time t_e .

c) Laser activation is beneficial for the chosen optimal collision avoidance strategy, which is either to slow the object down, or to accelerate it.

At the end of the search run the software switches over to simulate LightForce engagements, to an “illumination run”. Starting at day one, the engagement strategy from the first search run is now applied (e.g one week, red in fig 2). If a LightForce station is tasked to illuminate an object, the additional photon pressure derived force is added to the propagator. The illumination either happens during the first or second half of a pass, depending on whether accelerating or decelerating the object is more beneficial in reducing the P_c. All conjunctions and their final P_c are recorded throughout an illumination run. The state vectors of the objects at the end of the illumination run are input for the next pair of search and illumination runs. For the follow up search run, LightForce is turned off and a new illumination strategy is devised. The updated strategy is used for the follow on illumination run (fig. 2) which is a seamless continuation of the first illumination run, just with an updated illumination strategy that accounts for the changed environment. For the remainder of the simulation duration, search runs and illumination runs are alternating, simulating a LightForce system that reacts to incoming updated tracking data.

The duration of a search run is that of an illumination run plus the engagement time t_e , in order to allocate t_e of potential illumination time for each conjunction. For example, fig. 3 illustrates a 9 day search run followed by a 7 day illumination run, in order to allow for 48 hours of engagement time for each conjunction. This way, LightForce will engage a conjunction that occurs at the beginning of the second illumination run during the end of the first illumination run. Please note that the actual illumination time (where



“Search Run” Simulation (9 days)
Pass Engagement strategy
“Illumination Run” Simulation (7 days)
Pass State Vectors
“Search Run” Simulation (9 days)
Pass Engagement strategy
“Illumination Run” Simulation (7 days)
Pass State Vectors
“Search Run” Simulation (9 days)
Pass Engagement strategy
“Illumination Run” Simulation (7 days)
Simulation phases
1
5
10
15
20
Simulated Time (epoch day)

Fig. 2: Simulation Phases. Search runs are used to develop the engagement strategy (decide when to turn on LightForce). Illumination runs are used to record conjunction data and assess the LightForce efficiency

the debris is illuminated by the laser) will be much shorter than the engagement time, as each pass over a ground station lasts only a few minutes. The choice of a two day engagement time and seven day illumination runs was made in order to assess a basic system. Optimizing the strategy towards an ideal system is by no means trivial and would also require in depth simulation of the interaction of the existing tracking systems with the additional high precision laser tracking data provided by the LightForce system. That simulation would require a range of assumptions about the capabilities of existing and future tracking networks, hence we choose not to go that route and stick with a basic 9 day / 7 day schedule.

We also implemented a more complicated version which allows for search runs with activated LightForce system in order to check for producing accidental conjunctions with LightForce. If that option is used, search runs with and without LightForce are used to devise an optimized engagement strategy.

It is important to understand that while the computer is alternating between simulations that are used to either develop an engagement strategy or simulations used to test it, the final illumination runs are continuously using the same state vectors over the entire simulation duration (1 year), and record the final P_c of each occurring conjunction. These combined illumination runs build up a one year simulation of an active LightForce system. The results of that run are compared to the baseline in the next step.

Step 3: Assess the efficiency of the LightForce system

We compare the baseline list of conjunctions to the list from combined illumination runs (LightForce is active). We use two different metrics, the mitigation factor *M* and the reduction factor *R*:

The mitigation factor *M* is defined as

$$M = 1 - \frac{\text{Count conjunctions with} P_c > T_c \text{ with LightForce}}{\text{Count conjunctions with } P_c > T_c \text{ w/o LightForce}}$$

M can be derived directly from the simulation data when a reasonable threshold T_c is set. M answers the question what fraction *M* of conjunctions can a LightForce system mitigate, meaning, what fraction of high risk conjunctions can be mitigated to low risk conjunctions. This is useful for an operator who wants to know what fraction of conjunctions LightForce can mitigate.

The reduction factor *R* is defined as:

$$R = 1 - \frac{\sum\limits_{\text{all conjunctions with } P_c > T_c \text{ after LightForce activation}} P_c}{\sum\limits_{\text{all conjunctions with } P_c > T_c \text{ without LightForce}} P_c}$$

R provides an assessment on the global effect of a LightForce system on the debris environment, comparing the sum of P_c during the simulation time with a Lightforce system to the sum of P_c without. For T_c=0 and P_c based on exact error covariances, these these sums would represent the expected value of collisions. However, both conditions are not met. Hence R is a benchmark parameter that is only loosely related to the expected value of collisions.

1.3. Software implementation

Overview

The following sections describe the implemented physics and the chosen numerical approach. Simulating the positions of a large number of objects with high precision is computationally expensive. Hence we chose a highly parallel approach.

Propagator

The scheme used by the propagator for the numerical integration is a 4th/5th order Runge-Kutta scheme with variable time step [16, 17]. The forces taken into account during the propagation include Earth's gravitational field, the gravitational perturbations from the Moon and the Sun, atmospheric drag and the solar radiation pressure. The numerical implementation is built around the NAIF SPICE Toolkit [18] and the physical model used for each force is referenced in Table 1. We validated our propagator against STK's HPOP, an industry standard.

Laser induced photon pressure

Laser illumination entails four additional force components. Three of them are caused by conservation of photon momentum (photon pressure), a fourth is induced by temperature gradients in the surface of the illuminated object. The first force component is parallel to the incoming laser beam and caused by the momentum of all incoming photons. This is the most significant force component. Specular reflected photons add an additional force parallel to their outgoing direction (but with a negative sign). Diffuse reflection adds another force. Finally, temperature gradients on the surface could result in a net force through thermally emitted photons. However, surface reflectivities, as well as object orientation are not very well known for most of the objects. In addition, most objects over 600 km are assumed to be tumbling fast, which would result in cancelling the latter three effects for most cases [19]. Even if the object is not tumbling, the influence is comparably minor. Hence we ignore those additional effects and go with the conservative assumption of a debris object with zero reflectivity for the analysis presented in this paper.

Under this assumption, the additional force *F* on the object is [20]

$$F(t) = \frac{1}{c}\int I(x,y,t)dA,$$

where *c* is the speed of light, *I* is the irradiance at a point on the cross-section of the illuminated object at the time *t* . We update the irradiance for each time step. The irradiance *I* is calculated taking multiple effects into account. These effects are beam spread by diffraction, beam spread by atmospheric turbulence, and power losses by atmospheric absorption and scattering. All depend on the specific path between the laser ground station and the space object (determining distance and atmospheric conditions) and the technical specifications of the stations.

Table 2 in Section 4 (simulation results) summarizes those specifications. We assume a ground station with adaptive optics and a laser guide star to compensate some of the effects of turbulence. As assumption for the performance of the adaptive optics system we use the results of 1998 benchmark experiments on an adaptive optics system for a directed energy weapon system, compiled in a study of the American Physical Society [21]. Combining the different effects result in the irradiance and the force on the object. The details of the calculations are complex, please see references [9,10,22] and references therein for a step-by-step description.

Probability of Collision

We follow the method described by Patera [15]. For each conjunction we determine both the real and the maximum probability of collision [23, §11.7.2]. The real probability of collision takes into account the covariance determined by the initial TLE fit in step 1, while the maximum probability of collision is a value obtained by varying the orientation and the size of the uncertainty ellipsoid defined by that covariance. To be on the safe side, we evaluate the performance of the laser photon pressure against the maximum probability of collision for subsequent calculations. It is the maximum probability of collision we commonly denote as P_c.

Software architecture & All on All conjunction analysis

The software utilizes the Pleiades supercomputer at NASA Ames. Pleiades is a Linux cluster made of Intel Xeon processors. Our current software is implemented in C and C++ and uses the standard Message Passing Interface (MPI) for parallelization. On 1000 cores, it achieves a processing performance of approximately 5000 times real time for 12,000 objects.

After initialization, the code produces a time series of state vectors using the high precision propagator. The propagator uses a relative tolerance setting and dynamic time-steps. The propagator is allowed to "run free", producing a new position/velocity “anchor” point for an object whenever it is appropriate to maintain the specified relative error tolerance. This is done fully in parallel, with each MPI rank doing heavyweight propagation on a subset of the objects, and then exchanging the results with the other ranks. When complete, each MPI rank has the full set of information about every object.

With these anchor points in hand, each MPI rank interpolates the position of all objects for the beginning of the current time step, starting from the anchor points. These positions are then sorted along the X-axis. The code next does collision detection. Each MPI rank assigns itself a subset of the objects, and decides if an object might possibly interact with some other object during the current time step. Ultimately, this determination is made by calculating the time of closest approach, which determines whether a conjunction occurs in the current time step. If a conjunction occurs, at the TCA for a pair of objects the associated state and co-state information is used to calculate the probability of collision and the conjunction is recorded.

After the current time step is complete, we advance the clock and interpolate to the beginning of the next time step. If we no longer have state vectors available to reach the end of the next time step, the propagator is used again to calculate new “anchor point” state vectors. The cycle continues until the simulation is complete. The physics of applying LightForce (or not) is handled in the propagator.

A primary concern in this work was producing consistent answers. Since the propagator uses the current state as the basis for predicting the next state, even very tiny differences are quickly magnified, and after a few days an object could be many kilometers away from the position predicted by some previous run. Neither is better or more correct than the other, and both will probably produce very similar statistics. But it is

Table 1: Forces taken into account in the propagator and models used for their numerical implementation.

Force	Numerical implementation	Reference
Earth’s Gravitational Field	Earth Gravitational Model 1996 (EGM96)	[24]
Luni-Solar Perturbations	NASA JPL Planetary Ephemerides	[25]
Atmospheric Drag	NRL-MSISE-00 model	[26]
Solar Radiation Pressure	Debris modelled as a sphere, eclipses taken into account	[23]
Laser Radiation Pressure	In-house model	[9]

not easy to compare the two directly.

Great effort was expended to make it at least possible to avoid this sort of problem. For example, the code may save the state vectors at any time and then later restart from reading that file. This gives bitwise identical results compared to running the simulation straight through.

1.4. Short-term simulation results

Input parameters

In the following we summarize simulation results on the efficiency of a network of LightForce ground stations for space debris collision avoidance originally published in [12]. The presented case uses a set of parameters which are introduced in this section.

For the orbit propagation, we used the force models summarized in Table 1. The start date of our simulation was June 15, 2012 and 5 TLEs for fitting were acquired for each object before that date. We restricted the analysis to orbits with a perigee below 2000 km.

To compile the baseline list of conjunctions, we performed the previously described all-on-all conjunction analysis with a threshold T_c of 10^{-6} because it appears to be the standard value at which major international space players, commercial and institutional, start to get interested in the P_c metric. Actual collision avoidance maneuvers (using satellite maneuvers) will not be initialized until P_c is orders of magnitude higher [27].

The input parameters for the laser force model are stated in Table 2, translating to commercial off-the-shelf technology where possible, to cut down the cost of a potential system. For the same reason, assumptions about the adaptive optics technology are based on 1998 benchmarks [9, 10, 22]. The engagement time t_e is 48 hours, meaning that LightForce begins engaging objects 48 hours before the TCA for a specific conjunction. In this paper, we always assume a set of four stations with 20 kW laser output power, placed at the locations specified in table 2a.

We do not constrain our analysis to certain sun illumination conditions, but assume a laser engagement for each pass over a ground station in compliance with the requirements of Section 1.2 step 2.

The engagement strategy is updated on a weekly basis, using the state vectors at the end of a seven day illumination run and propagating for nine days (as in fig. 2). We choose to implement a double optimization cycle, propagating a first search run without LightForce and then another one with LightForce to ensure to capture potential secondary conjunctions. The simulation duration was one year.

There is still room for further optimization, because one would likely update the engagement strategy whenever new tracking data is available in a operational scenario, and not in seven day cycles.

Table 2a: Laser ground station locations for efficiency simulations

Location	Lat.	Long.	Altitude [km]
Antarctica (Ant.)	-80.4	77.4	4.1
Hawaii (HI)	20.7	-156.3	3.0
Australia (Aus.)	-35.3	149.0	0.8
Alaska (AK)	64.9	-148.5	0.5

Results

The first step to assessing LightForce is to create a baseline that will be compared to the results with an active LightForce system. In the simulation period spanning from June 15 2012 to June 14, 2013, approximately 30,000 conjunctions with $P_c > 10^{-6}$ are detected. There is a 4% decrease of propagated objects over 12 months, caused by natural decay of objects into the atmosphere. As both the baseline and the LightForce simulation use the same assumption, we do not expect any significant impact to the efficiency metrics presented.

As the next step, we assess a simulation with active LightForce stations, record the remaining conjunctions with $P_c > 10^{-6}$ and compare them to the baseline. Fig. 4 shows the effect of the defined 4 station LightForce network (Antarctica, HI, Aus, AK, see Table 2a) with a 20 kW laser each in a histogram view. It shows the distribution of the number of conjunctions over the defined P_c intervals.

Table 2: Laser ground station parameters used for efficiency simulations

Laser	IPG YLS-10000-SM	Telescope diameter	1.5 m
Power	20 kW continuous	Atmosphere model	US Standard 1976
Wavelength	1070 nm	Aerosol model	MODTRAN rural (VIS=23 km)
Beam quality	M^2=1.3	Turbulence model	Hufnagel/Valley 5/7
Engagement time t_e*	48 h	Adaptive optics	performance according to [21], Fig.21.1; additional beam degradation by tip/tilt anisoplanatism, see[21], appx. D4.4

* Each object is engaged while passing over a ground station in a 48 h window before the time of closest approach of the specific conjunction.

The total number of conjunctions with P_c>10^{-6} is decreased by 85%, resulting in an overall mitigation factor M=85% (as defined in section 1.2). Fig. 5 shows the mitigation factor M, for each of the P_c intervals defined in fig. 5. The remaining conjunctions are mostly naturally occurring conjunctions but also include conjunctions that appear because the whole space environment is changing due to LightForce application to some of the objects. However, it is obvious that the overall effect to the environment is positive.

This result is reinforced by assessing the cumulative P_c for all occurring conjunctions. Fig. 6 shows the increase of cumulative P_c over time, comparing the baseline case with the LightForce case. The observed steps occur when high risk conjunctions are detected. The cumulative increase is less with LightForce active, translating into a reduced number of expected collisions. The overall reduction factor R (see section 1.2) for the entire simulation duration is 94% (counting conjunctions with P_c <10^{-6} as zero), including all conjunctions that occur in the modified debris environment.

Finally, fig. 7 shows the influence of the simulation duration on the calculated M and R factors. Both stabilize with increased simulation duration. M shows less variation than R, as high risk and low risk conjunctions are weighed equally for M. High risk conjunctions dominate the R factor. As there are fewer of those, it takes more time for R to stabilize. Nevertheless the results are stable for the second half of the 12 months simulation duration.

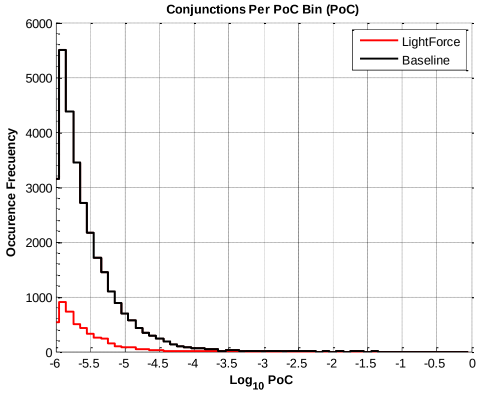


Fig.4: Histogram of the distribution of conjunctions with Pc >10-6 detected over a 52 week period. : Baseline: without LightForce. LightForce: Remaining conjunctions while 4 20kW LightForce stations are active. Each interval is defined as 10-6+0.1*n <Pc<10-6+0.1*n+1; starting with the interval 10-6<Pc<10-5.9 and increasing.

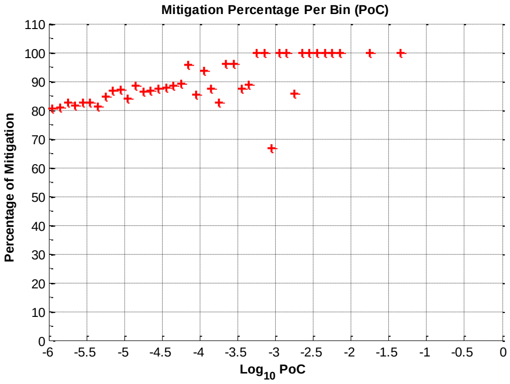


Fig.5: Mitigation factor M assessed for conjunctions sorted by Pc. Each point in the plot represents calculation of M for all conjunctions that originally appear within a $10^{-6+0.1*n}$ <P_c<$10^{-6+0.1*n+1}$ interval, starting with the interval 10^{-6}<P_c<$10^{-5.9}$ and increasing. Note: The underlying data gets sparse for P_c>10^{-3}

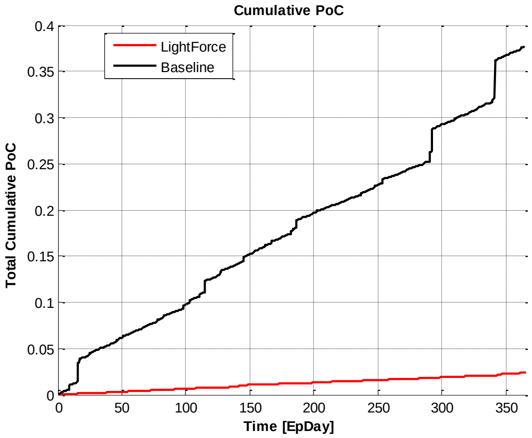


Fig.6: Cumulative P_c plotted over the duration of the simulation for both the baseline and the LightForce case.

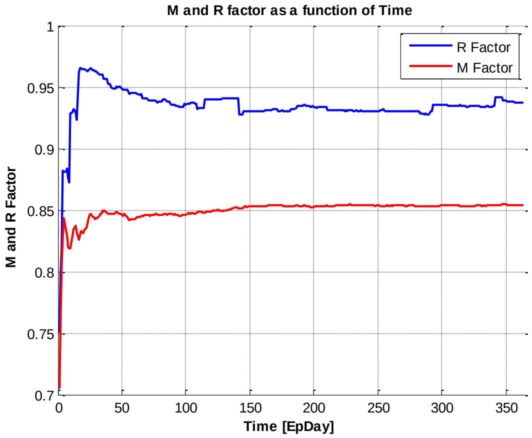


Fig.7: M and R factors depending on simulation duration (see section 1.2 for definitions)

2. PART II LONG-TERM DEBRIS SIMULATION

2.1. Goals for a long-term simulation

The results described in the last section are promising in regard to the utility for LightForce as a collision avoidance capability in today's debris environment. However, the remaining (unmitigated) conjunctions include heavy objects above 150 kg, which can be a major source of new debris. Still, preventing collisions between those and smaller objects would have a positive impact on the debris environment. The challenge is to quantify that impact. The ultimate motivation behind the upgrades of the simulation software described below is to provide that quantification. We describe the current status on the way towards that ultimate goal.

We have the following requirements for the software upgrade: 1) The software needs to project the long-term (several decades to hundred years) development of the debris environment, including new launches, collisions and spontaneous explosions. 2) The software needs to be simulating interaction on short time scales (seconds) in order to allow the implementation of collision avoidance maneuvers. 3) We want a flexible approach that enables us to assess various debris mitigation and remediation approaches.

The following section describes our approach.

2.2. Software changes to enable a Long-term simulation approach

2.2.1. General overview

The main challenge for long-term debris projections is that single events can determine the state of the entire environment. For example, combined, the destruction of the Chinese Fengyun spacecraft in 2007 and the 2009 collision between Iridium 33 and COSMOS 2251 doubled the population of fragmentation debris in orbit. While the first event was probably intentional, the second was one out of tens of thousands of close encounters (conjunctions) between objects that occur during a year. The simulation of each event is of a highly statistical nature, as minimal changes in initial positions and velocities of space objects, as well as changes in external forces, lead to significant changes in future positions that are several orders of magnitude larger than the object sizes. Similarly, changes in position on the order of the object size influence the likelihood of a collision. Hence, a Monte Carlo approach is needed to produce a projection of a likely future, randomizing initial object states, as it is done in NASA's LEGEND tool [7, 8].

Contrary to our second requirement for short time scales, LEGEND has a default time-step of five days. In order to gain the capability to assess the effect of arbitrary changes to individual space objects onto the environment, we choose a simulation approach where every object's position and velocity are simulated over time in high resolution (on the order of seconds, not days). Nikolaev et al. coined this as a *brute force* approach when they implemented a similar approach [28]. This gives the ability to manipulate arbitrary objects as needed to simulate collision-avoidance engagements, and also gives further options to simulate other scenarios, e.g. the deployment of drag devices or de-orbit maneuvers. Since each change happens on a timescale of seconds, this approach is computationally intensive, but allows high precision and maximum flexibility to implement various scenarios.

The software described in the first part of this paper provides a solid foundation to achieve the requirement lined out in section 2.1. The following changes are needed.

2.2.2. Orbit propagator

The physics of object propagation implemented before (section 1.2) is sufficient for long-term simulations. Major changes to the software have been implemented on the numerical side, because objects need to be added dynamically during a simulation run to simulate collisions, spontaneous explosions, and new launches. Especially in a parallel computing environment, that is not trivial. The simulation framework is designed to allow for addition and deletion of objects that either occurs at a certain time as defined by input files, or internally. Internal deletions happen when objects decay below a cut-off altitude, or when explosions or collisions occur. Spontaneous explosions are triggered randomly, as described later. The original object will be deleted and replaced by a debris cloud predicted by the NASA standard breakup model (see section 2.2.3 for details). The same approach for object deletion and creation is taken for object fragmentations that are caused by collisions between objects.

We have conceived additional changes to the propagator which would allow additional flexibility to simulate specific scenarios. For future revisions of the software, one could force the propagator to follow a predefined trajectory for an object, e.g. to implement specific de-orbit maneuvers or foul play. Another option to be implemented is to change object properties at given times, e.g. to account for the deployment of drag-enhancing devices at the end of a spacecraft's mission.

2.2.3. All-on-all conjunction analysis

Changes to the all-on-all conjunction analysis were needed to deliver increased accuracy of the distance between objects at TCA, as object overlap was chosen to trigger collisions (see section 2.2.3). We use a similar approach as described in section 1.3, but changed from linear to spline interpolation to calculate distances of closest approach. Both for the distance and the time of closest approach we now use an algorithm that uses spline interpolation between the state vectors of the two objects [10, p. 919-937]

As the propagator uses variable time-steps, determined by relative error tolerance, it is highly unlikely to find high-precision state vectors (referred to as “anchor” points) produced the propagator for both objects involved in a conjunction at the same time. As shown in figure 8, we always take one anchor point (of one object) and use spline interpolation to generate the state vector for the second object. We end up with four state vectors (two objects, both at t_1 and t_2) that are used to calculate the TCA and miss distance. As all related calculations are computationally expensive, the code first uses a couple of simple tests to try and (cheaply) eliminate the possibility that the two objects could interact during the investigated time step.

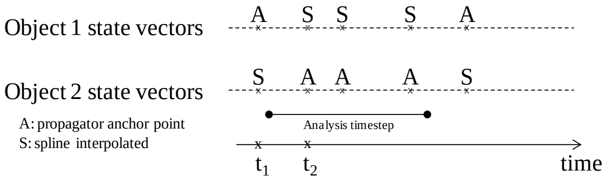


Fig.8: Interpolation process used by all-on-all conjunction analysis, utilizing propagator output (anchor points).

Note that the method for producing the t_1 and t_2 state vectors is independent of the size of the analysis time-step in the conjunction analysis and depends only on the anchor points produced by the propagator. This enables the simulation to dynamically vary the size of an analysis time-step for optimal speed, yet still maintain exact bitwise reproducibility.

2.2.3 Fragmentation through Collisions and Explosions

A major addition to the software was the inclusion fragmentation models for both collisions and spontaneous explosions.

The process of randomly triggering explosions is dependent on the input scenario, as described in section 2.3.3 below. To trigger collision fragmentation, we use the (miss) distance between objects at time of closest approach as the final benchmark to decide whether a collision has occurred. If the distance is closer than the combined object radii, fragmentation is initiated. This method is the same as used by Nikolaev [28]. It has the advantage that it is independent of state covariances, which might not be accurate. While (even with perfect initial states) the propagator is not able to predict an object position with an accuracy that is comparable to the object size, this method still gives reasonable results in combination with the Monte Carlo method, because uncertainties are modeled through the inherent randomness of the method [28].

The inserted debris clouds for both explosions and collisions are created using the approach outlined in the NASA EVOLVE 4.0 breakup model [29, 30]. We make the following three implementation choices: 1) EVOLVE provides area-to-masss distributions for fragments that are smaller than 8 cm and larger than 11 cm, but not between 8 cm and 11 cm. In order to close the gap we chose randomly between one of the two distributions. 2) Velocity change for the objects in the fragment cloud is assigned using the distributions given by EVOLVE. The direction of the velocity change is randomly assigned. 3) For explosions, we determine the number and size of pieces using a scaling factor S=1, which is valid for upper stages between 600-1000 kg.

2.2.1. Data analysis

We continue to track all conjunctions above a user-definable probability of collision and miss distance threshold. The probability of collision [15] and the state vectors of the objects involved are stored. As change to the previous software, the simulation code now also tracks the number of objects in orbit, divided by object class, i.e. debris, spacecraft and rocket bodies. New debris objects that are created during a simulation run are categorized by debris objects that are created by explosions and by collisions. In addition, state vectors of all objects involved in breakups are stored at the time of the breakup event. Finally, the simulation periodically stores state vectors of all objects with additional data including potential de-orbit time and the parent objects of newly created fragments.

The combined data products enable various analyses: On a global level, the development of object numbers over time gives insight into the development of populations and the influence of debris mitigation or remediation measures. The number of conjunctions gives a first insight into the development of collision risk. Tracking the probability of collision for each conjunction enables more sophisticated analyses, e.g. tracking the cumulative value. More detailed analyses for a class of spacecraft, a specific constellation or a specific spacecraft are possible as well, always comparing different scenarios or a different time frame in a specific scenario.

A Monte Carlo analysis of multiple simulation runs is the preferred method to gain information on simulation uncertainties, but we have not implemented that feature, yet. Currently, the software is ready to produce data of single runs. A Monte Carlo wrapper has not yet been implemented. The data shown in what follows thus represents “one possible future”, but not an average projection with error bounds.

2.3. Scenario development

2.3.1. Initial population

A simulation run assesses a specific input scenario. Each scenario provides an initial population, space weather, defines future launches and spontaneous explosions.

To provide the initial population of space objects, the code accepts a file with a list of objects that includes object state (position and velocity), area-to-mass ratio, area, ballistic coefficient and reflectivity, the state covariance and the epoch for that data. For statistical purposes, we also provide the object type (rocket body, spacecraft or debris).

The approach for generating this file is very similar to the one described in section 1.2 (step 1). As we aim for multi-decade simulations, we implemented some measures to improve the accuracy in orbital life-time prediction that were not necessary for a one year simulation. Also, data provided by space-track has changed. The changes are summarized below:

1) The start is updated to June 15, 2015. Out of 12725 records for objects in LEO, a successful fit was provided for 10256 objects. For the remaining objects we use a direct translation from TLE to state vector and assign a covariance matrix corresponding to the average covariance of the successfully fitted objects.
2) The area-to-mass ratio guesses from step 1 are not accurate enough to result in reasonable object life times for approximately 30% of the objects. To increase the accuracy of the area-to-mass ratio over the method described in [11], we use a list of area-to-mass ratios provided by our colleague Wang Ting (Princeton). He uses on average 500 TLEs to determine the area-to-mass ratio through fitting the semi-major axis decay of an object and also provides an RMS error of that fit. From his list, we removed 275 entries where the relative RMS error of the fit is larger than one, as well as 113 entries for negative area-to-mass ratios, the latter indicating maneuverable spacecraft. In total, we can match 11050 objects from the 12725 objects from step 1. Out of the remaining 1675, we provide values for about 70 of the 113 maneuvering spacecraft from a 2008 snapshot of ESA DISCOS data [31], using the average between the provided minimum and maximum area and the mass at beginning of life. For the rest we draw random area-to-mass ratios from objects belonging to the same class as the object with the missing data. For example, for each spacecraft with unknown area-to-mass ratio we assign the area-to-mass ratio of a random spacecraft with known data. Using that approach we preserve the original area-to-mass distributions.
3) The cross-sectional areas of the objects are assigned using radar cross-sections (RCS) in a first step. Space-track provided that data until July 2014. We convert RCS to a physical cross-section using the method given in [32]. After July 2014, space-track has been providing only classifications into “small”, “medium” and “large” RCS. These groups correspond to defined RCS bins. For all new objects with only classifications but no specific RCS data, we draw one existing RCS out of the appropriate group of “small”, “medium” and “large” objects, and translate that to a physical size.
4) The method described in [32] to translate RCS to physical size was developed for debris (using samples from hypervelocity impacts) and is not necessarily accurate for intact spacecraft and rocket bodies. Hence we replace the cross section for intacts with data from the 20008 DISCOS snapshot [31]. We find matching data for 2416 out of 3481 entries for rocket bodies and spacecraft. We translate the average dimension into an area, only omitting the IMAGE and Picosat 1&2 spacecraft. IMAGE is equipped with 4 250m radial antennas, which would translate into an unrealistic area. Picosat 1&2 are tethered. If no data is available, we keep the data derived from RCS data.

For ballistic coefficient we assign the value of 2.2 and 1 for reflectivity. Future work would include incorporating updated DISCOS data, as well as other available data for known objects.

2.3.2. Future Launches

The code accepts a list for future launches that is mostly identical to the one provided for the initial population. Instead of the epoch time for the fit, this file provides a date and time that specifies when the object is to appear in the simulation.

There are several options to create a future launch scenario. The simplest option is to set future launches to zero. A common practice for long-term space debris simulations to define a period of time (e.g. the last 10 years before the start of the simulation) and use the list of objects launched within this period recurrently as a forecast until the end of the simulation period. Although this method may provide reasonable future scenarios in a stable environment, the current satellite market is going through a transformation where commercial “new space” companies have started to dominate the numbers of newly launched spacecraft. We chose to develop a scenario based on the assumption that these new companies thrive.

To develop an optimistic “new space” scenario, we systematically gather available data on future launches and collect it in a database. We aim to build a database that covers all the publicly available launch related information regarding the companies which intend to launch satellites into Low Earth Orbit (LEO) between 2015-2030. These companies/constellations include, but are not limited to, Blacksky, CICERO, EROS, Landmapper, Leosat, Northstar, O3b, OmniEarth, OneWeb, Orbcomm, OuterNet, PlanetIQ, Planet Labs, Radarsat, RapidEye Next Generation, Sentinel, Skybox, SpaceX, and Spire. Data is gathered either through direct contact with the company or from online resources (i.e. company press releases, published interviews). Information collected includes statements on the number of launches for each year between 2015-2030, on the number of orbital planes the constellation will be distributed to, as well as apogee, perigee, inclination, spacecraft mass and area. Whenever data are not available, estimations were made considering the constellation’s purpose and company’s previous missions, if any. The database also takes into account possible newcomers into commercial earth observation and telecommunication markets as well as the replenishment launches of the current and upcoming constellations. Figure 9 shows a summary graph generated from that information for the time between 2015-2030.

Launched rocket bodies are strongly correlated to the number of spacecraft. However, another trend is that more spacecraft are being carried as secondary payloads or are deployed from the International Space Station. Therefore we decided to use the historical positive-sloped trend for launched rocket bodies as a baseline and add approximately 30 launches per year on top of that baseline starting from 2017, to account for the new constellations. Not all of these rocket bodies decay because of drag. Our analysis of the online satellite catalog (SATCAT) data [13] shows that between 2007 - 2015 around 25% of the upper stages in LEO decayed within the first 10 days after their launch date. Assuming a positive trend, with more strict rules and potential use of reusable launchers; we build the scenario around an assumption that 30% are de-orbited and 70% of rocket bodies naturally decay. The total number is plotted in figure 10. The apogee, perigee, and inclination data for these objects were estimated in correlation with the information gathered on announced spacecraft launches and the historical trends.

For modeling 2030 onward, available information is increasingly sparse. Therefore we chose to build the scenarios extrapolating the 2015-2030 data on yearly basis.

The compiled list of predictions for satellites and rocket bodies with their annual number estimates, as well as their apogee, perigee, inclination, mass, and area information needs to be transformed into an input file for the orbit propagator. An automated script was built to pull the necessary parameters from the launch database and convert them into a suitable format for the simulation.

The script converts the estimated Keplerian elements of each object into Cartesian position and velocity state vectors at the predicted launch epoch. The database does not provide details on the orbits, hence, the argument of periapsis, eccentricity, true anomaly and Right Ascension of the Ascending Node (RAAN) were assigned randomly using uniform distributions within the domains of each element. For Sun-Synchronous orbits, the script calculates the required inclination depending on the altitude and eccentricity of the object.

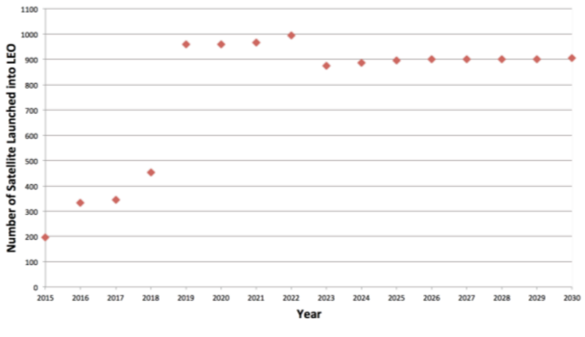


Fig. 9: Spacecraft launch scenario data built on compiled data on announced satellite launches to LEO

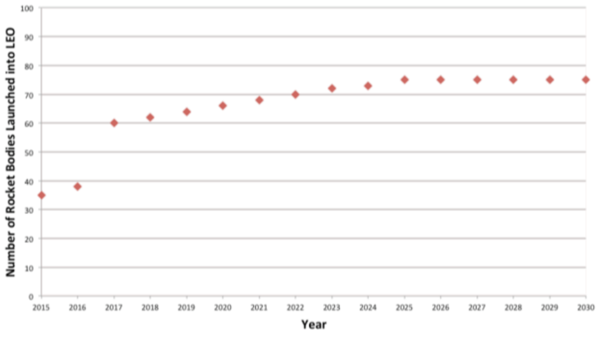


Fig. 10: Rocket body launch scenario

Launch epoch dates were assigned randomly within the launch year for each constellation from the database. The approach assigned a maximum of 15 objects for the same launch. Additional parameters, i.e. area-to-mass ratio, drag coefficient and reflectivity are assigned to each object according to their physical specifications. The positional covariance matrix was set to an average for all new objects. In order to calculate this average, the full set of objects from the June 2015 initial population is used.

2.3.3. Spontaneous on orbit explosions

Another source of orbital debris is spontaneous in-orbit explosions. In order to set a probability for those events, we evaluate the list of explosions given in the 14th edition of NASA's "History of On-Orbit Satellite Fragmentations" report [33]. On a yearly basis, this comprehensive list is correlated with the number of objects on orbit to obtain a measure for the probability of such an explosion. Figures 11 a and b show a general trend to a lesser probability of explosion per object over time for both rocket bodies and spacecraft. The x-axis shows the time in 5-year blocks, where the first period referes to the slot between 1963-1967, the second to 1968-1972, and so on. We fit that data to an exponential function.This function is used in the scenario to randomly trigger explosions, using the time-dependent number of rocket bodies and spacecraft as input.

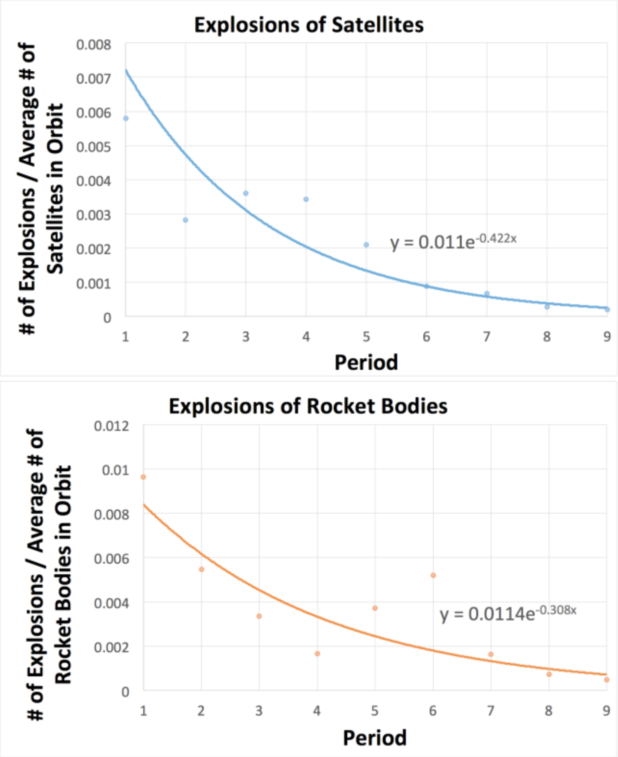


Fig.11: of explosions divided by objects on orbit over time for satellites and rocket bodies. Period 1 referes to the slot between 1963-1967, period 2 to 1968-1972, ... Exponential fit added.

2.3.4. Space Weather and SPICE input files

In order to allow long term simulations over a 100 year time frame, both the space weather input and the NAIF SPICE input data need to be provided.

The space weather data, which is needed to model the atmosphere's response to the Sun's activity, has to be projected up to July 2115. Our method of extending this data is purely statistical, not physical, and so will not capture all of the intricacies of the Sun. However, it should still capture the general trends, such as periodic variability with the Solar cycle, along with the large degree of day-to-day variation seen in observational data.

The method implemented here makes use of a significant amount of observational data, obtained from [34]. This data extends from January 1st 1957 to July 10th 2015, covering 4 complete cycles (minima and maxima). The data necessary for the NRL-MSISE-00 is the daily value of the 10.7cm flux (F10.7), along with the 81-day centered average of F10.7, and the daily values of the planetary index A_p in 3 hour intervals and a daily average.

In order to generate reasonable data for F10.7, we first determine its period by fitting a cosine function to the data: $f(x) = a_1 \cos(a_2 x + a_3) + a_4$.

The period is 10.79 years. The observational data includes four complete cycles, which we use as input. For the date of interest, we use the corresponding date in each of these cycles and consider a 6 month interval centered on this date. Combining these intervals, we have a distribution of F10.7 values, from which we select a value at random for our future date. The use of such large intervals is an attempt to mirror the short-term variability of F10.7 by providing as many potential values as possible. The results of this technique are shown in Figure 12. The 81-day centered averages are produced from that dataset. At the end of the dataset the input interval for each if those averages is reduced to the remaining values.

For the planetary index A_p there is no obvious repeating pattern. Hence, we use the observational data to create a probability distribution (table 5). According to this distribution an appropriate number of integers are randomly drawn out of each interval.

It should be noted that the use of daily values would be the ideal method of implementing this data. However, to reduce strain during data lookup, we approximate the data by using the values on the first day of each month as constant throughout that month. This still produces reasonable results, with the average altitude of the same object after a year differing by only tens of meters using the two methods.

In addition to the space weather, the NAIF SPICE toolkit needs planetary ephemerides and Earth orientation input data, which were provided by the JPL/NAIF team.

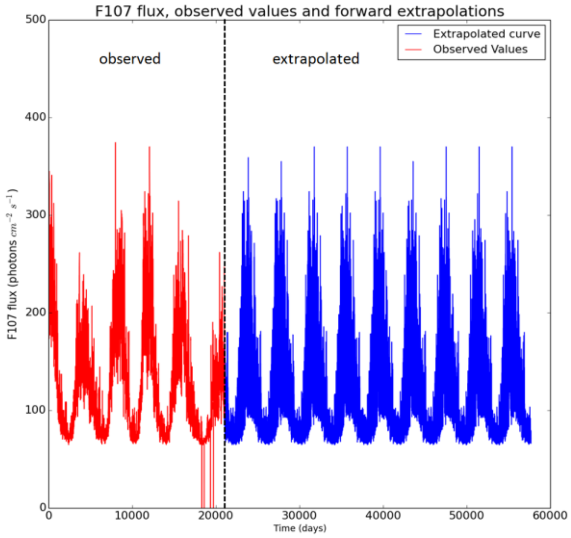


Fig. 12: Predicted values of F10.7 up to December 31st 2115. Note our curve appears “thicker” than the observed data. This is likely due to the fact that our data varies more than the observational data, given that it was produced by a statistical technique. For instance, the value of F10.7 on any given day has no bearing on the value the following day, which is unlikely to be the case in reality. For a similar reason, our data contains no relatively quiet Solar maxima, despite two appearing in the observational data (including the current one). This is because values from more active maxima are more likely to be selected. So even if a value from a quiet maximum is selected for one day, the next day, a value from an active maximum will likely be drawn.

Table 5: Observed historical distribution of the planetary index A_p and average difference of the implemented data to the historical observations

Interval	Proportion in Observations	Average Difference from Observed Proportion
0-10	52.732%	3.66%
10-20	28.372%	6.95%
20-40	14.265%	3.741%
40-80	3.741%	0.082%
80-120	0.614%	0.033%
120-160	0.171%	0.017%
160-200	0.067%	0.01%
200-240	0.024%	0.003%
240-280	0.009%	0.006%

2.4. Example simulations

In the following section we provide three examples of data products generated by the current software implementation. For all cases, we used used the environmental data as described in the last section.

Example 1: Decay of the June 2012 debris population, ignoring any future collisions or explosions

Figure 13 shows a plot of the debris population over time. The initial input was the June 2012 debris population from the space-track catalogue [13], using steps one to three from section 3.2.1 for data conditioning. Collision and explosion functionality had been disabled for this specific simulation run. Hence, figure 13 shows the natural decay of debris objects. The periodic changes in decay rate are a result of the influence of the solar cycle on the drag caused by the upper atmosphere.

Example 2: LEO population for optimistic “New Space” scenario including collisions and explosions

For the data plotted in figure 15, full collision and explosion functionality of the code are enabled. The initial population is using the space-track catalogue from June 2015 and full data conditioning as detailed in steps 1 to 4 in section 2.3.1. On top of that initial population we are introducing additional objects to the population over time, based on the results of the projections described in section 2.3.2 and 2.3.3. This assumes a thriving “New Space” economy, fulfilling currently announced plans for new satellite constellations and the required replenishing launches to keep them operational. This accounts for approximately 900 launched spacecraft per year. In this (singular) simulation run, the first collision occurs in late 2023. It involves two rocket bodies at 850 km, producing over 5000 pieces of debris. Further major collisions occur in late 2029, early 2030, 2034 and 2035.

Example 3: Number of LEO conunctions for “New Space” scenario

This example uses the same scenario as in example 2. Figure 14 shows the number of conjunctions with a probability of collision larger than 10^{-4} in each year.

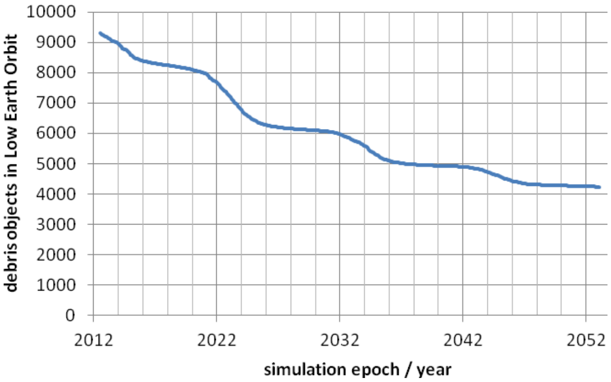


Fig.13: Simulated number of debris objects in Low Earth Orbit over time. Collisions and explosion functionalities are disabled.

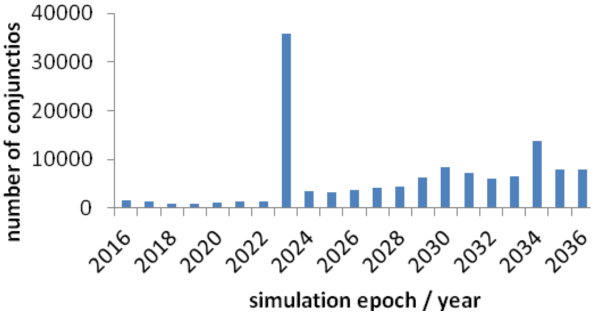


Fig. 14: Number of conjunctions with a probability of collisions larger than 10^{-4} for one example run using the optimistic “New Space” scenario.

There is an obvious increase over the years. The absolute maximum in 2023 is caused by follow up conjunctions after the initial collision

3. NEXT STEPS

While single runs already provide interesting information, e.g. about the total number of conjunctions in a certain debris environment, error bounds and average projections can be obtained with a full Monte Carlo treatment. Implementing that will be our main task for the future.

Further steps aim to increase flexibility in scenario development. We want to:

- Enable execution of predefined trajectories (e.g. for launches or de-orbit maneuvers),
- Enable changing object properties at given times (e.g. to simulate attachment of tethers or drag enhancement devices), and
- Increase the number of objects that can be simulated to enable the assessment of an environment with more particles or with broader knowledge of smaller particles.

4. CONCLUSION

We presented the status on our research on the LightForce just-in-time space debris collision avoidance concept. The first part of the paper presented an assessment of the efficiency of LightForce in today’s debris environment, utilizing a highly parallel simulation approach implemented on the NASA Ames’ Pleiades supercomputing cluster. Results indicate that utilizing a network of four stations with 20kW lasers, 85% of all conjunctions with $P_c > 10^{-6}$ can be reduced to below $P_c < 10^{-6}$. The reduction factor R that compares the cumulative P_c for a situation with a LightForce system to one computes to 90%.

The second part described our approach towards assessing LightForce’s utility in an evolving debris environment. We described our chosen methodology for

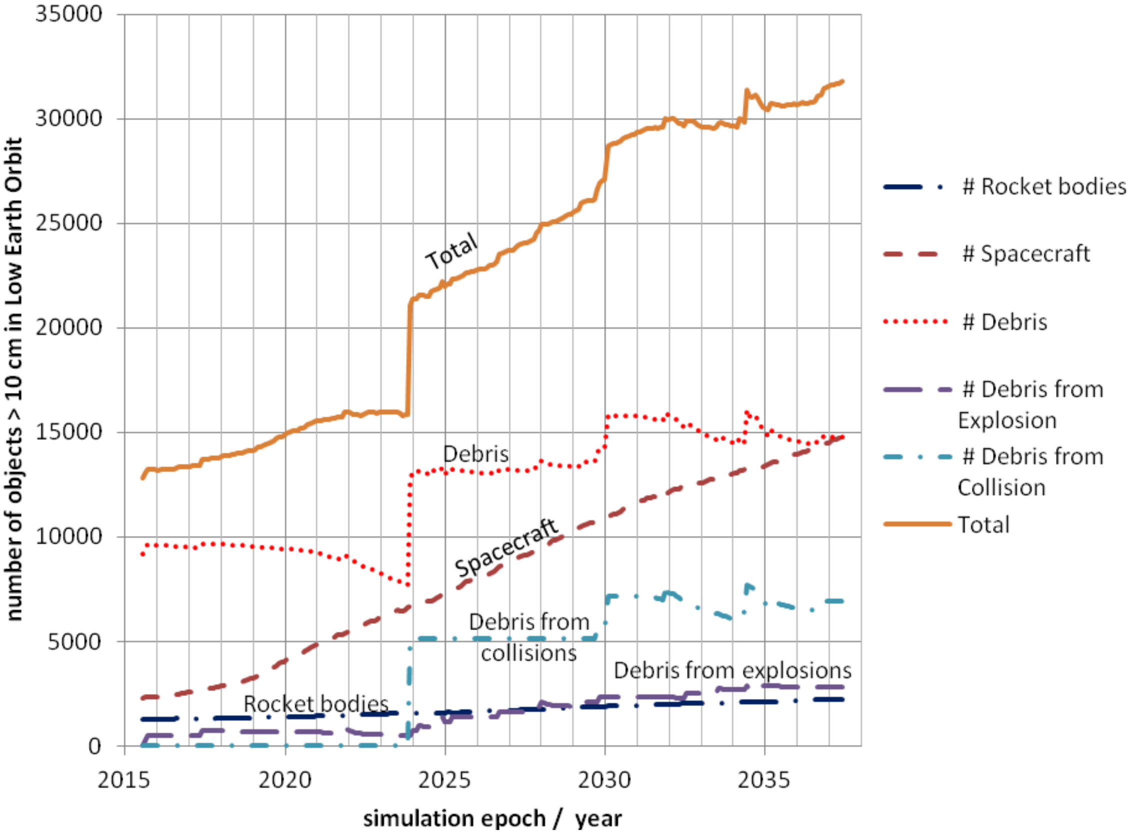


Fig. 15: Number of debris objects in Low Earth Orbit over time for one example run using a thriving “New Space” scenario.

a flexible, open scenario space debris simulation software, as well as its current status of implementation. We expanded our a highly parallel simulation approach. The current implementation propagates all objects in the simulation with a high precision propagator. Time steps are chosen dynamically, and determined by the chosen relative tolerance setting. We further described our approach for the development of input scenarios that includes both the current space environment, including spacecraft, rocket bodies and debris and also projects future launches. For the future launches we detailed one specific scenario that assumes a successful development of the current "New Space" economy. We provided examples of data outputs of the current software suite.

5. ACKNOWLEDGEMENTS

We would like to thank our colleagues and the center management at NASA Ames Research Center for continuing support. We also would like to thank the NAIF SPICE team at JPL, especially Nat Bachman for providing a long term version of their Earth orientation file. We thank Cyrus Foster both for his past contributions, implementing the original propagator, and for useful discussions over the last months. Special thanks go to Wang Ting (Princeton) for sharing his area-to-mass ratio database and for providing insight in a past implementation of the EVOLVE breakup model. For useful discussions we would like to thank Gene Stansbery, Paula Krisko, Carsten Wiedemann, Jonas Radtke and Holger Krag. We thank Andrew Shacker for providing useful input for future versions of the software. We thank Jonas Jonsson for useful discussions and his help editing this paper.

6. REFERENCES

1. Ailor, W.; Womack, J.; Peterson, G.; Lao, N.“ Effects of Space Debris on the Cost of Space Operation.” *61st International Astronautical Congress*; Prague; Czech Republic, 29. Sept-1 Oct. 2010; IAC-10.A6.2.10, 7p.
2. Liou, J.-C. and Johnson, N. “Instability of the present LEO satellite populations.” *Adv. Space Res.*, 41, 1046-1053, 2008.
3. Kessler, D. and Cour-Palais, B. “Collision frequency of artificial satellites: The creation of a debris belt.” *J. of Geophys*. Res., 83(A6), 2637–2646, 1978.
4. Inter-Agency Space Debris Coordination Committee, “Stability of the Future LEO Environment”, IADC-12-08, Rev. 1, January 2013
5. Liou, J.-C. and Johnson, N.L., Hill, N.M. “Controlling the growth of future LEO debris populations with active debris removal.” *Acta Astronautica* 66(2010), 648--653.
6. McKnight, D.S; Di Pentino, F; Kaczmarek, A.; Dingman, P.,”Sstem Engineering Analysis of Derelict Collision Prevention Options”, 63rd International Astronautical Congress, Naples, IT, 2012l IAC-12-A6.5.2.
7. Liou, J. C., Hall, D. T., Krisko, P. H., & Opiela, J. N. (2004),” LEGEND–A three-dimensional LEO-to-GEO debris evolutionary model”, Advances in Space Research, 34(5), 981-986.
8. Liou, J. C., “Collision activities in the future orbital debris environment”, Advances in Space Research, 38(9), 2102-2106.
9. Mason, J.; Stupl J.; Marshall, W.; Levit, C. "Orbital debris–debris collision avoidance." *Advances in Space Research* Vol. 48 No. 10 (2011): 1643-1655.
10. Stupl, J.; Mason, J.; Marshall, W.; Levit, C.; Smith, C.; Olivier, S.; Pertica. A.; De Vries, W.“LightForce: An Update on Orbital Collision Avoidance Using Photon Pressure.” *63rd International Astronautical Congress*; Naples, Italy; 1-5 Oct. 2012; IAC-12.A6.5.11.p1.x14662.
11. Stupl, J.; Faber, N.; Foster, C.; Yang Yang, F.; Levit, C.; “LightForce Photon-Pressure Collision Avoidance: Efficiency Assessment on an Entire Catalogue of Space Debris”AMOS 2013,10-13 September 2013,Wailea, HI
12. Stupl, J.; Faber, N.; Foster, C.; Yang Yang, F.; Nelson, B.; Aziz, J.; Nuttall,A.; Henze, C.; Levit; C., “LightForce Photon-Pressure Collision Avoidance: Updated Efficiency analysis Utilizing a Highly Parallel Simulation Approach”, AMOS 2014, September 10-12 2014,Wailea
13. www.space-track.org
14. Levit, C. and Marshall,W. "Improved orbit predictions using two-line elements." *Advances in Space Research* Vol. 47 No.7 (2011): 1107-1115.
15. Patera, R.P. "*General method for calculating satellite collision probability*." *Journal of Guidance, Control, and Dynamics* 24.4 (2001): 716-722.
16. Dormand, J. R.; Prince, P. J. (1980), "A family of embedded Runge-Kutta formulae", *Journal of Computational and Applied Mathematics* 6 (1): 19–26
17. Hairer, E.; Nørsett, S.P.; Wanner, G. “Solving ordinary differential equations I: Non stiff problems.” New York 2008.
18. Acton, C.H. "Ancillary data services of NASA's Navigation and Ancillary Information Facility." *Planetary and Space Science* 44.1 (1996): 65-70.
19. Phipps, C.R.; Albrecht, G.; Friedman, H.; Gavel, D.; George, E.V.; Murray, J.; Ho, C.; Priedhorsky,W.; Michaelis, M.M. and Reilly, J.P. “ORION: Clearing near-Earth space debris using a 20-kW, 530-nm, Earth-based, repetitively pulsed

laser." *Laser and Particle Beams* 14 no.1 pp. 1–44, 1996.

20. McInnes, C.K. "Solar sailing: technology, dynamics, and mission applications."Springer, 1999.
21. Barton, D. K.; Falcone, R.; Kleppner, D.; Lamb, F. K.; Lau, M.K.; Lynch, H. L.; Moncton, D.; Montague, D.; Mosher, D. E.; Priedhorsky, W.; Tigner, M. and Vaughan, D. R. "Report of the American Physical Society Study Group on Boost-Phase Intercept Systems for National Missile Defense: Scientic and Technical Issues." *Reviews of Modern Physics* 76, No. 3, S1-, 2004. See Fig.21.1, p. S323.
22. Stupl, J. and Neuneck, G. "Assessment of Long Range Laser Weapon Engagements: The Case of the Airborne Laser." *Science & Global Security* 18, no. 1 (2010): 1-60, see appendix A.
23. Vallado, D.A. "Fundamentals of Astrodynamics and Applications", New York 2013, 4th edition.
24. Lemoine, F.G.; Smith, D.E.; Kunz, L.; Smith, R.; Pavlis, E.C.; Pavlis, N.K.; Klosko, S.M.; Chinn, D.S.; Torrence, M.H.; Williamson, R.G.; Cox, C.M.; Rachlin, K.E.; Wang, Y.M.; Kenyon, S.C.; Salman, R.;Trimmer, R.; Rapp, R.H.; Nerem, R.S."The development of the NASA GSFC and NIMA joint geopotential model." *Gravity, Geoid and Marine Geodesy*, International Association of Geodesy Symposia Nr 117 (1997): 461-469.
25. Folkner, W.M.; Williams, J.G.; Boggs, D.H. "The planetary and lunar ephemeris DE 421." IPN Progress Report (2008): 42-178. http://tmo.jpl.nasa.gov/progress_report/42-178/178C.pdf
26. Picone, J. M.; Hedin, A.E; Drob, D.P.; Aikin, A.C. "NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues." *Journal of Geophysical Research: Space Physics* (1978–2012) 107.A12 (2002): SIA-15.
27. Laporte, F.; Moury, M.; Pena,X. "Operational experiences in Collision Avoidance for LEO satellites", *ISU Symposium* (February 2009), http://forum2.isunet.edu/index2.php?option=com_docman&task=doc_view&gid=775&Itemid=26 .
28. Nikolaev, S.; Phillion, D.; Springer, H.K.; de Vries,W.; Jiang,M.; Pertica, A.; Henderson, J.; Horsley, M.; Olivier, S. "Brute Force Modeling of the Kessler Syndrome", *AMOS 2012*.
29. Johnson, N. L., Krisko, P. H., Liou, J. C., & Anz-Meador, P. D. (2001), " NASA's new breakup model of EVOLVE 4.0", Advances in Space Research, 28(9), 1377-1384.
30. Krisko, P., "Proper implementation of the 1998 NASA breakup model.", Orbital Debris Quarterly News 15.4 (2011): 1-10.
31. ESA/ESOC, "ESA register of Objects in Space EsaROS", editor H. Klinkrad, Darmstadt 2008
32. Barton, D.K; Brillinger, D.; El-Shaarawi, A.H.; McDaniel, P.; Pollock, K.H.; Tuley, M.Y. "Final Report of the Haystack Orbital Debris Data Review Panel", NASA Technical Memorandum 4809, Feb. 1998.
33. NASA Orbital Debris Program Office, "HISTORY OF ON-ORBIT SATELLITE FRAGMENTATIONS", 14th Edition, June 2008, NASA/TM-2008-214779.
34. http://celestrak.com/SpaceData